

The matrix–vector complexity of $Ax = b$

Michał Dereziński

University of Michigan

DEREZIN@UMICH.EDU

Ethan N. Epperly

University of California Berkeley

EPPERLY@BERKELEY.EDU

Raphael A. Meyer

University of California Berkeley & ICSI

RAM900@BERKELEY.EDU

Editors: Steve Hanneke and Tor Lattimore

Abstract

Matrix–vector algorithms, particularly Krylov subspace methods, are widely viewed as the most effective algorithms for solving large systems of linear equations. This paper establishes lower bounds on the worst-case number of matrix–vector products needed by such an algorithm to approximately solve a general linear system. The first main result is that, for any matrix–vector algorithm which is allowed the use of randomization and can perform products with both a matrix and its transpose, $\Omega(\kappa \log(1/\varepsilon))$ matrix–vector products are necessary to solve a linear system with condition number κ to accuracy ε , matching an upper bound for conjugate gradient on the normal equations. The second main result is that one-sided algorithms, which lack access to the transpose, must use n matrix–vector products to solve an $n \times n$ linear system, even when the problem is perfectly conditioned. Both main results include explicit constants that match known upper bounds up to a factor of four. These results rigorously demonstrate the limitations of matrix–vector algorithms and confirm the optimality of widely used Krylov subspace algorithms.

Keywords: query complexity, matrix–vector complexity, linear systems of equations, Krylov subspace methods, polynomial approximation

1. Introduction

Solving systems of linear algebraic equations is a fundamental problem in computer science and mathematics, and algorithms for this task are used in almost every area of modern computation, including machine learning (Mohri et al., 2018), scientific computing (Greenbaum, 1997), and optimization (Boyd and Vandenberghe, 2004). Given an invertible matrix $A \in \mathbb{R}^{n \times n}$ and a vector $b \in \mathbb{R}^n$, the objective is to find the vector $x \in \mathbb{R}^n$ for which $Ax = b$. The standard algorithms for this task are variants of Gaussian elimination, which run in $\mathcal{O}(n^\omega)$ operations when implemented using fast matrix multiplication.¹ Determining the optimal complexity for the linear system problem is a major open question (Spielman, 2024).

If we are willing to settle for an *approximate* solution to $Ax = b$, the design space for algorithms becomes larger. Significant attention has gone to the class of matrix–vector algorithms, which learn about A *only* through the matrix–vector product (matvec) primitives $z \mapsto Az$ and $z \mapsto A^T z$. These methods are commonly understood to be among the only approaches for solving large linear systems (Greenbaum, 1997). The most popular matrix–vector algorithms for solving linear systems are *Krylov subspace methods*, such as the conjugate gradient and GMRES algorithms, which see wide use in practice. Given the importance of these methods, it is natural to ask:

1. Here, $\omega \leq 2.371 \dots$ is the matrix multiplication exponent.

How many matvecs are necessary and sufficient to approximately solve $\mathbf{Ax} = \mathbf{b}$?

To specify this problem completely, we must determine an accuracy requirement for an approximate solution $\tilde{\mathbf{x}}$. For this article, we seek a vector $\tilde{\mathbf{x}}$ with a small relative residual:

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon \|\mathbf{b}\|_2 \quad \text{for specified } \varepsilon > 0. \quad (1.1)$$

Throughout this paper, $\|\cdot\|_2$ will denote the ℓ_2 norm of a vector or the spectral norm of a matrix. See [Appendix E](#) for a discussion of other error metrics.

1.1. Background and research questions

Most work on matrix–vector complexity for the $\mathbf{Ax} = \mathbf{b}$ problem focuses on the case when \mathbf{A} is symmetric positive definite (SPD). In this case, the standard Krylov subspace algorithms are conjugate gradient and MINRES ([Greenbaum, 1997](#), Algs. 2 & 4). The latter achieves the guarantee (1.1) using $\mathcal{O}(\sqrt{\kappa} \log(1/\varepsilon))$ matvecs. Here, $\kappa := \text{cond}(\mathbf{A}) := \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = \sigma_{\max}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$ is the *condition number*. Classical lower bounds confirm that $\Omega(\sqrt{\kappa} \log(1/\varepsilon))$ matvecs are necessary for any deterministic algorithm to achieve this guarantee ([Nemirovsky and Yudin, 1983](#), Sec. 7.2). However, with the rapidly increasing use of randomization in computational linear algebra ([Martinsson and Tropp, 2020](#); [Murray et al., 2023](#); [Dereziński and Mahoney, 2024](#)), a lower bound against deterministic algorithms is insufficient to understand the difficulty of solving a given problem. This limitation was partially addressed by [Braverman et al. \(2020\)](#), who proved a weaker $\Omega(\sqrt{\kappa}/\text{polylog}(\kappa))$ lower bound for SPD linear systems against randomized algorithms.

It is natural to inquire how the matrix–vector complexity (of both deterministic and randomized algorithms) changes when we consider general linear systems for which \mathbf{A} is not SPD. In this setting, we can distinguish between a *two-sided matrix–vector algorithm*, which can perform matrix–vector products with both \mathbf{A} and \mathbf{A}^\top , and a *one-sided matrix–vector algorithm*, which can only perform matrix–vector products with \mathbf{A} . Both one- and two-sided algorithms are studied and used in practice ([Greenbaum, 1997](#)).

To solve $\mathbf{Ax} = \mathbf{b}$ using a two-sided matrix–vector algorithm, we can convert the problem into an SPD problem by passing to the *normal equations*

$$(\mathbf{A}^\top \mathbf{A})\mathbf{x} = \mathbf{A}^\top \mathbf{b}.$$

The matrix $\mathbf{A}^\top \mathbf{A}$ is SPD, so the conjugate gradient method can solve this system. The condition number of the normal equations is larger, $\text{cond}(\mathbf{A}^\top \mathbf{A}) = \kappa^2$, implying that conjugate gradient on the normal equations requires $\mathcal{O}(\sqrt{\kappa^2} \log(1/\varepsilon)) = \mathcal{O}(\kappa \log(1/\varepsilon))$ matvecs to produce an ε -accurate solution ([Nachtigal et al., 1992](#), Eqn. 2.12). This is known to be tight for deterministic algorithms ([Traub et al., 1988](#), p. 180). For randomized algorithms, no existing lower bounds extend beyond the SPD case. We ask:

Does every two-sided algorithm require $\Omega(\kappa \log(1/\varepsilon))$ matvecs in the worst case?

The most popular one-sided matrix–vector algorithm is the GMRES algorithm ([Saad and Schultz, 1986](#)). This algorithm has many desirable properties: When it is applied to an SPD system, it

achieves the same $\mathcal{O}(\sqrt{\kappa} \log(1/\varepsilon))$ complexity as MINRES, and it converges rapidly on many problems in practice. However, this algorithm also has a fundamental weakness: There exists a perfectly conditioned linear system ($\kappa = 1$) on which GMRES requires a full n iterations to solve the system to any accuracy level $\varepsilon < 1$ (Nachtigal et al., 1992, Ex. C). This raises the question:

Does every one-sided algorithm require $\Omega(n)$ matvecs in the worst case, even when $\kappa = 1$?

1.2. Our results

This paper answers both of these questions in the affirmative. Our first result concerns two-sided matrix–vector algorithms.

Theorem 1.1 (Linear systems: Lower bound against two-sided algorithms) *Fix $\eta > 0$. There does not exist any randomized algorithm that takes inputs κ , ε , and \mathbf{b} , which computes fewer than $\frac{1-\eta}{4}\kappa \log(1/\varepsilon)$ two-sided matrix–vector products with \mathbf{A} , and which returns a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon \|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all numbers $\varepsilon > 0$ and $\kappa \geq 1$ and matrices \mathbf{A} of condition number $\text{cond}(\mathbf{A}) \leq \kappa$.

This result shows that conjugate gradient on the normal equations is optimal from the perspective of worst-case complexity.² A strength of this result is its explicitness, with the complexity bound featuring no hidden constants. Conjugate gradient on the normal equations achieves the accuracy guarantee (1.1) in $(1 + o(1))\kappa \log(1/\varepsilon)$ matvecs (Greenbaum, 1997, Eqn. 3.6), so the lower and upper bounds match to a factor of roughly 4.

Our second result concerns one-sided, transpose-free algorithms.

Theorem 1.2 (Linear systems: Lower bound against one-sided algorithms) *Consider any randomized algorithm that takes a vector \mathbf{b} and a one-sided matrix–vector oracle for a matrix \mathbf{A} and outputs a vector $\tilde{\mathbf{x}}$, and suppose n is larger than some universal constant. Then:*

$$\begin{aligned} \text{Achieving } \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{2}\|\mathbf{b}\|_2 & \quad \text{with prob.} \geq \frac{2}{3} & \quad \text{requires at least } \lceil n/2 \rceil \text{ matvecs;} \\ \text{Achieving } \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{0.2}{\sqrt{n}}\|\mathbf{b}\|_2 & \quad \text{with prob.} \geq \frac{2}{3} & \quad \text{requires } n \text{ matvecs.} \end{aligned}$$

Moreover, the hard instance \mathbf{A} can be taken to be orthogonal ($\text{cond}(\mathbf{A}) = 1$).

This result confirms that GMRES’s requirement of all n matvecs is optimal, even for perfectly conditioned systems. More precisely, to solve $\mathbf{Ax} = \mathbf{b}$ to constant accuracy requires accessing half the matrix ($n/2$ matvecs), and solving to accuracy $\mathcal{O}(1/\sqrt{n})$ requires reading the entire matrix. Unless we assume some structure on \mathbf{A} beyond its condition number, access to the transpose \mathbf{A}^T is necessary to solve linear systems of equations with better than trivial complexity.

Algorithms like conjugate gradient and GMRES are already known to be optimal among certain classes of algorithms, like within the class of deterministic single-vector Krylov methods (see

2. We note that there are reasons in practice to avoid iterative methods based on the normal equations, including numerical stability (Paige and Saunders, 1982, Sec. 8.2) and better-than-worst case performance (Nachtigal et al., 1992).

Greenbaum (1997, Ch. 2) and Nemirovsky and Yudin (1983, Sec. 7.2)). However, prior works do not fully answer if these methods are optimal within the larger class of general randomized matrix–vector algorithms.³ Indeed, there have been many examples of randomized algorithms that achieve surprising improvements to the complexity of problems in linear algebra (Rokhlin and Tygert, 2008; Clarkson and Woodruff, 2009, 2017; Musco and Woodruff, 2017; Meyer et al., 2021; Peng and Vempala, 2024; Dereziński and Sidford, 2026). However, despite all efforts, no improvements to the worst-case complexity of solving general linear systems have materialized. Our research demonstrates why this is the case: Linear system solving is *not* amenable to such a speedup, in both the one- and two-sided cases.

Our Techniques. We use different techniques to handle the one- and two-sided cases. The two-sided lower bound (Theorem 1.1) uses an approach based on the *nonexistence* of good polynomial approximations, extending an argument of Chewi et al. (2024). We show that since no polynomial can approximate the function $f(x) = 1/x$ to error ε on the split interval $\mathcal{S} := [-\kappa, -1] \cup [1, \kappa]$ with degree less than $t = \mathcal{O}(\kappa \log(1/\varepsilon))$, no algorithm can use fewer than $\mathcal{O}(t)$ matvecs to estimate $\text{tr}(f(\mathbf{A})) = \text{tr}(\mathbf{A}^{-1})$ for all matrices with eigenvalues in \mathcal{S} . We then make a reduction, showing that any linear system solver satisfying (1.1) can be used to estimate $\text{tr}(\mathbf{A}^{-1})$, resulting in a proof of Theorem 1.1. See Section 2 for details.

The one-sided lower bound (Theorem 1.2) follows from a new *hidden Haar theorem* (Theorem 3.1), which shows that a one-sided matrix–vector algorithm cannot learn much information about the rows of a matrix unless it expends $\Omega(n)$ matvecs. Theorem 1.2 follows because, for an orthogonal linear system $\mathbf{Q}\mathbf{x} = \mathbf{b}$, computing the solution $\mathbf{Q}^{-1}\mathbf{b} = \mathbf{Q}^T\mathbf{b}$ requires learning a linear combination of \mathbf{Q} ’s rows. See Section 3 for details.

1.3. Extension: Stronger lower bounds for solving SPD systems

The focus of this article is on solving general, not necessarily SPD, linear systems. Indeed, the prior work of Braverman et al. (Braverman et al. (2020)) has already shown that $\Omega(\sqrt{\kappa}/\text{polylog}(\kappa))$ matvecs are needed for a randomized algorithm to solve a SPD linear system. However, our analysis is able to provide tighter lower bounds for the SPD case as well, removing unnecessary polylogarithmic factors in the condition number κ and obtain the correct logarithmic dependence on the inverse-accuracy $1/\varepsilon$:

Theorem 1.3 (SPD Linear systems: Lower bound against two-sided algorithms) *Fix $\eta > 0$. There does not exist any randomized algorithm that takes inputs κ , ε , and \mathbf{b} , which computes fewer than $\frac{1-\eta}{4}\sqrt{\kappa}\log(1/\varepsilon)$ two-sided matrix-vector products with \mathbf{A} , and which returns a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon\|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all numbers $\varepsilon > 0$ and $\kappa \geq 1$ and SPD matrices \mathbf{A} of condition number $\text{cond}(\mathbf{A}) \leq \kappa$.

This result sharpens the log factors over the prior work and, since conjugate gradient converges in $\mathcal{O}(\sqrt{\kappa}\log(1/\varepsilon))$ matvecs, shows that the dependence on ε in the upper bound analysis is tight. The proof of Theorem 1.3 uses the same basic argument as the proof of the lower bound in the general case Theorem 1.1. See Appendix E.2 for details.

3. Braverman et al. (2020) prove that $\Omega(\sqrt{\kappa}/\text{polylog}(\kappa))$ matvecs are needed to solve a SPD linear system to constant error, partially resolving this question.

1.4. Implication: Fine-grained lower bounds and complexity separation

Our main results describe the complexity of solving linear systems by parameterizing the input via the matrix dimension n and its condition number $\kappa = \text{cond}(\mathbf{A})$. Yet, it is widely accepted that Krylov methods often dramatically outperform the worst-case complexity bounds when the singular values of \mathbf{A} exhibit additional structure, such as clustering or outliers (Greenbaum, 1997).

An important and well-studied example of this arises when \mathbf{A} has a number of large outlying singular values that blow up its condition number, which arises in smoothed analysis (Spielman and Teng, 2009), optimization (Boyd and Vandenberghe, 2004), and machine learning (Zhang et al., 2013). In this setting, conjugate gradient on the normal equations is known to solve such linear systems in an improved $\mathcal{O}(k + \kappa_k \log(1/\varepsilon))$ matvecs (Axelsson and Lindskog, 1986), where $\kappa_k := \sigma_{k+1}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$ denotes the condition number without the top k singular values. This more fine-grained notation of complexity, parameterized by both k and κ_k , circumvents our basic two-sided lower bound (Theorem 1.1). However, our results can be extended to this different complexity measure, again showing that conjugate gradient on the normal equations is optimal:

Theorem 1.4 (Linear systems: Fine-grained lower bound) *There does not exist a randomized algorithm that takes inputs k , κ_k , ε , and \mathbf{b} , which computes fewer than $\Omega(k + \kappa_k \log(1/\varepsilon))$ two-sided matrix-vector products with \mathbf{A} and returns $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon\|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all matrices \mathbf{A} such that $\sigma_{k+1}(\mathbf{A})/\sigma_{\min}(\mathbf{A}) \leq \kappa_k$.

This corollary follows immediately by combining our results with a reduction from Dereziński et al. (2025, Lem. 27 & Rem. 8); see Appendix C for details.

Notably, this result implies a *complexity separation* between matrix–vector algorithms and general algorithms that can access the entries of \mathbf{A} directly. If \mathbf{A} is a dense matrix and we implement matvecs using the standard algorithm, then Theorem 1.4 shows that $\Omega(n^2k + n^2\kappa_k \log(1/\varepsilon))$ time is needed to solve a linear system. However, recent work (Dereziński and Yang, 2024; Dereziński et al., 2025; Dereziński and Sidford, 2026) shows that if \mathbf{A} is stored as a dense array, we can solve $\mathbf{Ax} = \mathbf{b}$ in $\tilde{\mathcal{O}}(k^3 + n^2\kappa_k \log(1/\varepsilon))$ time. So, for large n and k , having access to the matrix offers a direct improvement over matrix–vector access, demonstrating a complexity separation between matrix–vector algorithms and general algorithms. This is the first result of this kind for general, dense linear systems.

