Fast, Parallel, Query-Efficient Binary Classification

Ishani Karmarkar ISHANIK@STANFORD.EDU
Stanford University

Liam O'Carroll OCARROLL@STANFORD.EDU
Stanford University

Aaron Sidford SIDFORD@STANFORD.EDU
Stanford University

Editors: Steve Hanneke and Tor Lattimore

Abstract

We study the fundamental classification problem of computing a separating hyperplane for a binary-labeled dataset of size n with normalized d-dimensional features. Letting $\Phi \in \mathbb{R}^{n\times d}$ denote the feature matrix and γ the margin of the maximum-margin separating hyperplane, we present a randomized algorithm that solves this problem in $\tilde{O}(\gamma^{-2/3}\ \mathrm{nnz}(\Phi) + \gamma^{-2(\omega+1)/3})$-sequential running time (work), $\tilde{O}(\gamma^{-2/3})$-parallel (computational) depth, and accesses Φ only through $\tilde{O}(\gamma^{-2/3})$-matrix-vector queries (matvecs). We also present a second, faster randomized algorithm with a $\tilde{O}(\gamma^{-2/3}\ \mathrm{nnz}(\Phi) + \gamma^{-2})$-sequential running time that uses $\tilde{O}(\gamma^{-2/3})$-matvecs to Φ, but achieves only $\tilde{O}(\gamma^{-4/3})$-parallel depth. Both algorithms match the near-optimal deterministic matvec complexity recently established by (Kornowski and Shamir, 2025a; Karmarkar et al., 2026) and achieve improved sequential runtime and parallel depth, albeit at the expense of using randomness.

Keywords: Separating hyperplanes, hard-margin support vector machines, matrix games

1. Introduction

In this paper, we study the foundational binary data classification problem of finding a linear separator for two sets of points. Concretely, we consider the *separating hyperplane*, or *hard-margin support vector machine (SVM) problem*, in which we are given a dataset $\mathcal{D} = (\phi_i, l_i)_{i\in[n]}$ where $\phi_i \in \mathbb{B}^d := \{x \in \mathbb{R}^d : \|x\|_2 \le 1\}$ is a normalized feature vector and $l_i \in \{+1, -1\}$ is a binary label for each $i \in [n]$. We let $\gamma_{\mathcal{D}}$ denote the maximum-margin of a separating hyperplane for $\mathcal{D}$, i.e.,

$$\gamma_{\mathcal{D}} := \max_{w\in\mathbb{B}^d} \min_{i\in[n]} l_i\langle w, \phi_i\rangle, \text{ and hence, } \exists w^\star \in \mathbb{B}^d \text{ such that } l_i\langle w^\star, \phi_i\rangle \ge \gamma_{\mathcal{D}} \text{ for all } i \in [n]. \quad (1)$$

We consider the problem where, given $\rho \in (0, \gamma_{\mathcal{D}})$, we must find a hyperplane $\hat{w} \in \mathbb{B}^d$ which achieves a margin of $\gamma_{\mathcal{D}} - \rho$.[1] We formalize this problem in the following Definition 1.

Definition 1 (Maximum Margin Separating Hyperplane Problem) *In the ρ-(maximum margin) separating hyperplane problem, we are given a (binary-labeled)*dataset $\mathcal{D} = \{\phi_i \in \mathbb{B}^d, l_i \in \{-1, +1\}\}_{i\in[n]}$ *and $\rho > \gamma_{\mathcal{D}}$, and must output $\hat{w} \in \mathbb{B}^d$ such that $l_i\langle \hat{w}, \phi_i\rangle \ge \gamma_{\mathcal{D}} - \rho$ for all $i \in [n]$. We say that such a $\hat{w}$* induces a ρ-separating *hyperplane for $\mathcal{D}$.*

1. For simplicity, as in prior work (Kornowski and Shamir, 2025a; Karmarkar et al., 2026, 2025; Carmon et al., 2020b, 2024a, 2019), in this paper we focus on the setting of (1), which assumes linear separability through the origin with maximum margin $\gamma_{\mathcal{D}}$. However, using standard reductions, (1) and correspondingly Definition 1 can be relaxed to assume linear separability under arbitrary affine hyperplanes.

This is an incredibly well-studied problem in machine learning, dating back to (Rosenblatt, 1958; McCulloch and Pitts, 1943), with numerous applications (especially as it can be extended to non-linear settings Murty and Raghava (2016)). Our focus in this work is to introduce new, faster randomized algorithms for this foundational learning problem.