1.5. Related work

The most relevant background material, and its relationship to our work, were discussed in Sections 1.1 and 1.2. We briefly mention a few more areas of related work.

Upper and lower bounds for Krylov methods. Understanding the behavior of Krylov linear solvers has been a decades-long research challenge in numerical analysis. The literature is extensive; see Greenbaum (1997) for a summary of the theory. Lower bounds for the rate of convergence of Krylov methods can be derived from potential theory (Driscoll et al., 1998). These lower bounds apply only to specific Krylov methods, not general randomized matrix–vector algorithms.

Transpose-free linear algebra. Our work fits into the growing research topic of *transpose-free linear algebra* (Boullé et al., 2024; Halikias, 2025), which aims to understand the relative power of

one- and two-sided algorithms for various linear algebra problems. The results most similar to ours appear in the thesis of Halikias (2025, Sec. 5.2), who analyzes one-sided matrix–vector algorithms for solving linear systems and least-squares problems. Her results show that, even in the favorable situation when $\text{cond}(\mathbf{A}) = 1$ and we perform $n - 1$ matvecs $\mathbf{A}z_1, \dots, \mathbf{A}z_{n-1}$,⁴ the data collected by the algorithm is not sufficient to characterize the solution to a system of linear equations: There exists another well-conditioned matrix $\tilde{\mathbf{A}}$ that is undistinguished by the queries

$$\mathbf{A}z_i = \tilde{\mathbf{A}}z_i \quad \text{for } i = 1, \dots, n - 1$$

but for which the solutions of the linear systems are different

$$\|\tilde{\mathbf{A}}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{b}\|_2 > \delta \quad \text{for some } \delta > 0 \text{ depending on } \mathbf{A}, \mathbf{b}, z_1, \dots, z_{n-1}.$$

Halikias’s result establishes information-theoretic limitations on the power of one-sided matrix–vector algorithms. However, the level of accuracy δ and the hard-to-distinguish instance $\tilde{\mathbf{A}}$ depend on the random choices made by the algorithm. To establish lower bounds on matrix–vector complexity, we must exhibit a problem instance (\mathbf{A}, \mathbf{b}) and a fixed level of accuracy $\delta > 0$, independent of the random choices made by the algorithm, for which any algorithm fails. As such, while our analysis and Halikias’ share similar aims, our results are formally incomparable.

Quantum linear systems. Our $\Omega(\kappa)$ lower bound for two-sided matrix–vector algorithms mirrors the $\Omega(\kappa)$ lower bound for the quantum linear systems problem (Harrow et al., 2009, p. 150502-3). In that context, solving implicit sparse systems in $\mathcal{O}(\kappa^{1-\eta} \text{polylog}(n))$ time would imply BQP = PSPACE and allow for sub-linear simulation of arbitrary quantum circuits. While the $\Omega(\kappa)$ scaling in both settings is striking, it remains unclear how to derive one result from the other or reduce them both to some common principle.

1.6. Notation

All logarithms are natural. The vector ℓ_2 norm and the operator $\ell_2 \rightarrow \ell_2$ norm are denoted $\|\cdot\|_2$, and the Frobenius norm is $\|\cdot\|_F$. The orthogonal matrices come equipped with a unique rotation-invariant probability distribution, and a draw from this distribution is called a *Haar-random orthogonal matrix*. When this notation is used in the hypothesis of a result, we take this to mean that $a \leq Cb$ for a *sufficiently small* universal constant $C > 0$. We write $a \asymp b$ when $a \lesssim b \lesssim a$. For any function $f : \mathbb{R} \rightarrow \mathbb{R}$ and symmetric matrix \mathbf{A} with eigendecomposition $\mathbf{A} = \mathbf{U} \text{diag}(\lambda_1, \dots, \lambda_n) \mathbf{U}^\top$, the matrix function is $f(\mathbf{A}) := \mathbf{U} \text{diag}(f(\lambda_1), \dots, f(\lambda_n)) \mathbf{U}^\top$.

2. Lower bounds for two-sided algorithms

In this section, we develop lower bounds for solving linear systems using two-sided matrix–vector algorithms. Our analysis adopts and extends a relatively new approach to proving such bounds that was developed in recent work of Chewi et al. (2024), who used this technique to prove that $\Omega(\sqrt{\kappa} \log n)$ matvecs are necessary to estimate the trace-inverse $\text{tr}(\mathbf{A}^{-1})$ of an SPD matrix \mathbf{A} to relative error $\|\mathbf{A}^{-1}\|_F$. See also the follow-up work (Bakshi and Narayanan, 2023).

In this section, we present a generalized and refined version of their results that is able to prove lower bounds on the matrix–vector complexity of estimating spectral sums $\text{tr}(f(\mathbf{A}))$ or matrix–function–vector products $f(\mathbf{A})\mathbf{b}$ for arbitrary functions f and matrices \mathbf{A} with arbitrary eigenvalues.

4. Halikias also includes as an assumption that the exact solution $\mathbf{A}^{-1}\mathbf{b}$ is not in the span of the queries $\{z_1, \dots, z_{n-1}\}$.

The core theorem shows that if a function f cannot be well approximated by a degree t polynomial over a domain \mathcal{S} , then any method that computes $\text{tr}(f(\mathbf{A}))$ or $f(\mathbf{A})\mathbf{b}$ for matrices \mathbf{A} with eigenvalues belonging to \mathcal{S} must use $\Omega(t)$ matvecs. The proof architecture is adapted from [Chewi et al. \(2024\)](#), but the details have been entirely overhauled to support a general function f and a general set of eigenvalues \mathcal{S} , as well as to obtain large, explicit constants.

2.1. Matrix-vector complexity and polynomial approximability

We are interested in computing the solution $\mathbf{A}^{-1}\mathbf{b}$ to a linear system of equations. More generally, we might be interested in computing the matrix-function-vector product $f(\mathbf{A})\mathbf{b}$, for which the linear system problem is the special case $f(x) = 1/x$. It is straightforward to see that if f can be approximated to high accuracy by a degree- t polynomial *on the spectrum of \mathbf{A}* , then we can compute $f(\mathbf{A})\mathbf{b}$ to high accuracy using a matrix-vector algorithm:

Proposition 2.1 (Matrix-function-vector product: Upper bound) *Let \mathbf{A} be a normal matrix with eigenvalues belonging to a set $\mathcal{S} \subseteq \mathbb{C}$, and suppose that there exists a degree- t polynomial $p(x) = c_0 + c_1x + \dots + c_tx^t$ such that*

$$\max_{\lambda \in \mathcal{S}} |f(\lambda) - p(\lambda)| \leq \eta \max_{\lambda \in \mathcal{S}} |f(\lambda)|.$$

Then, the vector $\tilde{\mathbf{x}} := p(\mathbf{A})\mathbf{b} = c_0\mathbf{b} + c_1\mathbf{A}\mathbf{b} + \dots + c_t\mathbf{A}^t\mathbf{b}$ can be computed using t matrix-vector products with \mathbf{A} , and it approximates $f(\mathbf{A})\mathbf{b}$ well:

$$\|\tilde{\mathbf{x}} - f(\mathbf{A})\mathbf{b}\|_2 \leq \|p(\mathbf{A}) - f(\mathbf{A})\|_2 \|\mathbf{b}\|_2 \leq \eta \cdot \|f(\mathbf{A})\|_2 \|\mathbf{b}\|_2.$$

This type of bound is standard. It shows that we can accurately and efficiently approximate $f(\mathbf{A})\mathbf{b}$ for any function f that is close to a polynomial. In fact, nearly all known matrix-vector algorithms for efficiently computing $f(\mathbf{A})\mathbf{b}$ work directly or indirectly by forming a polynomial approximation to f . Therefore, it is natural to conjecture something of a converse to [Proposition 2.1](#): *If a function cannot be approximated by polynomials of small degree, then it is hard for a matrix-vector algorithm to compute $f(\mathbf{A})\mathbf{b}$.*

2.2. Spectral sums are easier than matrix-function-times-vector

Rather than proving hardness results directly for computing $f(\mathbf{A})\mathbf{b}$, we instead prove hardness for computing the *spectral sum* $\text{tr}(f(\mathbf{A}))$. The following result shows that if we can approximate $f(\mathbf{A})\mathbf{b}$ efficiently, then we can also estimate spectral sums efficiently. As such, lower bounds for the spectral sum problem immediately yield lower bounds for the $f(\mathbf{A})\mathbf{b}$ problem.

Proposition 2.2 (From matrix-function-times-vector to spectral sum) *Consider any oracle that takes \mathbf{A} and \mathbf{b} as inputs and returns a vector $\tilde{\mathbf{x}}$ such that*

$$\|\tilde{\mathbf{x}} - f(\mathbf{A})\mathbf{b}\|_2 \leq \frac{1}{n} \|f(\mathbf{A})\|_{\text{F}} \|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}.$$

Then there exists an algorithm which invokes the oracle once and outputs an estimate $\tilde{\text{tr}}$ such that

$$|\tilde{\text{tr}} - \text{tr}(f(\mathbf{A}))| \leq 5 \|f(\mathbf{A})\|_{\text{F}} \quad \text{with prob.} \geq \frac{2}{3}.$$

Stochastic trace estimation results of this form are standard; the proof is in [Appendix D.1](#).

2.3. Lower bound for estimating a spectral sum

With the necessary preparation in place, we can state a general lower bound for the spectral sum problem in terms of the polynomial (in)approximability of f . We make the following definition:

Definition 2.3 (Inapproximable function) *Let $f : \mathcal{S} \rightarrow \mathbb{R}$ be a function on a domain $\mathcal{S} \subseteq \mathbb{R}$. The function f is (t, α) -inapproximable on \mathcal{S} if all polynomials p of degree at most t satisfy*

$$\max_{\lambda \in \mathcal{S}} |p(\lambda) - f(\lambda)| \geq \alpha \cdot \max_{\lambda \in \mathcal{S}} |f(\lambda)|.$$

Our general result states that, for a sufficiently large symmetric matrix \mathbf{A} with eigenvalues in the set \mathcal{S} , at least t matvecs are needed to estimate spectral sums.

Theorem 2.4 (Spectral sum: Lower bound, special case of Theorem A.1) *Fix a bounded set $\mathcal{S} \subseteq \mathbb{R}$. Let $f : \mathcal{S} \rightarrow \mathbb{R}$ be a function that is (t, α) -inapproximable on \mathcal{S} . Suppose that $n \gtrsim t^6/\alpha^2$. Then any matrix–vector algorithm that returns an estimate $\tilde{\text{tr}}$ such that*

$$|\tilde{\text{tr}} - \text{tr}(f(\mathbf{A}))| \leq 5\|\mathbf{A}\|_{\text{F}} \quad \text{with prob.} \geq \frac{2}{3}$$

for all symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenvalues in \mathcal{S} must use at least $\lfloor t/2 \rfloor$ matvecs.

Together, Propositions 2.1 and 2.2 imply that we can estimate $\text{tr}(f(\mathbf{A}))$ with $\mathcal{O}(t)$ matvecs when the function f admits an accurate degree- t polynomial approximation. Theorem 2.4 proves a converse result, showing that if f is inapproximable by degree- t polynomials, then at least $\Omega(t)$ matvecs are needed to estimate $\text{tr}(f(\mathbf{A}))$. We prove Theorem 2.4 in Appendix A by generalizing and refining the work of Chewi et al. (2024). The basic idea is to construct two matrices \mathbf{A}_1 and \mathbf{A}_2 with a common set of eigenvalues repeated with different multiplicities in such a way that these matrices “look the same” to any algorithm using $t/2$ matvecs but for which $|\text{tr}(f(\mathbf{A}_1)) - \text{tr}(f(\mathbf{A}_2))| \gg 0$.

2.4. Lower bounds for trace-inverse

To use Theorem 2.4 to prove a lower bound on the $\text{tr}(\mathbf{A}^{-1})$ and $\mathbf{A}^{-1}\mathbf{b}$ problems, we need to show that $f(x) = 1/x$ is inapproximable over an appropriate domain $\mathcal{S} \subseteq \mathbb{R}$. Since we are interested in the complexity of these tasks for symmetric matrices \mathbf{A} , not necessarily SPD, with condition number at most κ , we take the domain to be the split interval $\mathcal{S} = [-\kappa, -1] \cup [1, \kappa]$. We show the following inapproximability result:

Lemma 2.5 (Inapproximability of $1/x$ on split interval) *Fix $\kappa \geq 2$ and an integer t . Then $f(x) = 1/x$ is $(t, \frac{1}{2}e^{-2(t+1)/\kappa})$ -inapproximable on $[-\kappa, -1] \cup [1, \kappa]$.*

The proof of Lemma 2.5 appears in Section 2.6. Now, we will use this result to prove lower bounds for computing the trace-inverse. We have the following result:

Theorem 2.6 (Trace-inverse: Lower bound) *Fix any $\delta, \eta \in (0, 1)$ and suppose that $n \gtrsim \kappa^{(6+\delta)/\eta}$. Any matrix–vector algorithm that produces an estimate $\tilde{\text{tr}}$ satisfying*

$$|\tilde{\text{tr}} - \text{tr}(\mathbf{A}^{-1})| \leq 5\|\mathbf{A}^{-1}\|_{\text{F}} \quad \text{with prob.} \geq \frac{2}{3} \quad (2.1)$$

for all symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with condition number κ requires at least $\frac{1-\eta}{4}\kappa \log n$ matvecs.

Proof Fix any $t \leq \frac{1-\eta}{4}\kappa \log(n) - 1$. Then, we know that $f(x) = 1/x$ is (t, α) -inapproximable on $[-\kappa, -1] \cup [1, \kappa]$ for

$$\alpha = \frac{1}{2}e^{-2(t+1)/\kappa} \geq \frac{1}{2}n^{-\frac{1-\eta}{2}},$$

which gives us

$$\frac{t^6}{\alpha^2} \lesssim \kappa^6 n^{1-\eta} \log^6 n \lesssim n.$$

The last inequality holds because $n \gtrsim \kappa^{(6+\delta)/\eta}$. Appealing to [Theorem 2.4](#) completes the proof. ■

It is informative to instantiate this result in a few regimes. First, taking the limit $\eta \rightarrow 0$ yields:

Corollary 2.7 *For any $\eta > 0$, any algorithm using fewer than $(\frac{1}{4} - \eta)\kappa \log n$ two-sided matvecs cannot estimate $|\tilde{\text{tr}} - \text{tr}(\mathbf{A}^{-1})| \leq 5\|\mathbf{A}^{-1}\|_{\text{F}}$ with probability at least $2/3$ for all matrices \mathbf{A} .*

This result gives us the sharpest constant in the complexity, but it does not specify the dimension n needed to produce a hard instance. If we take $\eta \rightarrow 1$, we see that it takes $\Omega(\kappa \log n)$ matvecs to estimate the spectral sum when $n \gtrsim \kappa^{6+\delta}$ for any $\delta > 0$. Curiously, our techniques provide no lower bound when $n \lesssim \kappa^6$. We leave this as a natural question for future research.

2.5. Lower bounds for linear systems of equations

As a corollary of the [Theorem 2.6](#), we show lower bound on the number of matvecs used by any method which achieves a standard guarantee for solving linear systems.

Theorem 2.8 (Linear systems: Lower bound) *Fix any $\delta, \eta \in (0, 1)$ and suppose $n \gtrsim \kappa^{(6+\delta)/\eta}$. Any algorithm using fewer than $\frac{1-\eta}{4}\kappa \log n$ two-sided matvecs with a matrix \mathbf{A} cannot return a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6} \quad (2.2)$$

for all symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\text{cond}(\mathbf{A}) \leq \kappa$ and all vectors $\mathbf{b} \in \mathbb{R}^n$. Moreover, attaining the guarantee

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{2}\|\mathbf{b}\|_2 \quad \text{with prob.} \geq 1 - \frac{1}{10 \log n} \quad (2.3)$$

for the same class of problems requires $\Omega(\kappa)$ matvecs.

The proof of the first result is immediate from the spectral sum lower bound ([Theorem 2.6](#)) together with the trace estimation upper bound ([Proposition 2.2](#)). The second result requires another ingredient, iterative refinement ([Proposition D.1](#)). See [Appendix D.2](#) for the full proofs.

An interesting feature of this result is that the hard instance \mathbf{A} is symmetric, for which there is no distinction between matvecs with \mathbf{A} and \mathbf{A}^\top . In particular, our results show that solving linear systems requires $\Omega(\kappa \log(1/\varepsilon))$ matvecs *even when \mathbf{A} is symmetric*. Of course, solving a linear system $\mathbf{Ax} = \mathbf{b}$ for a general matrix \mathbf{A} can only be more difficult than the symmetric case, so the lower bound also applies against general matrices \mathbf{A} with condition number κ as well.

A linear system is solved to residual accuracy ε when

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon\|\mathbf{b}\|_2,$$

which satisfies the bound (2.2) when $\varepsilon \asymp \frac{1}{n}$. In this “high accuracy” regime, we show that $t \gtrsim \frac{1}{4}\kappa \log(1/\varepsilon)$ matvecs are needed, recovering [Theorem 1.1](#). Further, in the “low accuracy” regime where $\varepsilon \leq \frac{1}{2}$, we show that $t \gtrsim \kappa$ matvecs are needed.

2.6. Inapproximability of the inverse function on a split interval

All that remains to prove [Theorem 1.1](#) is the polynomial inapproximability result ([Lemma 2.5](#)), which states that the function $f(x) = 1/x$ requires a degree $t = \Omega(\kappa \log(1/\varepsilon))$ polynomial to approximate it to accuracy ε . The proof starts with a standard bound on the inapproximability of the function $f(x) = 1/x$ on a non-split interval.

Imported Lemma 2.9 (Inapproximability of $1/x$ on interval; Kraus et al. (2010, Cor. 2.2)) *The function $f(x) = 1/x$ is (t, α) -inapproximable on the interval $[1, b]$ for*

$$\alpha = \frac{1}{2} \left(1 - \frac{2}{\sqrt{b+1}}\right)^{t+1} \left(1 - \frac{1}{\sqrt{b}}\right)^2 \geq \frac{1}{2} e^{-4(t+1)/\sqrt{b}}.$$

Using the fact that $f(x) = 1/x$ is odd, we reduce the inapproximability of f on the split interval $[-\kappa, -1] \cup [1, \kappa]$ to the inapproximability of f on the one-sided interval $[1, \kappa^2]$, yielding [Lemma 2.5](#).

Proof of Lemma 2.5 Let p be any polynomial of degree at most t , which we think of as an approximation to $f(x) = 1/x$ on the split interval $\mathcal{S} := [-\kappa, -1] \cup [1, \kappa]$. Extract the odd part $\tilde{p}(x) := \frac{1}{2}(p(x) - p(-x))$ of p , which is a polynomial of degree at most t . Note that

$$\tilde{p}(x) - f(x) = \frac{p(x) - f(x)}{2} - \frac{p(-x) + f(x)}{2}.$$

Since $f(x) = -f(-x)$ and \mathcal{S} is symmetric,

$$|\tilde{p}(x) - f(x)| \leq \frac{|p(x) - f(x)|}{2} + \frac{|p(-x) - f(-x)|}{2} \leq \max_{x \in \mathcal{S}} |p(x) - f(x)|.$$

That is, \tilde{p} is no worse than p at approximating f . Next, since $\tilde{p}(x)$ is an odd polynomial, we can write $\tilde{p}(x) = xq(x^2)$ for some polynomial q of degree at most $t_q := \frac{1}{2}(t-1)$. Then, for any $x \in \mathcal{S}$,

$$|\tilde{p}(x) - f(x)| = |xq(x^2) - 1/x| = |x|q(x^2) - 1/x^2| \geq |q(x^2) - 1/x^2|.$$

Defining $y := x^2 \in [1, \kappa^2]$ then [Imported Lemma 2.9](#) implies that

$$\max_{x \in \mathcal{S}} |p(x) - f(x)| \geq \max_{x \in \mathcal{S}} |\tilde{p}(x) - f(x)| \geq \max_{y \in [1, \kappa^2]} |q(y) - 1/y| \geq \frac{1}{2} e^{-4(t_q+1)/\kappa}.$$

Substituting $t_q = \frac{1}{2}(t-1)$ completes the proof. ■

3. Lower bound against one-sided algorithms

In this section, we develop lower bounds for one-sided matrix–vector algorithms. Our proofs use the hidden random matrix method and require the development of a new *hidden Haar theorem*.

3.1. Hidden Haar theorem

One of the most powerful proof techniques for proving lower bounds on matrix–vector algorithms is the *hidden random matrix method* ([Simchowit et al., 2018](#); [Braverman et al., 2020](#); [Amsel et al., 2026](#)). The method works by identifying a random matrix \mathbf{A} with the following property: Up to rotation, \mathbf{A} contains a submatrix that follows a known distribution and is independent of the matrix–vector queries. One then shows how the uncertainty induced by the hidden matrix prevents any

algorithm from solving a given linear algebra task. Previous works have taken \mathbf{A} to be a Wishart matrix (Braverman et al., 2020; Chewi et al., 2024; Meyer, 2024; Amsel et al., 2026; Meyer and Avron, 2026) or GOE matrix (Simchowitz et al., 2018; Jiang et al., 2021; Woodruff et al., 2022).

In this work, we use a Haar-random (i.e. uniformly random) orthogonal matrix. The key ingredient to using this random matrix is a *hidden Haar theorem*, which characterizes the distribution of the unobserved part of the matrix.

Theorem 3.1 (Hidden Haar) *Let $\mathbf{Q} \in \mathbb{R}^{n \times n}$ be a Haar-random orthogonal matrix, and consider the transcript*

$$\mathcal{T} = (\mathbf{z}_1, \mathbf{Q}\mathbf{z}_1, \dots, \mathbf{z}_t, \mathbf{Q}\mathbf{z}_t)$$

of a deterministic one-sided matrix-vector algorithm making linearly independent queries $\mathbf{z}_1, \dots, \mathbf{z}_t$. Then we can construct an orthogonal matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$ and a matrix $\mathbf{V} \in \mathbb{R}^{n \times t}$ with orthonormal columns, depending only on the transcript \mathcal{T} , such that

$$\mathbf{QU} = \begin{bmatrix} \mathbf{V} & \mathbf{W} \end{bmatrix} \text{ is a Haar-random orthogonal matrix.}$$

Further, conditioned on the transcript \mathcal{T} , the matrix $\mathbf{W} \in \mathbb{R}^{n \times (n-t)}$ is distributed uniformly at random over all orthonormal matrices spanning the subspace orthogonal to $\text{range}(\mathbf{V})$.

Here, \mathbf{W} is the hidden submatrix. We prove the hidden Haar theorem in [Appendix B](#). The assumption that the algorithm makes linearly independent queries is without loss of generality, as any algorithm making t matrix-vector queries can be simulated by an algorithm making $\tilde{t} \leq t$ linearly independent queries. Haar matrices have also been previously used in information-theoretic lower bounds for first-order optimization [Carmon et al. \(2020\)](#), though we are unaware of any work that uses them in a matrix-vector context.

3.2. Lower bounds for linear systems

We can use the hidden Haar theorem to prove lower bounds on the complexity of one-sided linear system solvers. We begin by stating some elementary concentration and anticoncentration results for uniformly random points on a sphere.

Proposition 3.2 (Random vectors on the sphere) *Let $\mathbf{u} \in \mathbb{R}^n$ be a uniformly random vector on the sphere of radius R , and assume $n \gtrsim 1$. Then*

- (a) **One coordinate is not too small.** $|\mathbf{u}_1| \geq 0.2 \cdot R/\sqrt{n}$ with probability $> 4/5$.
- (b) **Many coordinates are large.** $[\mathbf{u}_1]^2 + \dots + [\mathbf{u}_{\lfloor n/2 \rfloor}]^2 \geq 0.49 \cdot R^2$ with probability $> 3/5$.
- (c) **Far from any point.** For any vector \mathbf{v} , $\|\mathbf{v} - \mathbf{u}\|_2^2 \geq 0.99 \cdot R^2$ with probability $> 3/5$.

These results can be directly shown by realizing $\mathbf{u} = R \cdot \mathbf{g}/\|\mathbf{g}\|_2$ by rescaling a standard Gaussian vector $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applying simple (anti)concentration arguments for Gaussian random variables. We omit the easy proof.

Theorem 3.3 (Lower bound against one-sided solvers) *Consider the linear system of equations $\mathbf{Qx} = \mathbf{e}_1$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a Haar-random orthogonal matrix and $n \gtrsim 1$.*

(a) **Low accuracy requires $\Omega(n)$ matvecs.** Any one-sided matrix–vector algorithm producing a vector $\tilde{\mathbf{x}}$ satisfying

$$\|\mathbf{Q}\tilde{\mathbf{x}} - \mathbf{e}_1\|_2 \leq \frac{1}{2} \quad \text{with prob.} \geq \frac{2}{3}$$

requires $\lceil n/2 \rceil$ matvecs.

(b) **Modest accuracy requires n matvecs.** Any one-sided matrix–vector algorithm producing a vector $\tilde{\mathbf{x}}$ satisfying

$$\|\mathbf{Q}\tilde{\mathbf{x}} - \mathbf{e}_1\|_2 \leq \frac{0.2}{\sqrt{n}} \quad \text{with prob.} \geq \frac{2}{3}$$

requires n matvecs.

Proof By Yao’s minimax theorem (Motwani and Raghavan, 1995, Sec. 2.2), we can assume without loss of generality that the algorithm is deterministic. Consider a one-sided matrix–vector algorithm performing t matrix–vector products with transcript \mathcal{T} , which we can assume without loss of generality are linearly independent. Instantiate the matrices \mathbf{Q} , \mathbf{U} , \mathbf{W} , and \mathbf{V} from the hidden Haar theorem (Theorem 3.1). The solution to $\mathbf{Q}\mathbf{x} = \mathbf{e}_1$ is

$$\mathbf{x} = \mathbf{Q}^\top \mathbf{e}_1 = \mathbf{U} \begin{bmatrix} \mathbf{V}^\top \mathbf{e}_1 \\ \mathbf{W}^\top \mathbf{e}_1 \end{bmatrix}.$$

Let $\tilde{\mathbf{x}}$ denote the output of the algorithm, and introduce the notation

$$\tilde{\mathbf{x}} =: \mathbf{U} \begin{bmatrix} \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_2 \end{bmatrix}.$$

Then

$$\|\mathbf{Q}\tilde{\mathbf{x}} - \mathbf{e}_1\|_2^2 = \|\mathbf{V}^\top \mathbf{e}_1 - \tilde{\mathbf{x}}_1\|_2^2 + \|\mathbf{W}^\top \mathbf{e}_1 - \tilde{\mathbf{x}}_2\|_2^2 \geq \|\mathbf{W}^\top \mathbf{e}_1 - \tilde{\mathbf{x}}_2\|_2^2.$$

Therefore, a necessary condition for the solution to satisfy $\|\mathbf{Q}\tilde{\mathbf{x}} - \mathbf{e}_1\|_2 \leq \varepsilon$ is for the component $\tilde{\mathbf{x}}_2$ to satisfy

$$\|\mathbf{W}^\top \mathbf{e}_1 - \tilde{\mathbf{x}}_2\|_2^2 \leq \varepsilon^2.$$

The matrix \mathbf{V} is a uniformly random orthonormal matrix, and, conditional on \mathcal{T} , the matrix \mathbf{W} is uniformly random among orthonormal matrices spanning $[\text{range}(\mathbf{V})]^\perp$. Consequently, $\mathbf{V}^\top \mathbf{e}_1$ is distributed as the first t entries of a uniformly random vector of the unit sphere and, conditional on the transcript $\mathbf{W}^\top \mathbf{e}_1$ is uniformly distributed on the sphere of radius $\sqrt{1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2}$.

We now prove parts (a) and (b) separately.

Proof of Theorem 3.3(a). Suppose $t < \lceil n/2 \rceil$. By Proposition 3.2(b),

$$1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2 \geq 0.49 \quad \text{with prob.} > 3/5$$

The component $\tilde{\mathbf{x}}_2$ and \mathbf{W} are independent conditional on \mathcal{T} and thus, by Proposition 3.2(c),

$$\|\mathbf{W}^\top \mathbf{e}_1 - \tilde{\mathbf{x}}_2\|_2^2 \geq 0.99(1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2) \quad \text{with prob.} > 3/5.$$

These events are independent, so they occur with probability $(3/5)^2 = 0.36 > 1/3$. Ergo, no one-sided matrix–vector algorithm using $\lceil n/2 \rceil$ matvecs can achieve the guarantee

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \frac{1}{2} \|\mathbf{x}\|_2 \leq \sqrt{0.49 \cdot 0.99} \cdot \|\mathbf{x}\|_2$$

with probability $\geq 2/3$.

Proof of Theorem 3.3(b). Now suppose $t = n - 1$. By Proposition 3.2(a),

$$\sqrt{1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2} \geq 0.2/\sqrt{n} \quad \text{with prob. } > 4/5.$$

Moreover, conditional on \mathcal{T} , the number $\mathbf{W}^\top \mathbf{e}_1 \in \mathbb{R}^1$ is uniformly random between $\pm(1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2)$. The component \tilde{x}_2 can be $< (1 - \|\mathbf{V}^\top \mathbf{e}_1\|_2^2)$ with probability at most $1/2$. Thus, any matrix-vector algorithm using $n - 1$ matvecs must fail to achieve the guarantee

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \frac{0.2}{\sqrt{n}} \|\mathbf{x}\|_2$$

probability at least $(1/2)(4/5) = 2/5 > 1/3$. ■

4. Conclusion

In this paper, we resolve the worst-case matrix-vector complexity of solving linear systems $\mathbf{Ax} = \mathbf{b}$ for one- and two-sided matrix-vector algorithms, up to small constant factors. Along the way, we also prove a lower bound on the complexity of estimating the inverse-trace $\text{tr}(\mathbf{A}^{-1})$. Several natural questions remain to be explored in this space. Most directly, can further work on polynomial inapproximability enable similar matching upper and lower bounds for the $f(\mathbf{A})\mathbf{b}$ and $\text{tr}(f(\mathbf{A}))$ problems for a much wider family of functions f ?

Second, despite the $\Omega(n)$ worst-case bound on the complexity of solving linear systems with one-sided algorithms, we see that methods like GMRES converge in far fewer than n iterations for a very large class of matrices. This is even theoretically proven for certain matrices, like those with (complex) eigenvalues belonging in the right half of the complex plane. However, it remains unclear: What *natural* and *minimal* assumptions on the structure of \mathbf{A} characterize the rate at which one-sided matrix-vector methods can solve linear systems?

Lastly, and most technically, we see that the spectral sum lower bound in Theorem 2.4 does not yield a lower bound against estimating the trace to relative error ε . To estimate $\text{tr}(\mathbf{A})$ to relative error ε , the optimal rate is $\Theta(1/\varepsilon)$ (Meyer et al., 2021; Meyer, 2024). Is it possible to show a $\Omega(t/\varepsilon)$ lower bound for computing $\text{tr}(f(\mathbf{A}))$ to relative error ε if f is inapproximable up to degree t ?

Appendix A. Matrix–vector lower bounds from polynomial inapproximability

In this section, we prove the following theorem:

Theorem A.1 (Spectral sums: Lower bound) *Fix a bounded set $\mathcal{S} \subset \mathbb{R}$. Let $f : \mathcal{S} \rightarrow \mathbb{R}$ be a function that is (t, α) -inapproximable on \mathcal{S} . Suppose that*

$$n \gtrsim t^4 \cdot \max\{t^2, 1/\alpha\}.$$

Then, any matrix–vector algorithm that returns an estimate $\tilde{\text{tr}}$ such that

$$|\tilde{\text{tr}} - \text{tr}(f(\mathbf{A}))| \lesssim \frac{\alpha\sqrt{n}}{t^3} \cdot \|f(\mathbf{A})\|_{\text{F}} \quad \text{with prob.} \geq \frac{2}{3}$$

for all symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenvalues belonging to the set \mathcal{S} must use at least $\lfloor t/2 \rfloor$ matvecs.

The result [Theorem 2.4](#) stated in the text follows as a direct corollary.