Our results. For notational convenience, for a dataset $\mathcal{D} = \{\phi_i \in \mathbb{B}^d, l_i \in \{+1, -1\}\}_{i\in[n]}$ we use $\Phi_{\mathcal{D}} \in \mathbb{R}^{n\times d}$ to denote the covariate matrix whose i-th row is given by $l_i\phi_i^\top$. In addition, for $A \in \mathbb{R}^{n\times d}$, $\mathrm{nnz}(A) := |\{(i,j) : A_{ij} \neq 0\}| + n + d$ denotes an augmented nonzero count.

Our main result, given in Theorems 2 and 3 below, is two parallel, query-efficient algorithms for the maximum-margin separating hyperplane problem (Definition 1) which offer different trade-offs between complexity metrics. The complexity metrics we study are *depth*, *work* (or sequential runtime), and *matvec* complexity.

- *Depth and work*: Following Jambulapati et al. (2024a), we say an algorithm has *(computational) depth* D if the number of sequential rounds of computation is D. In particular, we assume element-wise vector operations (e.g., adding/scaling vectors) in $\mathbb{R}^k$ incur $O(1)$-depth and $O(k)$-work and that dot products and matrix-vector multiplications require $O(\log k)$ depth. We let $\omega < 2.3714$ denote the fast matrix multiplication (FMM) constant (Alman et al., 2025); namely, two $k \times k$ matrices can be multiplied in $O(k^\omega)$-work, as well as $\tilde{O}(1)$-depth Pan and Reif (1985); Pan (1987).

- *Matvec complexity*: Formally, we say that an algorithm can be implemented using T matvecs to a matrix $A \in \mathbb{R}^{n\times d}$ if it can be implemented using T queries to a *matvec oracle for* A, which for any query $(x, y) \in \mathbb{R}^d \times \mathbb{R}^n$ outputs $(A^\top y, Ax)$.

In the following Theorems 2 and 3 (and throughout the paper) we use $\tilde{O}(\cdot)$ to suppress polylogarithmic factors in n, d, ρ^{-1}, and ϵ^{-1} (which appears later). Theorem 2 corresponds to our most parallelizable algorithm, notably achieving $\tilde{O}(\rho^{-2/3})$-depth.

Theorem 2 (Fast, parallel binary data classification) *There is a randomized algorithm which, with probability at least* $2/3$*, solves the* ρ*-separating hyperplane problem and runs in*

$$\tilde{O}(\rho^{-2/3}\,\mathrm{nnz}(\Phi_{\mathcal{D}}) + \rho^{-2(\omega+1)/3})\text{-}\textit{work and } \tilde{O}(\rho^{-2/3})\text{-}\textit{depth}\,.$$

Moreover, the algorithm can be implemented with $\tilde{O}(\rho^{-2/3})$*-matvecs to* $\Phi_{\mathcal{D}}$*.*

Interestingly, our next result shows that the poly(ρ^{-1}) dependence in Theorem 2 can be further improved (unless $\omega = 2$), albeit at the cost of increased depth.

Theorem 3 (Faster binary data classification) *There is a randomized algorithm which, with probability at least* $2/3$*, solves the* ρ*-separating hyperplane problem and runs in*

$$\tilde{O}(\rho^{-2/3}\,\mathrm{nnz}(\Phi_{\mathcal{D}}) + \rho^{-2})\text{-}\textit{work and } \tilde{O}(\rho^{-4/3})\text{-}\textit{depth}\,.$$

Moreover, the algorithm can be implemented with $\tilde{O}(\rho^{-2/3})$*-matvecs to* $\Phi_{\mathcal{D}}$*.*

Next, we compare Theorems 2 and 3 to the prior art with respect to each of the three complexity metrics, work, depth, and matvecs, that we study.

Total work. Table 1 summarizes our results in this metric compared to the prior art. Recalling $\mathrm{nnz}(\Phi_{\mathcal{D}}) = \Omega(n+d)$, we have the following comparisons: The work achieved in Theorem 3 improves the $(n+d)\rho^{-2}$ complexity of the second row in the moderate-to-high precision or sparse regime $\rho \ll (\frac{n+d}{\mathrm{nnz}(\Phi_{\mathcal{D}})})^{3/4}$. Theorem 3 improves upon the $\mathrm{nnz}(\Phi_{\mathcal{D}}) + \sqrt{\mathrm{nnz}(\Phi_{\mathcal{D}}) \cdot (n+d)}\rho^{-1}$ work of the third row (and therefore also the first row) in the sparse, moderate precision regime

$$\frac{1}{\sqrt{\mathrm{nnz}(\Phi_{\mathcal{D}})(n+d)}} \ll \rho \ll \left(\frac{n+d}{\mathrm{nnz}(\Phi_{\mathcal{D}})}\right)^{3/2}.$$

Theorem 3 improves upon the $nd + nd^{2/3}\rho^{-2/3} + d\rho^{-2}$ work of the fourth row in the sufficiently sparse regime $\mathrm{nnz}(\Phi_{\mathcal{D}}) \ll nd^{2/3}$ or the moderate-to-high precision regime $\rho \ll (\frac{d}{\mathrm{nnz}(\Phi_{\mathcal{D}})})^{3/4}$. Importantly, we improve upon *all* prior art in the sparse, low-to-moderate precision regime where $\mathrm{nnz}(\Phi_{\mathcal{D}}) \approx n+d$, $\frac{1}{n+d} \ll \rho \ll 1$, and $d \gg 1$. This regime includes $\rho = 1/\sqrt{n}$, which is standard in empirical risk minimization problems where statistical noise limits meaningful precision to $1/\sqrt{n}$.

We note that additional total work complexities may be obtained using interior point methods (IPMs), e.g., (Cohen et al., 2021; Van Den Brand et al., 2021), which may improve on Theorems 2 and 3 in certain high-accuracy regimes (where ρ is very small). However, current IPMs have at least a *quadratic* dependence on $\min\{n, d\}$, unlike our methods and most of the remaining methods in Table 1 (in sufficiently sparse regimes). As we do not focus on this high-accuracy regime, for simplicity, we do not compare in detail against IPMs and defer the reader to, e.g., (Carmon et al., 2024a) for more details.

Method	Total work
(Exact) gradient methods (Nemirovski, 2004; Nesterov, 2007)	$\mathrm{nnz}(\Phi_{\mathcal{D}})\rho^{-1}$
Row-col randomized method (Grigoriadis and Khachiyan, 1995; Palaniappan and Bach, 2016; Clarkson et al., 2012)	$(n+d)\rho^{-2}$
Row-col randomized method with variance-reduction (Palaniappan and Bach, 2016; Carmon et al., 2019)	$\mathrm{nnz}(\Phi_{\mathcal{D}}) + \sqrt{\mathrm{nnz}(\Phi_{\mathcal{D}}) \cdot (n+d)}\rho^{-1}$
Primal ball-accelerated stochastic methods (Carmon et al., 2024a)	$nd + nd^{2/3}\rho^{-2/3} + d\rho^{-2}$
Theorem 2	$\mathrm{nnz}(\Phi_{\mathcal{D}})\rho^{-2/3} + \rho^{-2(\omega+1)/3}$
Theorem 3	$\mathrm{nnz}(\Phi_{\mathcal{D}})\rho^{-2/3} + \rho^{-2}$

Table 1: Comparison to prior work. The table shows the total work complexities of the prior art, up to polylogarithmic factors in n, d and ρ^{-1}. The complexities in the abstract follow from $\rho = \gamma_{\mathcal{D}}/2$.

Depth. Mirror descent Nemirovskij and Yudin (1983); Beck and Teboulle (2003) achieves $\tilde{O}(\rho^{-2})$-depth for this problem, and a variety of deterministic algorithms (accelerated gradient descent, mirror prox, dual extrapolation) achieve an improved $\tilde{O}(\rho^{-1})$-depth Nesterov (2005); Nemirovski (2004); Nesterov (2007). We are not aware of any prior deterministic or randomized algorithms which improve upon $\tilde{O}(\rho^{-1})$-depth in the nearly dimension-free parallel regime we study.

Note that although (Karmarkar et al., 2025, 2026) achieve a $\tilde{o}(\rho^{-1})$-matvec complexity for the problem, they do not immediately yield improved parallel depths for the problem. The algorithms of (Karmarkar et al., 2025, 2026) perform $\tilde{O}(\rho^{-2/3})$ *outer loop iterations*; however, each outer loop iteration in turn requires solving multiple constrained, convex optimization sub-problems to high accuracy. Consequently, the work and depth complexities of their inner loop itself scale polynomially in n, d, ρ^{-1}. Due to the complex nature of these sub-problems, it is perhaps not straightforward to bound this polynomial; (Karmarkar et al., 2025, 2026) do not bound the work or depth complexity of their method and leave this as a direction for future work.

We also note that interior point methods (IPMs) or other existing parallel algorithms Nemirovski (1994); Duchi et al. (2012); Bubeck et al. (2019); Carmon et al. (2023); Jambulapati et al. (2024b) have *dimension-dependent* parallel complexities which may achieve improvements in high-accuracy regimes, which are not the focus of this paper.