[Theorem A.1](#) shows that for any function f that is hard to approximate using a degree- t polynomial, estimating $\text{tr}(f(\mathbf{A}))$ to additive error $\|f(\mathbf{A})\|_{\text{F}}$ requires at least $\Omega(t)$ matvecs. The proof is a direct generalization and systematization of the argument of [Chewi et al. \(2024, Sec. 5\)](#).

Remark A.2 (One-sided and two-sided) *This section concerns algorithms applied to a symmetric matrix \mathbf{A} , for which there is no distinction between one-sided and two-sided matrix–vector algorithms. If a linear algebra problem is hard to solve by a matrix–vector algorithm for a symmetric matrices, then it is also hard to solve for general matrices by two-sided matrix–vector algorithms.*

A.1. Review: Total variation distance

We begin with a brief review of the total variation distance and its role in proving computational hardness. This material is standard, but it may be less familiar to those in numerical analysis.

Given two probability measures μ and ν on a common space, the *total variation (TV) distance* ([Canonne, 2022](#), Def. 1.1) is the maximum difference in probability between the two measures

$$D_{\text{TV}}(\mu, \nu) := \sup_{\mathcal{E}} |\mu(\mathcal{E}) - \nu(\mathcal{E})|.$$

As the name suggests, the TV distance is a metric on the set of probability measures. The TV distance quantifies how easy it is to determine which of two distributions a sample is drawn from:

Imported Theorem A.3 (Neyman–Pearson lemma; [Canonne \(2022, Lem. 1.4\)](#)) *Let μ and ν be probability distributions. Select one of these two distributions with equal probability, and draws a sample x from said distribution. Any algorithm that observes only x and guesses which distribution x was drawn from can succeed with probability at most $\frac{1}{2}(1 + D_{\text{TV}}(\mu, \nu))$.*

An important and elementary result is the *data processing inequality* ([Canonne, 2022](#), Fact 1.1), which states for any function f ,

$$D_{\text{TV}}(f(x), f(y)) \leq D_{\text{TV}}(x, y). \quad (\text{A.1})$$

That is, any way of processing the data (i.e., applying a function f) can only make it harder it is to distinguish a draw of the random variable x from a draw of the random variable y .

A final useful result is the *coupling lemma*. A joint distribution (x, y) is a coupling of μ and ν if x is marginally distributed as μ and y is marginally distributed as ν . Total variation exactly measures the maximum probability under which a coupling can have $x = y$:

Imported Lemma A.4 (Couplings lemma; Levin and Peres (2017, Prop. 4.7)) *Let μ and ν be probability distributions. Then,*

$$D_{\text{TV}}(\mu, \nu) = \inf \{ \mathbb{P}\{x \neq y\} : (x, y) \text{ is a coupling of } \mu \text{ and } \nu \}.$$

A.2. Ingredient 1: Arbitrary matrix–vector algorithms can’t beat randomized block Krylov

An important class of matrix–vector algorithms are randomized block Krylov algorithms, which generate random vectors and perform matvecs exclusively in order to build (a subset of) the *block Krylov sequence*

$$\text{BlkKrylov}(\mathbf{A}; b, t) := (\mathbf{A}^j \mathbf{g}_i : 1 \leq i \leq b, 0 \leq j \leq t) \quad \text{for } \mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (\text{A.2})$$

The parameter b is called the *block size* and t is called the *depth* of the Krylov subspace. Randomized block Krylov algorithms⁵ have been shown to be optimal for many tasks in linear algebra (Simchowitz et al., 2018; Braverman et al., 2020; Meyer et al., 2021; Bakshi and Narayanan, 2023), and they are believed to be optimal for many more (Bakshi et al., 2022; Bakshi and Narayanan, 2023). The starting point for this analysis is the following powerful theorem of Chewi et al. (2024) which shows that, when working with a rotationally invariant random matrix, an arbitrary matrix–vector algorithm performing ℓ matvecs can be simulated using a randomized block Krylov algorithm with block size ℓ and depth $\ell + 1$.

Imported Theorem A.5 (Reduction to block Krylov; Chewi et al. (2024, Lem. 5.16)) *Let $\mathbf{U} \in \mathbb{R}^{n \times n}$ be a Haar-random orthogonal matrix and $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix, and set $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$. Suppose that an adaptive deterministic algorithm \mathcal{A} computes matrix–vector products $\mathbf{Ax}_1, \dots, \mathbf{Ax}_\ell$ where $\ell^2 < n$. Let $\mathbf{g}_1, \dots, \mathbf{g}_\ell \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and introduce the triangular block Krylov sequence*

$$\text{TriBlkKrylov}(\mathbf{A}; \ell) := (\mathbf{A}^j \mathbf{g}_i : i \geq 1, j \geq 0, i + j \leq \ell + 1).$$

Then there exists a deterministic map F , depending only on the algorithm \mathcal{A} , such that

$$F(\text{TriBlkKrylov}(\mathbf{A}; \ell)) \stackrel{\text{dist}}{=} (\mathbf{Ax}_i : 1 \leq i \leq \ell).$$

This result shows that, in distribution, any adaptive deterministic sequence of ℓ matrix–vector products can be obtained from the *triangular* block Krylov sequence $\text{TriBlkKrylov}(\mathbf{A}; \ell)$, which is a subset of the traditional block Krylov sequence $\text{BlkKrylov}(\mathbf{A}; \ell, \ell + 1)$ defined above in (A.2).

A.3. Ingredient 2: From block Krylov to bilinear power sequences

Our strategy will be to generate two matrices \mathbf{A} and $\tilde{\mathbf{A}}$ for which the triangular block Krylov sequences are close in total variation (TV) distance, but for which $\text{tr}(f(\mathbf{A}))$ and $\text{tr}(f(\tilde{\mathbf{A}}))$ are far apart. To aid in this task, we introduce another auxiliary object, the *bilinear power sequence*

$$\text{BilinearPower}(\mathbf{A}; b, t) := (\mathbf{g}_i^\top \mathbf{A}^j \mathbf{g}_{i'} : 1 \leq i, i' \leq b \text{ and } 0 \leq j \leq t) \quad \text{for } \mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (\text{A.3})$$

The following lemma shows that if the bilinear power sequences of diagonal matrices \mathbf{D} and $\tilde{\mathbf{D}}$ are close in TV distance, then the block Krylov sequences of random rotations $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ and $\tilde{\mathbf{A}} = \mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$ generated by a Haar-random orthogonal matrix \mathbf{U} will be close as well.

5. Note that this definition of randomized block Krylov algorithms is broader than standard usage, including as special cases both randomized *single-vector* Krylov algorithms ($b = 1$) and non-adaptive randomized algorithms ($t = 1$).

Imported Lemma A.6 (Reduction to bilinear powers; similar to Chewi et al. (2024, Lem. 5.8))

Let $\mathbf{D}, \tilde{\mathbf{D}}$ be diagonal matrices, and fix integers $b, t \in \mathbb{N}$. In addition, let \mathbf{U} be a Haar-random orthogonal matrix, and set $\mathbf{A} := \mathbf{U}\mathbf{D}\mathbf{U}^\top$, $\tilde{\mathbf{A}} := \mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$. Then

$$\begin{aligned} & D_{\text{TV}}(\text{BlkKrylov}(\mathbf{A}; b, t), \text{BlkKrylov}(\tilde{\mathbf{A}}; b, t)) \\ & \leq D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, 2t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, 2t)). \end{aligned} \quad (\text{A.4})$$

Proof Let γ denote the right-hand side of (A.4). By the coupling lemma (Imported Lemma A.4), there exist a coupling between standard Gaussian vectors $\mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\tilde{\mathbf{g}}_1, \dots, \tilde{\mathbf{g}}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ such that, with probability $1 - \gamma$,

$$\mathbf{g}_i^\top \mathbf{D}^j \mathbf{g}_{i'} = \tilde{\mathbf{g}}_i \tilde{\mathbf{D}}^j \tilde{\mathbf{g}}_{i'} \quad \text{for all } i, i' \in [b] \text{ and } j \in \{0, \dots, 2t\}.$$

This condition implies that

$$\langle \mathbf{D}^j \mathbf{g}_i, \mathbf{D}^{j'} \mathbf{g}_{i'} \rangle = \langle \tilde{\mathbf{D}}^j \tilde{\mathbf{g}}_i, \tilde{\mathbf{D}}^{j'} \tilde{\mathbf{g}}_{i'} \rangle \quad \text{for all } i, i' \in [b] \text{ and } j, j' \in \{0, \dots, t\}.$$

This above condition asserts that the $\mathbf{D}^j \mathbf{g}_i$ vectors are merely a rotation of the $\tilde{\mathbf{D}}^j \tilde{\mathbf{g}}_i$ vectors. That is, for some orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$, it holds that $\mathbf{D}^j \mathbf{g}_i = \mathbf{Q} \cdot \tilde{\mathbf{D}}^j \tilde{\mathbf{g}}_i$ for all i and j . Next, we define $\mathbf{z}_i := \mathbf{U} \mathbf{g}_i$ and $\tilde{\mathbf{z}}_i := \mathbf{U} \mathbf{Q} \tilde{\mathbf{g}}_i$, which are both distributed as $\mathcal{N}(\mathbf{0}, \mathbf{I})$ vectors, and set $\mathbf{A} := \mathbf{U}\mathbf{D}\mathbf{U}^\top$ and $\tilde{\mathbf{A}} := \mathbf{U}\mathbf{Q}\tilde{\mathbf{D}}\mathbf{Q}^\top\mathbf{U}^\top$ for a Haar-random matrix \mathbf{A} . The matrix $\tilde{\mathbf{A}}$ has the same distribution as $\mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$, and it is independent of $\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_b$. Further, with probability at least $1 - \gamma$,

$$\mathbf{A}^j \mathbf{z}_i = \mathbf{U} \mathbf{D}^j \mathbf{U}^\top \mathbf{U} \mathbf{g}_i = \mathbf{U} \mathbf{D}^j \mathbf{g}_i = \mathbf{U} \mathbf{Q} \tilde{\mathbf{D}}^j \tilde{\mathbf{g}}_i = \mathbf{U} \mathbf{Q} \tilde{\mathbf{D}}^j \mathbf{Q}^\top \mathbf{U}^\top \cdot \mathbf{U} \mathbf{Q} \tilde{\mathbf{g}}_i = \tilde{\mathbf{A}}^j \tilde{\mathbf{z}}_i.$$

Therefore, we have manifested a coupling under which $\text{BlkKrylov}(\mathbf{A}, b, t)$ and $\text{BlkKrylov}(\tilde{\mathbf{A}}, b, t)$ are equal with probability $\geq 1 - \gamma$. The comparison (A.4) follows by Imported Lemma A.4. \blacksquare

Combining the ingredients to this point, we obtain a general hardness result.

Corollary A.7 (Hardness from bilinear powers; similar to Chewi et al. (2024, Lem. 5.8)) Let \mathbf{D} and $\tilde{\mathbf{D}}$ be diagonal matrices, fix parameters $b, t \in \mathbb{N}$, and suppose that the bilinear power sequences are close in TV distance:

$$D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \leq 1/6.$$

Let \mathbf{U} be a Haar-random orthogonal matrix, and define $\mathbf{A} := \mathbf{U}\mathbf{D}\mathbf{U}^\top$ and $\tilde{\mathbf{A}} := \mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$. Any matrix–vector algorithm that can distinguish \mathbf{A} from $\tilde{\mathbf{A}}$ with probability at least $2/3$ must compute at least $\min\{b - 1, \lfloor t/2 \rfloor\} \gtrsim \min\{b, t\}$ matvecs.

Proof By Yao’s minimax principle (Motwani and Raghavan, 1995, Sec. 2.2), we can restrict our attention to deterministic algorithms. Consider any matrix–vector algorithm which can make ℓ matvecs, and let $\mathcal{T} = (\mathbf{A}\mathbf{x}_i : 1 \leq i \leq \ell)$ and $\tilde{\mathcal{T}} = (\tilde{\mathbf{A}}\tilde{\mathbf{x}}_i : 1 \leq i \leq \ell)$ denote transcripts of matvecs produced by the algorithm on \mathbf{A} and $\tilde{\mathbf{A}}$, respectively. By the data processing inequality (A.1), the reduction to triangular block Krylov (Imported Theorem A.5), the observation that triangular block

Krylov is a subset of a general block Krylov method, and the reduction to bilinear power sequences (Imported Lemma A.6), it holds that

$$\begin{aligned} D_{\text{TV}}(\mathcal{T}, \tilde{\mathcal{T}}) &\leq D_{\text{TV}}(\text{TriBlkKrylov}(\mathbf{A}; \ell), \text{TriBlkKrylov}(\tilde{\mathbf{A}}; \ell)) && \text{(Imported Theorem A.5)} \\ &\leq D_{\text{TV}}(\text{BlkKrylov}(\mathbf{A}; \ell + 1, \ell), \text{BlkKrylov}(\tilde{\mathbf{A}}; \ell + 1, \ell)) && \text{(Data Processing)} \\ &\leq D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; \ell + 1, 2\ell), \text{BilinearPower}(\tilde{\mathbf{D}}; \ell + 1, 2\ell)). && \text{(Imported Lemma A.6)} \end{aligned}$$

Choose $\ell := \min\{b - 1, \lfloor t/2 \rfloor\}$. By the hypothesis, the right-hand side is bounded by a sufficiently small positive constant. Ergo, $D_{\text{TV}}(\mathcal{T}, \tilde{\mathcal{T}}) < 1/6$, and we conclude by Imported Theorem A.3 that any matrix–vector algorithm for distinguishing between \mathbf{A} and $\tilde{\mathbf{A}}$ using ℓ matvecs must succeed with probability less than $2/3$. \blacksquare

A.4. Ingredient 3: Constructing a finite set of eigenvalues

To complete the proof, all we need is two diagonal matrices \mathbf{D} and $\tilde{\mathbf{D}}$ for which the bilinear power sequences (A.3) are similar (in TV distance) but where the traces $\text{tr}(f(\mathbf{D}))$ and $\text{tr}(f(\tilde{\mathbf{D}}))$ are far apart. We need to construct these matrices by only using the fact that f is inapproximable on some domain \mathcal{S} . Since \mathcal{S} may have infinitely many entries, our first step is to discretize \mathcal{S} into a much smaller finite set, from which we will extract the eigenvalues of \mathbf{D} and $\tilde{\mathbf{D}}$.

Imported Theorem A.8 (Inapproximability on a finite set; Rivlin (1981, Thm. 1.14)) *Let f be continuous and (t, α) -inapproximable on a bounded set $\mathcal{S} \subseteq \mathbb{R}$. Then there exists values*

$$\lambda_1, \dots, \lambda_{t+2} \in \mathcal{S}$$

such that f is inapproximable on $\mathcal{S}' := \{\lambda_1, \dots, \lambda_{t+2}\}$.

Strictly speaking, Rivlin (1981) treats only the case when \mathcal{S} is an interval or a finite set, but the same proof carries over without issue to general bounded sets \mathcal{S} provided one uses an appropriately general version of the Chebyshev equioscillation theorem (DeVore and Lorentz, 1993, Thm. 5.1).

A.5. Ingredient 4: Constructing eigenvalue multiplicities

To complete the proof, all we need is two diagonal matrices \mathbf{D} and $\tilde{\mathbf{D}}$ whose bilinear power sequences (A.3) are similar in TV distance, but whose spectral sums $\text{tr}(f(\mathbf{D}))$ and $\text{tr}(f(\tilde{\mathbf{D}}))$ are far apart. To do so, we shall take the values $\lambda_1, \dots, \lambda_{t+2}$ generated in the last section and construct \mathbf{D} and $\tilde{\mathbf{D}}$ by including each number λ_i with an appropriate multiplicity.

We construct these multiplicities in steps. First, in Lemma A.9, we use the inapproximability property to find fractional multiplicities which look similar to polynomial functions, but distinguish the function f . Then, in Theorem A.10, we modify these multiplicities slightly to ensure other desirable properties, such as ensuring that each number λ_i receives some non-negligible mass and that the multiplicities for \mathbf{D} and $\tilde{\mathbf{D}}$ are close in relative distance. Finally, in Construction A.11, we round these fractional multiplicities to integers. We begin with this program below by constructing fractional multiplicities that distinguish the function f .

Lemma A.9 (Eigenvalue multiplicities I; variant of Chewi et al. (2024, Prop. 5.5)) *Suppose the function f is (t, α) -inapproximable on $\{\lambda_1, \dots, \lambda_{t+2}\}$, and fix $n \geq 1$. Then, there exists non-negative weights x_1, \dots, x_{t+2} and x'_1, \dots, x'_{t+2} such that*

(a) **Mass.** The total mass is $\sum_{i=1}^{t+2} x_i = \sum_{i=1}^{t+2} \tilde{x}_i = n$.

(b) **Not distinguished by polynomials.** For $j \in [t]$, it holds that $\sum_{i=1}^{t+2} x_i \lambda_i^j = \sum_{i=1}^{t+2} \tilde{x}_i \lambda_i^j$.

(c) **Distinguished by f .** It holds that $\sum_{i=1}^{t+2} x_i f(\lambda_i) - \sum_{i=1}^{t+2} \tilde{x}_i f(\lambda_i) \geq 2\alpha n \cdot \max_{i \in [t+2]} |f(\lambda_i)|$.

Proof The proof is by linear programming duality. Set L to be the maximum value of $\sum_{i=1}^{t+2} x_i f(\lambda_i) - \sum_{i=1}^{t+2} \tilde{x}_i f(\lambda_i)$ over all $\{x_i\}$ and $\{\tilde{x}_i\}$ satisfying (a) and (b). The constraints (a) and (b) are linear, so L is the optimal value of the following linear program:

$$\max_{\bar{\mathbf{x}} \in \mathbb{R}^{2(t+2)}} \mathbf{c}^\top \bar{\mathbf{x}} \quad \text{subject to } \mathbf{M}\bar{\mathbf{x}} = \mathbf{b}, \bar{\mathbf{x}} \geq 0,$$

where

$$\bar{\mathbf{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_{t+2} \\ x'_1 \\ \vdots \\ x'_{t+2} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} f(\lambda_1) \\ \vdots \\ f(\lambda_{t+2}) \\ -f(\lambda_1) \\ \vdots \\ -f(\lambda_{t+2}) \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & -1 & \cdots & -1 \\ \lambda_1 & \cdots & \lambda_{t+2} & -\lambda_1 & \cdots & -\lambda_{t+2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_1^t & \cdots & \lambda_{t+2}^t & -\lambda_1^t & \cdots & -\lambda_{t+2}^t \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2n \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Let $\mathbf{V} = [\lambda_i^j]_{i \in [t+2], j \in \{0, \dots, t\}} \in \mathbb{R}^{(t+2) \times (t+2)}$ be the Vandermonde matrix of degree t defined on the nodes $\lambda_1, \dots, \lambda_{t+2}$, and let $\mathbf{e} \in \mathbb{R}^{t+2}$ be the all-ones vector, so that $\mathbf{M}^\top = \begin{bmatrix} \mathbf{e} & \mathbf{V} \\ \mathbf{e} & -\mathbf{V} \end{bmatrix}$. Then, the dual problem is

$$\min_{\mathbf{y} \in \mathbb{R}^{t+2}} \mathbf{b}^\top \mathbf{y} \quad \text{subject to } \mathbf{M}^\top \mathbf{y} \geq \mathbf{c}.$$

We partition $\mathbf{y} = \begin{bmatrix} w \\ \mathbf{y}_{\text{bot}} \end{bmatrix}$ so that $[\mathbf{V}\mathbf{y}_{\text{bot}}]_i = p(\lambda_i)$ for some polynomial p with coefficients determined by \mathbf{y}_{bot} . The constraint of the dual problem then becomes

$$\mathbf{M}^\top \mathbf{y} = \begin{bmatrix} \mathbf{e} & \mathbf{V} \\ \mathbf{e} & -\mathbf{V} \end{bmatrix} \begin{bmatrix} w \\ \mathbf{y}_{\text{bot}} \end{bmatrix} = \begin{bmatrix} w + p(\lambda_1) \\ \vdots \\ w + p(\lambda_{t+2}) \\ w - p(\lambda_1) \\ \vdots \\ w - p(\lambda_{t+2}) \end{bmatrix} \geq \begin{bmatrix} f(\lambda_1) \\ \vdots \\ f(\lambda_{t+2}) \\ -f(\lambda_1) \\ \vdots \\ -f(\lambda_{t+2}) \end{bmatrix}.$$

Notice that the inequalities above are equivalent to saying that $|p(\lambda_i) - f(\lambda_i)| \leq w$ for all i . Next, since $\mathbf{b}^\top \mathbf{y} = 2nw$, we can rewrite the dual problem as an optimization over polynomials p and the number w :

$$\min_{\mathbf{M}^\top \mathbf{y} \geq \mathbf{c}} \mathbf{b}^\top \mathbf{y} = \min_{\substack{\deg(p) \leq t \\ |p(\lambda_i) - f(\lambda_i)| \leq w}} 2nw = \min_{\deg(p) \leq t} \max_{i \in [t+2]} |p(\lambda_i) - f(\lambda_i)| \geq 2n\alpha \cdot \max_{i \in [t+2]} |f(\lambda_i)|.$$

The inequality is inapproximability of the function f . By weak duality, we conclude that $L \geq 2n\alpha \cdot \max_{i \in [t+2]} |f(\lambda_i)|$, as desired. \blacksquare

Next, we modify the multiplicities to ensure they are uniformly lower-bounded and close in relative distance.

Corollary A.10 (Eigenvalue multiplicities II; variant of Chewi et al. (2024, Cor. 5.6)) Fix a parameter $\eta \in (0, 1/2)$. Let f be (t, α) -inapproximable on $\{\lambda_1, \dots, \lambda_{t+2}\}$, and fix $n \geq 1$. Then, there exists non-negative weights y_1, \dots, y_{t+2} and y'_1, \dots, y'_{t+2} such that

- (a) **Mass.** The total mass is $\sum_{i=1}^{t+2} y_i = \sum_{i=1}^{t+2} \tilde{y}_i = n$.
- (b) **Not distinguished by polynomials.** For $j \in [t]$, it holds that $\sum_{i=1}^{t+2} y_i \lambda_i^j = \sum_{i=1}^{t+2} \tilde{y}_i \lambda_i^j$.
- (c) **Distinguished by f .** $\sum_{i=1}^{t+2} y_i f(\lambda_i) - \sum_{i=1}^{t+2} \tilde{y}_i f(\lambda_i) \geq \eta \alpha n \cdot \max_{i \in [t+2]} |f(\lambda_i)|$.
- (d) **Uniformly lower-bounded.** For every $i \in [t+2]$, it holds that $y_i \geq \frac{n}{2(t+2)}$ and $\tilde{y}_i \geq \frac{n}{2(t+2)}$.
- (e) **Close in relative distance.** For every $i \in [t+2]$, it holds that $|y_i - \tilde{y}_i| \leq 4\eta \cdot y_i$.

Proof Let $\{x_i\}$ and $\{\tilde{x}_i\}$ be the weights furnished by Lemma A.9, and set

$$w_i = \frac{1}{4} \left(x_i + \frac{n}{t+2} \right) \quad \text{and} \quad w'_i = \frac{1}{2} \left(\tilde{x}_i + \frac{n}{t+2} \right)$$

and

$$y_i = \frac{1+\eta}{2} w_i + \frac{1-\eta}{2} w'_i \quad \text{and} \quad \tilde{y}_i = \frac{1-\eta}{2} w_i + \frac{1+\eta}{2} w'_i.$$

The five desired properties are readily verified. ■

Using this corollary, we are now ready to define a pair of diagonal matrices which we will use to prove hardness.

Construction A.11 (Indistinguishable pair) Given a function f that is (t, α) -inapproximable on $\{\lambda_1, \dots, \lambda_{t+2}\}$ and an integer $n \in \mathbb{N}$, we construct a pair of diagonal matrices $\mathbf{D}, \tilde{\mathbf{D}} \in \mathbb{R}^{n \times n}$ that cannot be distinguished from their bilinear power sequences. Namely, let $\{y_i\}_{i=1}^{t+2}$ and $\{\tilde{y}_i\}_{i=1}^{t+2}$ be the weights furnished by Theorem A.10. Round each weight to integers $\{N_i\}_{i=1}^{t+2}$ and $\{\tilde{N}_i\}_{i=1}^{t+2}$ by taking either the floor or the ceiling, while ensuring that $\sum_{i=1}^{t+2} N_i = \sum_{i=1}^{t+2} \tilde{N}_i = n$. Finally, define

$$\mathbf{D} := \text{diag}(\underbrace{\lambda_1, \dots, \lambda_1}_{N_1 \text{ times}}, \dots, \underbrace{\lambda_{t+2}, \dots, \lambda_{t+2}}_{N_{t+2} \text{ times}}) \quad \text{and} \quad \tilde{\mathbf{D}} := \text{diag}(\underbrace{\lambda_1, \dots, \lambda_1}_{\tilde{N}_1 \text{ times}}, \dots, \underbrace{\lambda_{t+2}, \dots, \lambda_{t+2}}_{\tilde{N}_{t+2} \text{ times}}).$$

Finally, let \mathbf{U} be a Haar-random orthogonal matrix, and set $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ and $\tilde{\mathbf{A}} = \mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$.

A.6. Ingredient 5: From bilinear power sequences to Wishart matrices

Now, we need to bound the total variation distance between the bilinear power sequences sequences (A.3) for \mathbf{D} and $\tilde{\mathbf{D}}$. To do so, we use the following formula:

Definition A.12 (Wishart matrix) Let $\mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be independent Gaussian vectors of length N , and consider a $b \times b$ matrix \mathbf{W} with entries $[\mathbf{W}]_{ii'} := \langle \mathbf{g}_i, \mathbf{g}_{i'} \rangle$. Then the random matrix \mathbf{W} is said to follow Wishart distribution with N degrees of freedom. We write $\mathbf{W} \sim \text{WISHART}(b, N)$.

Proposition A.13 (Bilinear power sequences: Wishart formula) Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix constructed by repeating numbers $\lambda_1, \dots, \lambda_{t+2} \in \mathbb{R}$ with multiplicities $N_1 + \dots + N_{t+2} = n$, and let $\mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then there exist independent Wishart matrices

$$\mathbf{W}^{(1)} \sim \text{WISHART}(b, N_1), \dots, \mathbf{W}^{(t+2)} \sim \text{WISHART}(b, N_{t+2})$$

for which

$$\mathbf{g}_i^\top \mathbf{D}^q \mathbf{g}_{i'} = \sum_{j=1}^{t+2} \lambda_j^q [\mathbf{W}^{(j)}]_{ii'} \quad \text{for every } q \geq 0 \text{ and } 1 \leq i, i' \leq b.$$

Proof Let $\mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be a Gaussian vectors, and let $\mathbf{g}_i^{(j)} \in \mathbb{R}^{N_j}$ denote the restriction of \mathbf{g}_i to the dimensions corresponding to where eigenvalue λ_j appears in \mathbf{D} . Similarly, let $\tilde{\mathbf{g}}_i^{(j)}$ denote the restriction of \mathbf{g}_i to the positions in which eigenvalue λ_j appears in $\tilde{\mathbf{D}}$. Observe that

$$\mathbf{g}_i^\top \mathbf{D}^q \mathbf{g}_{i'} = \sum_{j=1}^{t+2} \lambda_j^q \langle \mathbf{g}_i^{(j)}, \mathbf{g}_{i'}^{(j)} \rangle \quad \text{for every } q \geq 0 \text{ and } 1 \leq i, i' \leq b.$$

In particular, we conclude that bilinear power sequences of \mathbf{D} only depend on the *Gram matrices*

$$\mathbf{W}^{(j)} := \left[\langle \mathbf{g}_i^{(j)}, \mathbf{g}_{i'}^{(j)} \rangle \right]_{1 \leq i, i' \leq b} \sim \text{WISHART}(b, N_j) \quad \text{for } j = 1, \dots, t+2.$$

of the collection of restricted Gaussian vectors $\mathbf{g}_i^{(j)}$, which follow a Wishart distribution. The stated conclusion follows. \blacksquare

This result shows us that the bilinear power sequence for a diagonal matrix \mathbf{D} is generated by a family of Wishart matrices $\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(t+2)}$ with degrees of freedom N_1, \dots, N_{t+2} . As such, given two diagonal matrices \mathbf{D} and $\tilde{\mathbf{D}}$ with corresponding Wishart matrices $\{\mathbf{W}^{(j)}\}$ and $\{\tilde{\mathbf{W}}^{(j)}\}$, it is tempting to try directly bounding the TV distance between the bilinear power sequences by the TV distance between the families of Wishart matrices, i.e.,

$$\begin{aligned} & D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \\ & \leq D_{\text{TV}}((\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(t+2)}), (\tilde{\mathbf{W}}^{(1)}, \dots, \tilde{\mathbf{W}}^{(t+2)})). \end{aligned} \quad (\text{A.5})$$

But there is a problem: As the families $\{\mathbf{W}^{(j)}\}$ and $\{\tilde{\mathbf{W}}^{(j)}\}$ have different numbers of degrees of freedom, the TV distances between these sequences can be large. As such, the inequality (A.5) is not powerful enough for our purposes. Fortunately, a simple variation of the trick works.

Corollary A.14 (From bilinear power sequences to Wishart matrices) *Construct diagonal matrices $\mathbf{D}, \mathbf{D}' \in \mathbb{R}^{n \times n}$ via [Construction A.11](#), and define independent Wishart matrices*

$$\mathbf{W}^{(i)} \sim \text{WISHART}(b, N_i) \quad \text{and} \quad \tilde{\mathbf{W}}^{(i)} \sim \text{WISHART}(b, \tilde{N}_i) \quad \text{for } i = 1, \dots, t+2.$$

and define $\widehat{\mathbf{W}}^{(i)} = \tilde{\mathbf{W}}^{(i)} + (y_i - \tilde{y}_i)\mathbf{I}$. Then,

$$\begin{aligned} & D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \\ & \leq D_{\text{TV}}((\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(t+2)}), (\widehat{\mathbf{W}}^{(1)}, \dots, \widehat{\mathbf{W}}^{(t+2)})). \end{aligned} \quad (\text{A.6})$$

Proof Introduce Gaussian vectors $\mathbf{g}_1, \dots, \mathbf{g}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By [Proposition A.13](#), there exist independent Wishart matrices $\tilde{\mathbf{W}}^{(1)} \sim \text{WISHART}(b, \tilde{N}_1), \dots, \tilde{\mathbf{W}}^{(t+2)} \sim \text{WISHART}(b, \tilde{N}_{t+2})$ such that

$$\mathbf{g}_i^\top \tilde{\mathbf{D}}^q \mathbf{g}_{i'} = \sum_{j=1}^{t+2} \lambda_j^q \tilde{\mathbf{W}}_{ii'}^{(j)} \quad \text{for } q \geq 0 \text{ and } i, i' \in [b].$$

By properties (a) and (b) of the weights $\{y_i\}$ and $\{\tilde{y}_i\}$ generated by [Theorem A.10](#), it holds that

$$\sum_{j=1}^{t+2} \lambda_j^q (y_j - \tilde{y}_j) = 0 \quad \text{for } q \in \{0, \dots, t\}.$$

Combining the two previous displays, we obtain that

$$\mathbf{g}_i^\top \tilde{\mathbf{D}}^q \mathbf{g}_{i'} = \sum_{j=1}^{t+2} \lambda_j^q [\tilde{\mathbf{W}}^{(j)} + (y_j - \tilde{y}_j) \mathbf{I}]_{ii'} \quad \text{for } q \in \{0, \dots, t\} \text{ and } i, i' \in [b].$$

The conclusion follows by [Proposition A.13](#) and the data processing inequality [\(A.1\)](#). \blacksquare

The inequality [\(A.6\)](#) is more useful than [\(A.5\)](#) since the shifted Wishart matrices $\widehat{\mathbf{W}}^{(j)}$ to have nearly the same mean as their cousins $\mathbf{W}^{(j)}$. In particular, the means

$$\mathbb{E}[\widehat{\mathbf{W}}^{(j)}] = (y_j - \tilde{y}_j + \tilde{N}_j) \mathbf{I} \quad \text{and} \quad \mathbb{E}[\mathbf{W}^{(j)}] = N_j \mathbf{I}$$

differ by at most

$$|(y_j - \tilde{y}_j + \tilde{N}_j) - N_j| \leq |y_j - N_j| + |\tilde{y}_j - \tilde{N}_j| \leq 2. \quad (\text{A.7})$$

In the final inequality, we recognize that N_j and \tilde{N}_j are obtained by rounding y_j and \tilde{y}_j up or down to the nearest integer.

A.7. Ingredient 6: Total variation distance between Wishart matrices

To instantiate [Theorem A.14](#), we need to bound the total variation distance between two Wishart matrices. We state a general version of such a result here:

Lemma A.15 (Total variation distance between Wishart matrices) *Let*

$$\mathbf{W} \sim \text{WISHART}(b, N) \quad \text{and} \quad \tilde{\mathbf{W}} \sim \text{WISHART}(b, \tilde{N})$$

be Wishart matrices of common dimension b and degrees of freedom N and \tilde{N} . For any $\beta \in \mathbb{R}$, it holds that

$$D_{\text{TV}}(\mathbf{W}, \tilde{\mathbf{W}} + \beta \mathbf{I}) \lesssim b \cdot \frac{|\tilde{N} + \beta - N|}{\sqrt{N}} + b^{3/2} \cdot \frac{1}{\sqrt{N}} + b^2 \cdot \frac{|\tilde{N} - N|}{N}$$

A bound similar to this appears in the proof of ([Chewi et al., 2024](#), Lem. 5.7). To prove this lemma, we employ the following results:

Imported Lemma A.16 (Wishart vs. GOE; Meyer (2026, Thm. 1.3)) *Let $\mathbf{Y} := N\mathbf{I} + \sqrt{2N}\mathbf{G}$ be a shift and scaling of the GOE matrix⁶ $\mathbf{G} \sim \text{GOE}(b)$, and let $\mathbf{W} \sim \text{WISHART}(b, N)$ be Wishart. Then, the total variation between \mathbf{W} and \mathbf{Y} is $\lesssim b^{3/2}/N^{1/2}$.*

6. Recall that an unnormalized GOE matrix $\mathbf{G} \sim \text{GOE}(b)$ is a random symmetric $b \times b$ matrix where the entries on the upper triangle ($[\mathbf{G}]_{ii'} : i \leq i'$) are independent, mean-zero Gaussian random variables. The diagonal entries have variance 2, and the off-diagonal entries have variance 1.

Imported Lemma A.17 (TV between Gaussians; Devroye et al. (2022, Thm. 1.3)) *The TV distance between the one-dimensional Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ is at most*

$$D_{\text{TV}}(\mathcal{N}(\mu_1, \sigma_1^2), \mathcal{N}(\mu_2, \sigma_2^2)) \lesssim \frac{|\sigma_2^2 - \sigma_1^2|}{\sigma_1^2} + \frac{|\mu_1 - \mu_2|}{\sigma_1}.$$

With these ingredients in place, we prove [Lemma A.15](#).

Proof of Lemma A.15 Draw shifted GOE matrices

$$\mathbf{Y} \sim N\mathbf{I} + \sqrt{2N} \text{GOE}(b) \quad \text{and} \quad \tilde{\mathbf{Y}} \sim \tilde{N}\mathbf{I} + \sqrt{2\tilde{N}} \text{GOE}(b).$$

By [Imported Lemma A.16](#), the triangle inequality, and the data processing inequality [\(A.1\)](#),

$$\begin{aligned} D_{\text{TV}}(\mathbf{W}, \tilde{\mathbf{W}} + \beta\mathbf{I}) &\leq D_{\text{TV}}(\mathbf{W}, \mathbf{Y}) + D_{\text{TV}}(\mathbf{Y}, \tilde{\mathbf{Y}} + \beta\mathbf{I}) + D_{\text{TV}}(\tilde{\mathbf{Y}}, \tilde{\mathbf{W}}) \\ &\lesssim D_{\text{TV}}(\mathbf{Y}, \tilde{\mathbf{Y}} + \beta\mathbf{I}) + \frac{b^{3/2}}{N^{1/2}}. \end{aligned} \tag{A.8}$$

The upper triangles of the matrices \mathbf{Y} and $\tilde{\mathbf{Y}} + \beta\mathbf{I}$ have independent entries, so we may bound the total variation distance by summing over the matrix entries

$$\begin{aligned} D_{\text{TV}}(\mathbf{Y}, \tilde{\mathbf{Y}} + \beta\mathbf{I}) &\leq \sum_{1 \leq i \leq i' \leq b} D_{\text{TV}}([\mathbf{Y}]_{ii'}, [\tilde{\mathbf{Y}} + \beta\mathbf{I}]_{ii'}) \\ &\lesssim b D_{\text{TV}}(\mathcal{N}(N, 4N), \mathcal{N}(\tilde{N} + \beta, 4\tilde{N})) + b^2 D_{\text{TV}}(\mathcal{N}(0, 2N), \mathcal{N}(0, 2\tilde{N})). \end{aligned}$$

The first term comes from the b diagonal entries, and the second term comes from the $\binom{b}{2} \lesssim b^2$ entries on the strictly upper triangular portion of the matrix. Applying [Imported Lemma A.17](#), we conclude that

$$\begin{aligned} D_{\text{TV}}(\mathbf{Y}, \tilde{\mathbf{Y}} + \beta\mathbf{I}) &\lesssim b \cdot \left[\frac{|\tilde{N} - N|}{N} + \frac{|\tilde{N} + \beta - N|}{\sqrt{N}} \right] + b^2 \cdot \frac{|\tilde{N} - N|}{N} \\ &\asymp b \cdot \frac{|\tilde{N} + \beta - N|}{\sqrt{N}} + b^2 \cdot \frac{|\tilde{N} - N|}{N}. \end{aligned}$$

Combining with [\(A.8\)](#) yields the stated conclusion. \blacksquare

A.8. Putting it all together

Let us now combine all of the ingredients we have assembled to prove [Theorem 2.4](#). We begin with one final lemma.

Lemma A.18 (Hard-to-distinguish matrices; variant of Chewi et al. (2024, Lem. 5.7)) *Fix parameter $\eta \in (0, 1)$, a function f that is (t, α) -inapproximable on $\{\lambda_1, \dots, \lambda_{t+2}\}$, and a size $n \in \mathbb{N}$. Let \mathbf{D} and $\tilde{\mathbf{D}}$ by the matrices furnished by [Construction A.11](#). Then*

- (a) *Distinguished by f .* $|\text{tr}(f(\mathbf{D})) - \text{tr}(f(\tilde{\mathbf{D}}))| \geq (\eta\alpha n - 2(t+2)) \cdot \max_{i \in [t+2]} |f(\lambda_i)|$.
- (b) *Bilinear power sequences are close.* It holds that

$$D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \lesssim \frac{b^{3/2}t^{3/2}}{n^{1/2}} + \eta b^2 t.$$

Proof Property (a) is easily verified from [Theorem A.10\(c\)](#). We now verify (b). By [Theorem A.14](#), we have

$$\begin{aligned} & D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \\ & \leq D_{\text{TV}}((\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(t+2)}), (\tilde{\mathbf{W}}^{(1)} + (y_1 - \tilde{y}_1)\mathbf{I}, \dots, \tilde{\mathbf{W}}^{(t+2)} + (y_{t+2} - \tilde{y}_{t+2})\mathbf{I})) \\ & \leq \sum_{j=1}^{t+2} D_{\text{TV}}(\mathbf{W}^{(j)}, \tilde{\mathbf{W}}^{(j)} + (y_j - \tilde{y}_j)\mathbf{I}). \end{aligned}$$

Invoking [Lemma A.15](#), we deduce the bound

$$\begin{aligned} & D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \\ & \lesssim \sum_{j=1}^{t+2} \left[b \frac{|(y_j - \tilde{y}_j + \tilde{N}_j) - N_j| + \sqrt{b}}{\sqrt{\tilde{N}_j}} + b^2 \frac{|\tilde{N}_j - N_j|}{N_j} \right]. \end{aligned}$$

We bound the first term by invoking the maximum mean difference bound [\(A.7\)](#) to control the numerator and property [Theorem A.10\(d\)](#) to control the denominator as $N_j \gtrsim n/t$, obtaining

$$\frac{|(y_j - \tilde{y}_j + \tilde{N}_j) - N_j| + \sqrt{b}}{\sqrt{\tilde{N}_j}} \lesssim \frac{b^{1/2}t^{1/2}}{n^{1/2}}.$$

To bound the second term, observe that [Theorem A.10\(d\)](#) and [Theorem A.10\(e\)](#) give us

$$\frac{|\tilde{N}_j - N_j|}{N_j} \leq \frac{|\tilde{N}_j - \tilde{y}_j| + |\tilde{y}_j - y_j| + |N_j - y_j|}{y_j - 1} \lesssim \frac{t}{n} + \eta.$$

Combining this bound with the previous display, we obtain

$$D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \lesssim \frac{b^{3/2}t^{3/2}}{n^{1/2}} + \frac{b^2t^2}{n} + \eta b^2t. \quad (\text{A.9})$$

We end the proof by observing that we can drop the second term in [\(A.9\)](#). There are two cases: Either $n \leq bt$ or $n > bt$. In the former case, $b^{3/2}t^{3/2}/n^{1/2} \geq bt \geq 1$, and the bound is vacuous since the TV distance is at most 1. Therefore, in this case, the second term can be dropped. In the latter case, it holds that $b^2t^2/n < b^{3/2}t^{3/2}/n^{1/2}$, and the second term is dominated by the first. ■

Proof of [Theorem A.1](#) Let $\{\lambda_1, \dots, \lambda_{t+2}\}$ be the eigenvalues generated by [Imported Theorem A.8](#), and instate \mathbf{D} , $\tilde{\mathbf{D}}$, \mathbf{A} , and $\tilde{\mathbf{A}}$ as generated by [Construction A.11](#). Set $b := t$ and $\eta := ct^{-3}$ for a sufficiently small absolute constant $c > 0$. By the hypothesis $n \gtrsim t^6$ and [Lemma A.18\(b\)](#),

$$D_{\text{TV}}(\text{BilinearPower}(\mathbf{D}; b, t), \text{BilinearPower}(\tilde{\mathbf{D}}; b, t)) \lesssim \frac{t^3}{n^{0.5}} + \eta t^3$$

is smaller than the value $1/6$ needed to activate [Theorem A.7](#). Thus, by [Theorem A.7](#), at least $\min\{t-1, \lfloor t/2 \rfloor\} = \lfloor t/2 \rfloor$ matvecs are needed to distinguish the matrices \mathbf{A} and $\tilde{\mathbf{A}}$ with probability at least $2/3$.

To finish the proof, we must show that any method able to computing $\text{tr}(f(\mathbf{A}))$ to the prescribed accuracy can distinguish \mathbf{A} from $\tilde{\mathbf{A}}$. First, note that

$$\|f(\mathbf{A})\|_{\mathbb{F}}^2 = \sum_{i=1}^{t+2} N_i |f(\lambda_i)|^2 \leq \left(\sum_{i=1}^{t+2} N_i \right) \cdot \max_{i \in [t+2]} |f(\lambda_i)|^2 = n \cdot \max_{i \in [t+2]} |f(\lambda_i)|^2.$$

Combining this with [Lemma A.18\(a\)](#), we obtain

$$|\text{tr}(f(\mathbf{A})) - \text{tr}(f(\tilde{\mathbf{A}}))| \geq \left(ct^{-3}\alpha n - 2(t+2) \right) \max_{i \in [t+2]} |f(\lambda_i)| \geq \left(ct^{-3}\alpha n - 2(t+2) \right) \frac{\|f(\mathbf{A})\|_{\mathbb{F}}}{\sqrt{n}}.$$

By the hypothesis $n \gtrsim t^4/\alpha$, the first term dominates the second, and we conclude that

$$|\text{tr}(f(\mathbf{A})) - \text{tr}(f(\tilde{\mathbf{A}}))| \gtrsim \frac{\alpha\sqrt{n}}{t^3} \|f(\mathbf{A})\|_{\mathbb{F}}.$$

Therefore, any algorithm that can compute $\tilde{\text{tr}}$ such that $|\tilde{\text{tr}} - \text{tr}(f(\mathbf{A}))| \lesssim \frac{\alpha\sqrt{n}}{t^3} \|\mathbf{A}\|_{\mathbb{F}}$ can distinguish \mathbf{A} from $\tilde{\mathbf{A}}$. We know this task requires $\lfloor t/2 \rfloor$ matvecs, completing the proof. \blacksquare

Appendix B. Proof of the hidden Haar theorem

We let $\mathbb{O}(n)$ denote the set of $n \times n$ orthogonal matrices and $\text{HAAR}(\mathbb{O}(n))$ denote the Haar distribution over $\mathbb{O}(n)$. The defining property of the Haar distribution is rotational invariance:

If $\mathbf{Q} \sim \text{HAAR}(\mathbb{O}(n))$ and $\mathbf{U} \in \mathbb{O}(n)$ is deterministic, then $\mathbf{Q}\mathbf{U} \sim \text{HAAR}(\mathbb{O}(n))$.

Rotational invariance can extend to cases in which \mathbf{U} is not deterministic. Indeed, suppose $\mathbf{U} \in \mathbb{O}(n)$ is *random* but independent of \mathbf{Q} . Then, conditioned on the realization of \mathbf{U} , we observe that $\mathbf{Q}\mathbf{U} \sim \text{HAAR}(\mathbb{O}(n))$. Since the conditional distribution of $\mathbf{Q}\mathbf{U}$ does not depend on \mathbf{U} itself, we infer the following *extended* form of the rotational invariance property:

Fact B.1 (Extended rotational invariance) *If $\mathbf{Q} \sim \text{HAAR}(\mathbb{O}(n))$ and $\mathbf{U} \in \mathbb{O}(n)$ is independent of \mathbf{Q} , then $\mathbf{Q}\mathbf{U} \sim \text{HAAR}(\mathbb{O}(n))$ and $\mathbf{Q}\mathbf{U}$ is independent of \mathbf{U} .*

This rotational invariance can even hold when this orthogonal matrix \mathbf{U} depends on some of the columns of \mathbf{Q} , so long as \mathbf{U} acts only on a subset of coordinates.

Lemma B.2 (Reading a few columns is uninformative) *Fix $0 \leq t \leq n$, let $\mathbf{Q} \sim \text{HAAR}(\mathbb{O}(n))$ be a Haar-random orthogonal matrix, and suppose $\mathbf{T} \in \mathbb{O}(n-t)$ depends deterministically on the first t columns of \mathbf{Q} . Then $\mathbf{Q} \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \sim \text{HAAR}(\mathbb{O}(n))$.*

Proof By extended rotational invariance, the Haar-random matrix \mathbf{Q} can be realized as a product

$$\mathbf{Q} = \mathbf{Q}_1 \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 \end{bmatrix} \quad \text{for } \mathbf{Q}_1 \sim \text{HAAR}(\mathbb{O}(n)) \text{ and } \mathbf{Q}_2 \sim \text{HAAR}(\mathbb{O}(n-t)) \text{ independent.}$$

The matrix \mathbf{Q}_2 is independent of the first t columns of \mathbf{Q} and thus is independent of \mathbf{T} . So, by extended rotational invariance, $\mathbf{Q}_2\mathbf{T}$ is independent of \mathbf{Q}_1 . We conclude again by extended rotational invariance that

$$\mathbf{Q} \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} = \mathbf{Q}_1 \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2\mathbf{T} \end{bmatrix} \sim \text{HAAR}(\mathbb{O}(n)). \quad \blacksquare$$

With this result in place, we can now prove the hidden Haar theorem.

Proof of Theorem 3.1 We begin by building the matrices \mathbf{V} and \mathbf{U} . Assume without loss of generality that the queries $\{z_i\}$ are orthonormal.⁷ For each $i = 1, \dots, t$, define the query matrix

$$\mathbf{Z}_i := [z_1 \ \cdots \ z_i] \in \mathbb{R}^{n \times i},$$

and introduce the response matrix

$$\mathbf{V}_i := [v_1 \ \cdots \ v_i] = [Qz_1 \ \cdots \ Qz_i] \in \mathbb{R}^{n \times i}.$$

Use any deterministic procedure to complete \mathbf{Z}_i into a square orthogonal matrix

$$\mathbf{U}_i = [\mathbf{Z}_i \ \mathbf{Z}_i^\perp] \in \mathbb{O}(n).$$

Since the algorithm is deterministic, \mathbf{Z}_{i+1} (and thus \mathbf{U}_{i+1}) are both fully determined by the first i responses \mathbf{V}_i . Moreover, since the first i columns of \mathbf{U}_{i+1} and \mathbf{U}_i are the equal, there is an orthogonal matrix $\mathbf{T}_i \in \mathbb{O}(n-i)$ such that

$$\mathbf{U}_{i+1} = \mathbf{U}_i \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i \end{bmatrix}. \quad (\text{B.1})$$

Observe that the matrix \mathbf{T}_i is also fully determined by the responses \mathbf{V}_i . We define $\mathbf{U} := \mathbf{U}_t$ and $\mathbf{V} := \mathbf{V}_t$ and observe that

$$\mathbf{QU} = [\mathbf{V} \ \mathbf{W}] \quad \text{for some } \mathbf{W} \in \mathbb{R}^{n \times (n-t)}.$$

As desired, \mathbf{U} and \mathbf{V} depend only on the transcript \mathcal{T} . Next, we must establish that $\mathbf{QU} = \mathbf{QU}_t \sim \text{HAAR}(\mathbb{O}(n))$. To do so, we show by induction that $\mathbf{QU}_i \sim \text{HAAR}(\mathbb{O}(n))$ for each $i = 0, \dots, t$. The base case $i = 0$ is immediate. Next, we assume that $\mathbf{QU}_i \sim \text{HAAR}(\mathbb{O}(n))$ and prove $\mathbf{QU}_{i+1} \sim \text{HAAR}(\mathbb{O}(n))$. By (B.1), it holds that

$$\mathbf{QU}_{i+1} = (\mathbf{QU}_i) \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i \end{bmatrix}.$$

The matrix \mathbf{T}_i depends only on \mathbf{V}_i , which consists of the first i columns of \mathbf{QU}_i . By Lemma B.2, we conclude that $\mathbf{QU}_{i+1} \sim \text{HAAR}(\mathbb{O}(n))$, as desired.

Because the algorithm is deterministic, there is a one-to-one correspondence between transcripts \mathcal{T} and response matrices \mathbf{V} . Therefore, conditioning on \mathcal{T} is the same as conditioning on \mathbf{V} , and we conclude that \mathbf{W} has the specified conditional distribution. \blacksquare

7. Indeed, let z_1, \dots, z_t denote a general linearly independent query vectors, and reduce it to an orthonormal set $\tilde{z}_1, \dots, \tilde{z}_t$ by Gram-Schmidt orthonormalization. Every vector z_i is a linear combination of the orthonormal query vectors $\{\tilde{z}_j\}$, so the transcript $\mathcal{T} = (z_1, Qz_1, \dots, Qz_t)$ is a deterministic function of $\tilde{\mathcal{T}} = (\tilde{z}_1, Q\tilde{z}_1, \dots, Q\tilde{z}_t)$.

Appendix C. Proof of the fine-grained lower bound

In this section, we establish [Theorem 1.4](#), which gives a more fine-grained lower bound on the two-sided matrix–vector complexity of linear systems, as parametrized by a number k of outlying singular values and a reduced condition number $\kappa_k = \sigma_{k+1}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$. We use [Theorem 1.1](#) together with a reduction from [Dereziński et al. \(2025, Lem. 34\)](#).

We begin by importing the following lower bound from [Chewi et al. \(2024\)](#):

Imported Theorem C.1 (Trace-inverse: $\Omega(n)$ bound; Chewi et al. (2024, Thm. 4.2)) *There is a universal constant $\delta > 0$ such that any algorithm that produces an estimate $\tilde{\text{tr}}$ satisfying*

$$|\tilde{\text{tr}} - \text{tr}(\mathbf{A}^{-1})| \leq \frac{1}{2} \text{tr}(\mathbf{A}^{-1}) \quad \text{with prob. } \geq 1 - \delta$$

for all SPD matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ requires at least $\Omega(n)$ matvecs.

From this lower bound for the trace-inverse, we derive a lower bound for the linear system problem.

Lemma C.2 (Linear systems: $\Omega(n)$ lower bound) *Any algorithm that takes as input \mathbf{b} , makes two-sided matrix–vector queries with \mathbf{A} , and outputs a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{6n} \|\mathbf{b}\|_2 \quad \text{with prob. } \geq \frac{5}{6} \quad (\text{C.1})$$

for all SPD matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ must use at least $\Omega(n)$ matvecs.

Proof The proof uses a similar linear-systems-to-trace-inverse reduction to [Theorem 2.8](#), but there are additional complications associated with reducing the error to level $\frac{1}{2}$ and with boosting the success probability to level $1 - \delta$, as specified in the trace-inverse lower bound in [Imported Theorem C.1](#).

Let $\delta > 0$ be the universal constant in [Imported Theorem C.1](#), and let $\mathbf{b}_1, \dots, \mathbf{b}_k$ be iid vectors, each with iid Rademacher entries. Then by standard analysis of the Girard–Hutchinson trace estimator ([Meyer et al., 2021, Lem. 2.1](#)), setting $k = \mathcal{O}(\log(1/\delta)) = \mathcal{O}(1)$ suffices to achieve

$$\left| \frac{1}{k} \sum_{i=1}^k \mathbf{b}_i^\top \mathbf{A}^{-1} \mathbf{b}_i - \text{tr}(\mathbf{A}^{-1}) \right| \leq \frac{1}{4} \|\mathbf{A}^{-1}\|_{\text{F}} \quad \text{with prob. } \geq 1 - \frac{\delta}{2}. \quad (\text{C.2})$$

Suppose there is an algorithm that attains the residual guarantee [\(C.1\)](#) while using at most at most t matvecs. By taking the geometric median of $\mathcal{O}(\log(k/\delta)) = \mathcal{O}(1)$ independent executions of the algorithm, the success probability of the algorithm can be boosted to $1 - \delta/2k$ at the cost of inflating the residual by a factor of at most 1.5 ([Minsker, 2015, Thm. 3.1](#)). Applying this success-probability boosted procedure to each \mathbf{b}_i (and noting that $\|\mathbf{b}_i\|_2 = \sqrt{n}$), we obtain vectors $\tilde{\mathbf{x}}_i$ such that, for each i ,

$$\|\mathbf{A}\tilde{\mathbf{x}}_i - \mathbf{b}_i\|_2 \leq \frac{1}{4\sqrt{n}} \quad \text{with prob. } \geq 1 - \frac{\delta}{2k}.$$

The cost is $\mathcal{O}(tk \log(k/\delta)) = \mathcal{O}(t)$ matvecs. By union-bounding over these k outcomes, we see that, with probability at least $1 - \delta/2$,

$$\max_{i \in [k]} \|\tilde{\mathbf{x}}_i - \mathbf{A}^{-1} \mathbf{b}_i\|_2 \leq \|\mathbf{A}^{-1}\|_2 \max_{i \in [k]} \|\mathbf{A}\tilde{\mathbf{x}}_i - \mathbf{b}_i\|_2 \leq \frac{1}{4\sqrt{n}} \|\mathbf{A}^{-1}\|_{\text{F}}.$$

In turn, we find that $\tilde{\text{tr}} := \frac{1}{k} \sum_{i=1}^k \mathbf{b}_i^\top \tilde{\mathbf{x}}_i$ satisfies

$$\begin{aligned} \left| \tilde{\text{tr}} - \frac{1}{k} \sum_{i=1}^k \mathbf{b}_i^\top \mathbf{A}^{-1} \mathbf{b}_i \right| &\leq \frac{1}{k} \sum_{i=1}^k |\mathbf{b}_i^\top \tilde{\mathbf{x}}_i - \mathbf{b}_i^\top \mathbf{A}^{-1} \mathbf{b}_i| \leq \frac{1}{k} \sum_{i=1}^k \|\mathbf{b}_i\|_2 \|\tilde{\mathbf{x}}_i - \mathbf{A}^{-1} \mathbf{b}_i\|_2 \\ &\leq \frac{1}{k} \sum_{i=1}^k \left(\sqrt{n} \cdot \frac{1}{4\sqrt{n}} \|\mathbf{A}^{-1}\|_{\text{F}} \right) = \frac{1}{4} \|\mathbf{A}^{-1}\|_{\text{F}}. \end{aligned}$$

Finally, combining with guarantee (C.2) and union bounding over each bound succeeding, we see that with probability at least $1 - \delta$, it holds that

$$|\tilde{\text{tr}} - \text{tr}(\mathbf{A}^{-1})| \leq |H_k - \text{tr}(\mathbf{A}^{-1})| + |\tilde{\text{tr}} - H_k| \leq \frac{1}{2} \|\mathbf{A}^{-1}\|_{\text{F}} \leq \frac{1}{2} \text{tr}(\mathbf{A}^{-1}).$$

The last inequality is the comparison $\|\mathbf{M}\|_{\text{F}} \leq \text{tr}(\mathbf{M})$, which is valid for any positive semidefinite matrix \mathbf{M} . We have computed an estimator satisfying the guarantee in [Imported Theorem C.1](#) using $\mathcal{O}(t)$ matvecs. We conclude that $t \geq \Omega(n)$, as desired. \blacksquare

The proof of [Theorem 1.4](#) is a corollary of the $\Omega(\kappa \log(1/\varepsilon))$ matvecs lower bound in [Theorem 1.1](#), the $\Omega(n)$ lower bound in [Lemma C.2](#), and a result from prior work explained below.

Definition C.3 (Complexity) Fix any set of square matrices $\mathcal{A} \subseteq \mathbb{R}^{n \times n}$. Then the matrix-vector complexity for solving linear systems in \mathcal{A} , denoted $T_{\mathcal{A}}(\varepsilon)$, is the smallest number t such that there exists a two-sided matrix-vector algorithm which computes at most t matrix-vector products and returns a vector $\tilde{\mathbf{x}}$ such that

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \varepsilon \|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all $\mathbf{A} \in \mathcal{A}$ and all $\mathbf{b} \in \mathbb{R}^n$.

For instance, if $\mathcal{M}(n)$ denotes the set of all $n \times n$ matrices, then [Lemma C.2](#) says that $T_{\mathcal{M}(n)}(1/6n) = \Omega(n)$. Similarly, if $\mathcal{B}(n, \kappa)$ denotes the set of all $n \times n$ with condition number at most κ , then $T_{\mathcal{B}(n, \kappa)}(1/n) = \Omega(\kappa \log(1/n))$ by [Theorem 2.8](#). For the purposes of fine-grained complexity, we define $\mathcal{F}(n, k, \kappa_k)$ to be the set of all matrices \mathbf{A} with $\sigma_{k+1}(\mathbf{A})/\sigma_{\min}(\mathbf{A}) \leq \kappa_k$. [Dereziński et al. \(2025\)](#) shows how we can relate the matvec complexity of the fine-grained problem to that of the coarser problems:

Imported Theorem C.4 (Complexity reduction; [Dereziński et al. \(2025, Lem. 27 and Rem. 8\)](#))
With $\mathcal{M}(n)$, $\mathcal{B}(n, \kappa)$, and $\mathcal{F}(n, k, \kappa_k)$ as defined above,

$$T_{\mathcal{F}(n, k, \kappa_k)}(\varepsilon) \gtrsim T_{\mathcal{M}(k)}(\varepsilon) + T_{\mathcal{B}(n-k, \kappa_k)}(\varepsilon).$$

That is, the worst-case fine-grained matvec complexity of linear system solving is the sum of the matvec complexity of solving an arbitrary $k \times k$ linear system plus the matvec complexity of solving a $(n-k) \times (n-k)$ linear system with condition number κ_k . We can now prove [Theorem 1.4](#).

Proof of Theorem 1.4 Fix an accuracy level $\varepsilon > 0$, and choose k and n sufficiently large so that we can instantiate [Theorem 2.8](#) and so that $1/6k < \varepsilon$ and $1/(n-k) < \varepsilon$. Clearly, the complexity is decreasing as a function of the error tolerance ε . Therefore, [Lemma C.2](#) implies that $T_{\mathcal{M}(k)}(\varepsilon) = \Omega(k)$ and

[Theorem 2.8](#) implies that $T_{\mathcal{B}(n-k, \kappa_k)}(\varepsilon) = \Omega(\kappa_k \log(1/\varepsilon))$. Invoking [Imported Theorem C.4](#) yields the desired conclusion:

$$T_{\mathcal{F}(n, k, \kappa_k)}(\varepsilon) = \Omega(k + \kappa_k \log(1/\varepsilon)). \quad \blacksquare$$

Appendix D. Deferred proofs

In this section, we provide additional proofs of statements from the main text.

D.1. Proof of [Proposition 2.2](#)

We let $\mathbf{b} \in \mathbb{R}^n$ be a uniformly random vector such that $\|\mathbf{b}\|_2 = \sqrt{n}$. Then, by [Girard \(1987, Prop. 3.1\)](#)⁸, we know that

$$\mathbb{E}[\mathbf{b}^\top f(\mathbf{A})\mathbf{b}] = \text{tr}(f(\mathbf{A})) \quad \text{and} \quad \text{Var}[\mathbf{b}^\top f(\mathbf{A})\mathbf{b}] \leq 2\|f(\mathbf{A})\|_{\mathbb{F}}^2.$$

So, with probability at least $5/6$, we have $|\mathbf{b}^\top f(\mathbf{A})\mathbf{b} - \text{tr}(\mathbf{A})| \leq \sqrt{12}\|f(\mathbf{A})\|_{\mathbb{F}}$. Then the output $\tilde{\mathbf{x}}$ of our oracle has

$$|\mathbf{b}^\top f(\mathbf{A})\mathbf{b} - \mathbf{b}^\top \tilde{\mathbf{x}}| \leq \|\mathbf{b}\|_2 \|f(\mathbf{A})\mathbf{b} - \tilde{\mathbf{x}}\|_2 \leq \frac{1}{n} \|\mathbf{b}\|_2^2 \|f(\mathbf{A})\|_{\mathbb{F}} = \|f(\mathbf{A})\|_{\mathbb{F}}$$

with probability at least $5/6$. We conclude that the trace estimator $\tilde{\text{tr}} = \mathbf{b}^\top \tilde{\mathbf{x}}$ has

$$|\tilde{\text{tr}} - \text{tr}(f(\mathbf{A}))| \leq |\text{tr}(f(\mathbf{A})) - \mathbf{b}^\top f(\mathbf{A})\mathbf{b}| + |\mathbf{b}^\top f(\mathbf{A})\mathbf{b} - \mathbf{b}^\top \tilde{\mathbf{x}}| \leq (1 + \sqrt{12})\|f(\mathbf{A})\|_{\mathbb{F}} \leq 5\|f(\mathbf{A})\|_{\mathbb{F}}$$

with probability at least $2/3$.

D.2. Proof of [Theorem 2.8](#)

Before proving [Theorem 2.8](#), we instate a common technique from the literature.

Proposition D.1 (Iterative refinement) *Suppose we are given access to an oracle which, given inputs \mathbf{A} and \mathbf{b} , returns a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{2}\|\mathbf{b}\|_2 \quad \text{with prob. } 1 - \delta.$$

Then, there exists an algorithm which uses the oracle r times and performs r one-sided matrix-vector products to produce a solution $\tilde{\mathbf{x}}$ satisfying

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq 2^{-r}\|\mathbf{b}\|_2 \quad \text{with prob. } \geq 1 - r\delta.$$

This type of result and its proof are standard ([Greenbaum, 1997](#), Sec. 2.1). The idea is to use the oracle to produce an approximate solution $\tilde{\mathbf{x}}_1 \approx \mathbf{A}^{-1}\mathbf{b}$. Then, we form the residual $\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}_1$, use the oracle to produce an approximate solution $\tilde{\delta}_1 \approx \mathbf{A}^{-1}\mathbf{r}_1$, and form the corrected solution $\tilde{\mathbf{x}}_2 = \tilde{\mathbf{x}}_1 + \tilde{\delta}_1$. Continuing this procedure for r rounds produces a solution $\tilde{\mathbf{x}}_r$ satisfying the guarantee; we omit a detailed proof. Using this result, we prove [Theorem 2.8](#).

⁸ This article is in French. See, e.g., [Epperly \(2025, Fact. 13.2\)](#) for a statement in English.

Proof of Theorem 2.8 Fix any matrix \mathbf{A} of condition number at most κ and any vector \mathbf{b} . Let $\tilde{\mathbf{x}}$ be any vector returned by a two-sided matrix-vector algorithm such that

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{2}{3}.$$

Then, we can ensure that

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 = \|\mathbf{A}^{-1}\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b}\|_2 \leq \|\mathbf{A}^{-1}\|_2\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{A}^{-1}\|_{\text{F}}\|\mathbf{b}\|_2.$$

also holds with probability at least $2/3$. Proposition 2.2 tells us that this matrix-vector algorithm can estimate $\text{tr}(\mathbf{A}^{-1})$ with probability at least $2/3$, and Theorem 2.6 tells us that our matrix-vector algorithm must have used at least $\frac{1}{4}\kappa \log(n)$ matvecs, completing the first part of the proof.

For the second part of the proof, suppose that a two-sided matrix-vector algorithm uses at most t matvecs to satisfy (2.3). We then apply Proposition D.1 with $r = \Theta(\log(n))$, resulting in a vector $\tilde{\mathbf{x}}$ that satisfies (2.2) with a total of $r(t+1)$ matvecs. Since it takes $\Omega(\kappa \log n)$ matvecs to satisfy (2.2), we know that our low-accuracy solver must have used $t = \Omega(\kappa)$ matvecs. \blacksquare

Appendix E. Additional lower bounds

In this section, we briefly note how our methods give variably tight lower bounds for a variety of other matrix-vector complexities in and around linear system solving.

E.1. ℓ_2 error guarantees

Using a variant of the proof of Theorem 2.8, we obtain lower bounds for the *forward error* measured in the ℓ_2 norm.

Corollary E.1 (Linear systems lower bound in ℓ_2 norm) *Suppose that $n \gtrsim \kappa^{31}$. Any algorithm using fewer than $\frac{1}{5}\kappa \log n$ two-sided matvecs with a matrix \mathbf{A} cannot return a vector $\tilde{\mathbf{x}}$ such that*

$$\|\tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{A}^{-1}\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\text{cond}(\mathbf{A}) \leq \kappa$ and all vectors $\mathbf{b} \in \mathbb{R}^n$. Moreover, attaining the guarantee

$$\|\tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b}\|_2 \leq \frac{1}{2}\|\mathbf{A}^{-1}\mathbf{b}\|_2 \quad \text{with prob.} \geq 1 - \frac{1}{10 \log n}$$

for the same class of problems requires $\Omega(\kappa)$ matvecs.

The proof of Corollary E.1 is omitted; the proof is nearly the same as the proof of Theorem 2.8 found in Appendix D.2. The key step is realizing the guarantee Corollary E.1 implies

$$\|\tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{A}^{-1}\mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{A}^{-1}\|_2\|\mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{A}^{-1}\|_{\text{F}}\|\mathbf{b}\|_2,$$

which was the same bound established in the proof of 2.8. Corollary E.1 shows that resolving a linear system to relative forward error ε also requires $\Omega(\kappa \log(1/\varepsilon))$ matvecs. The best-known upper bounds for achieving an ε -small relative forward error require at most $\mathcal{O}(\kappa \log(\kappa/\varepsilon))$ matvecs. The extra $\log(\kappa)$ factor comes from a comparison between the norms $\|\mathbf{A}^{-1}\mathbf{b}\|_2$ and $\|\mathbf{b}\|_2$. To sharpen the lower bound to match the upper bound, it would suffice to produce a lower bound of $\kappa \log(\kappa)$ matvecs being needed in the low-accuracy regime $\varepsilon \asymp 1$.

E.2. Lower bound for SPD systems

Lastly, we show an $\Omega(\sqrt{\kappa} \log(1/\varepsilon))$ lower bound for SPD matrices. An $\Omega(\sqrt{\kappa} \log(1/\varepsilon))$ lower bound is known for deterministic algorithms (Nemirovsky and Yudin, 1983, Sec. 7.2) and an $\Omega(\sqrt{\kappa}/\text{polylog}(\kappa))$ lower bound is known for randomized algorithms Braverman et al. (2020).

Theorem E.2 (Linear systems: Lower bound, PSD Variant) *Fix any $\delta, \eta \in (0, 1)$ and suppose that $n \gtrsim \kappa^{(6+\delta)/\eta}$. Any randomized algorithm using fewer than $\frac{1-\eta}{4}\sqrt{\kappa} \log n$ two-sided matvecs with a matrix \mathbf{A} cannot return a vector $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{n}\|\mathbf{b}\|_2 \quad \text{with prob.} \geq \frac{5}{6}$$

for all SPD matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\text{cond}(\mathbf{A}) \leq \kappa$ and all vectors $\mathbf{b} \in \mathbb{R}^n$. Moreover, attaining the guarantee

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \frac{1}{2}\|\mathbf{b}\|_2 \quad \text{with prob.} \geq 1 - \frac{1}{10 \log n}$$

for the same class of problems requires $\Omega(\sqrt{\kappa})$ matvecs.

Proof The proof matches that of Theorem 2.8, except that it uses Imported Lemma 2.9 in place of Lemma 2.5. ■

Lower bounds for the forward error in the ℓ_2 - and \mathbf{A} -norms are also possible, showing that $\Omega(\sqrt{\kappa} \log(1/\varepsilon))$ matvecs are needed in both cases.

References

- Noah Amsel, Tyler Chen, Feyza Duman Keles, Diana Halikias, Cameron Musco, and Christopher Musco. Fixed-sparsity matrix approximation from matrix-vector products. *SIAM Journal on Matrix Analysis and Applications*, 47(2):483–511, June 2026. doi: [10.1137/25M1742710](https://doi.org/10.1137/25M1742710). (Cited on pages 10 and 11.)
- Owe Axelsson and Gunhild Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48:499–523, 1986. doi: [10.1007/BF01389448](https://doi.org/10.1007/BF01389448). (Cited on page 5.)
- Ainesh Bakshi and Shyam Narayanan. Krylov methods are (nearly) optimal for low-rank approximation. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science*, pages 2093–2101. IEEE, 2023. doi: [10.1109/FOCS57990.2023.00128](https://doi.org/10.1109/FOCS57990.2023.00128). (Cited on pages 6 and 15.)
- Ainesh Bakshi, Kenneth L. Clarkson, and David P. Woodruff. Low-rank approximation with $\varepsilon^{1/3}$ matrix-vector products. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1130–1143. ACM, June 2022. doi: [10.1145/3519935.3519988](https://doi.org/10.1145/3519935.3519988). (Cited on page 15.)
- Nicolas Boullé, Diana Halikias, Samuel E. Otto, and Alex Townsend. Operator learning without the adjoint. *Journal of Machine Learning Research*, 25(364):1–54, 2024. URL <http://jmlr.org/papers/v25/24-0162.html>. (Cited on page 5.)
- Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004. doi: [10.1017/CBO9780511804441](https://doi.org/10.1017/CBO9780511804441). (Cited on pages 1 and 5.)
- Mark Braverman, Elad Hazan, Max Simchowitz, and Blake Woodworth. The gradient complexity of linear regression. In *Conference on Learning Theory*, pages 627–647. PMLR, 2020. URL <https://proceedings.mlr.press/v125/braverman20a.html>. (Cited on pages 2, 4, 10, 11, 15, and 30.)
- Clément L. Canonne. *Topics and Techniques in Distribution Testing*. Now Publishers, 2022. doi: [10.1561/9781638281016](https://doi.org/10.1561/9781638281016). (Cited on page 14.)
- Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020. doi: [10.1007/s10107-019-01406-y](https://doi.org/10.1007/s10107-019-01406-y). (Cited on page 11.)
- Sinho Chewi, Jaume de Dios Pont, Jerry Li, Chen Lu, and Shyam Narayanan. Query lower bounds for log-concave sampling. *Journal of the ACM*, 71(4):1–42, 2024. doi: [10.1145/3673651](https://doi.org/10.1145/3673651). (Cited on pages 4, 6, 7, 8, 11, 14, 15, 16, 17, 19, 21, 22, and 26.)
- Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, STOC '09, pages 205–214. Association for Computing Machinery, May 2009. doi: [10.1145/1536414.1536445](https://doi.org/10.1145/1536414.1536445). (Cited on page 4.)
- Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63(6):1–45, February 2017. doi: [10.1145/3019134](https://doi.org/10.1145/3019134). (Cited on page 4.)
- 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. doi: [10.1145/3637528.3671461](https://doi.org/10.1145/3637528.3671461). (Cited on page 2.)

- Michał Dereziński and Aaron Sidford. Approaching optimality for solving dense linear systems with low-rank structure. In *Proceedings of the 37th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2026)*. SIAM / ACM, 2026. doi: [10.1137/1.9781611978971.37](https://doi.org/10.1137/1.9781611978971.37). (Cited on pages 4 and 5.)
- Michał Dereziński and Jiaming Yang. Solving dense linear systems faster than via preconditioning. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1118–1129, 2024. doi: [10.1145/3618260.3649694](https://doi.org/10.1145/3618260.3649694). (Cited on page 5.)
- Michał Dereziński, Daniel LeJeune, Deanna Needell, and Elizaveta Rebrova. Fine-grained analysis and faster algorithms for iteratively solving linear systems. *Journal of Machine Learning Research*, 26(144):1–49, 2025. URL <https://www.jmlr.org/papers/v26/24-1906.html>. (Cited on pages 5, 26, and 27.)
- Michał Dereziński, Christopher Musco, and Jiaming Yang. Faster linear systems and matrix norm approximation via multi-level sketched preconditioning. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1972–2004. SIAM, January 2025. doi: [10.1137/1.9781611978322.62](https://doi.org/10.1137/1.9781611978322.62). (Cited on page 5.)
- Ronald A DeVore and George G Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993. (Cited on page 17.)
- Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional Gaussians with the same mean. *arXiv preprint arXiv:1810.08693v7*, February 2022. (Cited on page 22.)
- Tobin A. Driscoll, Kim-Chuan Toh, and Lloyd N. Trefethen. From potential theory to matrix iterations in six steps. *SIAM Review*, 40(3):547–578, January 1998. doi: [10.1137/S0036144596305582](https://doi.org/10.1137/S0036144596305582). (Cited on page 5.)
- Ethan N. Epperly. *Make the most of what you have: Resource-efficient randomized algorithms for matrix computations*. PhD thesis, California Institute of Technology, 2025. (Cited on page 28.)
- Didier Girard. Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille, 1987. URL <https://membres-ljk.imag.fr/Didier.Girard/TR-665-M-IMAG.pdf>. (Cited on page 28.)
- Anne Greenbaum. *Iterative methods for solving linear systems*. Number 17 in Frontiers in applied mathematics. Society for Industrial and Applied Mathematics, 1997. doi: [10.1137/1.9781611970937](https://doi.org/10.1137/1.9781611970937). (Cited on pages 1, 2, 3, 4, 5, and 28.)
- Diana Halikias. *Structured Matrix Recovery and Approximation from Matrix-Vector Products*. PhD, Cornell University, 2025. (Cited on pages 5 and 6.)
- Aram W. Harrow, Avinandan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, October 2009. doi: [10.1103/PhysRevLett.103.150502](https://doi.org/10.1103/PhysRevLett.103.150502). (Cited on page 6.)
- Shuli Jiang, Hai Pham, David Woodruff, and Richard Zhang. Optimal sketching for trace estimation. *Advances in Neural Information Processing Systems*, 34:23741–23753, 2021. URL <https://dl.acm.org/doi/10.5555/3540261.3542079>. (Cited on page 11.)
- Johannes K Kraus, Panayot S Vassilevski, and Ludmil T Zikatanov. Polynomial of best uniform approximation to x^{-1} and smoothing in two-level methods. *arXiv preprint arXiv:1002.1859v3*, 2010. (Cited on page 10.)
- David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. (Cited on page 15.)

- Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020. doi: [10.1017/S0962492920000021](https://doi.org/10.1017/S0962492920000021). (Cited on page 2.)
- Raphael A. Meyer. A non-asymptotic bound on the TV distance between a Wishart matrix and an appropriately scaled GOE matrix. *arXiv preprint arXiv:2606.16018*, 2026. (Cited on page 21.)
- Raphael A. Meyer and Haim Avron. Hutchinson’s estimator is bad at Kronecker-Trace-Estimation. *SIAM Journal on Matrix Analysis and Applications*, 47(1):1–32, 2026. doi: [10.1137/23M1595180](https://doi.org/10.1137/23M1595180). (Cited on page 11.)
- Raphael A. Meyer, Cameron Musco, Christopher Musco, and David P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *Proceedings of the 2021 Symposium on Simplicity in Algorithms*, pages 142–155. SIAM, 2021. doi: [10.1137/1.9781611976496.16](https://doi.org/10.1137/1.9781611976496.16). (Cited on pages 4, 13, 15, and 26.)
- Raphael Arkady Meyer. *Towards Optimal Matrix-Vector Complexity in Numerical Linear Algebra*. PhD thesis, New York University Tandon School of Engineering, 2024. URL https://ram900.com/assets/thesis-main_final_v3.pdf. (Cited on pages 11 and 13.)
- Stanislav Minsker. Geometric median and robust estimation in Banach spaces. *Bernoulli*, pages 2308–2335, 2015. doi: [10.3150/14-BEJ645](https://doi.org/10.3150/14-BEJ645). (Cited on page 26.)
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT press, 2018. (Cited on page 1.)
- Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1st edition edition, August 1995. doi: [10.1017/CBO9780511814075](https://doi.org/10.1017/CBO9780511814075). (Cited on pages 12 and 16.)
- Riley Murray, James Demmel, Michael W Mahoney, N Benjamin Erichson, Maksim Melnichenko, Osman Asif Malik, Laura Grigori, Piotr Luszczek, Michał Dereziński, Miles E Lopes, Tianyu Liang, Hengrui Luo, and Jack Dongarra. Randomized numerical linear algebra: A perspective on the field with an eye to software. *arXiv preprint arXiv:2302.11474v2*, 2023. (Cited on page 2.)
- C. Musco and D. P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, pages 672–683, October 2017. doi: [10.1109/FOCS.2017.68](https://doi.org/10.1109/FOCS.2017.68). (Cited on page 4.)
- Noël M. Nachtigal, Satish C. Reddy, and Lloyd N. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM Journal on Matrix Analysis and Applications*, 13(3):778–795, July 1992. doi: [10.1137/0613049](https://doi.org/10.1137/0613049). (Cited on pages 2 and 3.)
- Arkadij Semenovič Nemirovsky and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983. URL https://www2.isye.gatech.edu/~nemirovs/Nemirovskii_Yudin_1983.pdf. (Cited on pages 2, 4, and 30.)
- Christopher C. Paige and Michael A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982. doi: [10.1145/355984.355989](https://doi.org/10.1145/355984.355989). (Cited on page 3.)
- Richard Peng and Santosh S. Vempala. Solving sparse linear systems faster than matrix multiplication. *Commun. ACM*, 67(7):79–86, July 2024. doi: [10.1145/3615679](https://doi.org/10.1145/3615679). (Cited on page 4.)
- Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover Books on Advanced Mathematics. Dover, 1981. (Cited on page 17.)
- 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, September 2008. doi: [10.1073/pnas.0804869105](https://doi.org/10.1073/pnas.0804869105). (Cited on page 4.)

- Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986. doi: [10.1137/0907058](https://doi.org/10.1137/0907058). (Cited on page 2.)
- Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. Tight query complexity lower bounds for PCA via finite sample deformed Wigner law. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1249–1259, 2018. doi: [10.1145/3188745.3188796](https://doi.org/10.1145/3188745.3188796). (Cited on pages 10, 11, and 15.)
- Daniel Spielman. Solve for x: Technical perspective. *Commun. ACM*, 67(7):78, July 2024. doi: [10.1145/3643835](https://doi.org/10.1145/3643835). (Cited on page 1.)
- Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009. doi: [10.1145/1562764.1562785](https://doi.org/10.1145/1562764.1562785). (Cited on page 5.)
- J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity*. Academic Press Professional, Inc., 1988. (Cited on page 2.)
- David Woodruff, Fred Zhang, and Richard Zhang. Optimal query complexities for dynamic trace estimation. *Advances in Neural Information Processing Systems*, 35:35049–35060, 2022. URL <https://dl.acm.org/doi/10.5555/3600270.3602810>. (Cited on page 11.)
- Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592–617. PMLR, 2013. URL <https://proceedings.mlr.press/v30/Zhang13.html>. (Cited on page 5.)