Matvec complexity. Importantly, our algorithms apply even in the setting where $\Phi_{\mathcal{D}}$ is unknown, but can be accessed with a matvec oracle. Until recently, the state-of-the-art matvec complexity for the separating hyperplane problem—among both deterministic and randomized algorithms—was $\tilde{O}(\rho^{-1})$ (Nesterov, 2007; Nemirovski, 2004; Rakhlin and Sridharan, 2013). However, very recently, a line of work (Kornowski and Shamir, 2025a; Karmarkar et al., 2025, 2026) studied the matvec complexity of a broader class of problems, known as ℓ_2-ℓ_1 and ℓ_1-ℓ_1 *matrix games*, the former of which encompasses the separating hyperplane problem (as we discuss further in Section 3). These works show that $\tilde{\Theta}(\rho^{-2/3})$ matvecs to $\Phi_{\mathcal{D}}$ is necessary and sufficient for deterministic algorithms.

While these works settle the deterministic matvec complexity of the separating hyperplane problem (up to polylogarithmic factors), Karmarkar et al. (2025, 2026) do not analyze other notions of computational complexity. In other words, obtaining faster algorithms or algorithms with improved depth were left as directions for future research. Hence, it is perhaps natural to wonder whether these recent information theoretic improvements come at the cost of greater work.

Our results indicate that this is not necessarily the case (at least for randomized algorithms). Indeed, Theorems 2 and 3 demonstrate that, if one is willing to use randomness, then $\tilde{O}(\rho^{-2/3})$ matvec complexities can be obtained at the cost of only $\mathsf{poly}(\rho^{-1})$-overhead in work.

Additional related work on matrix games. As mentioned above, the separating hyperplane problem can be reduced to the problem of computing an ϵ-solution of an ℓ_2-ℓ_1 matrix game, which we define formally in Section 3. Our work builds upon a rich line of work on matrix games more broadly (Carmon et al., 2019, 2020b, 2024a; Karmarkar et al., 2025, 2026; Kornowski and Shamir, 2025a,b). As our focus is on the separating hyperplane problem, Table 1 compares against works which make the natural normalizing assumption that each feature vector has Euclidean norm at most 1 (i.e., $\|\Phi_{\mathcal{D}}\|_{2\to\infty} = \max_{i\in[n]} \|\phi_i\|_2 \leq 1$). However, (Carmon et al., 2020b, 2019) obtain alternative total work complexities under alternative normalization assumptions on the underlying data matrix $\Phi_{\mathcal{D}}$. Additionally, alternative total work complexities can be obtained when the rows and columns of $\Phi_{\mathcal{D}}$ are uniformly sparse, and we defer to (Carmon et al., 2020b; Clarkson et al., 2012) for further discussion.

Paper organization. We provide notation in Section 2, an overview of our approach in Section 3, and conclude in Section 4. In Appendix A, we leverage ball acceleration techniques (Carmon et al., 2020a, 2021, 2023, 2024a) to show that the separating hyperplane problem can be reduced to solving a sequence of regularized linear systems to high accuracy. In Appendix B, we discuss how to solve these linear systems efficiently, leveraging subspace embeddings and preconditioned iterative solvers (Nelson and Nguyên, 2013; Cohen, 2016; Chenakkod et al., 2024a,b; Derezinski et al., 2025; Derezinski and Sidford, 2026). In Appendix C, we show how to leverage the sample reuse framework of (Jin et al., 2026) to further improve efficiency.

2. Preliminaries

General notation. For $x \in \mathbb{R}^d$, $\|x\|_p$ is its ℓ_p norm and x_i or $[x]_i$ its i-th entry (we may use the latter in particular when there are multiple subscripts). We let $\mathbb{B}_r^d(\bar{x}) := \{x \in \mathbb{R}^d : \|x - \bar{x}\|_2 \leq r\}$ denote the Euclidean ball of radius $r > 0$ centered at $\bar{x} \in \mathbb{R}^d$. We may drop r if $r = 1$ or $\bar{x}$ if $\bar{x} = 0$ for brevity. Given $S \subseteq \mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}$, and $\epsilon > 0$, we say that $x \in S$ is ϵ-optimal or an ϵ-minimizer of f over S if $f(x) - f(x') \leq \epsilon$ for all $x' \in S$. For vectors $a, b \in \mathbb{R}^n$ and $r \geq 1$, we use $a \approx_r b$ to denote that $r^{-1}[b]_i \leq [a]_i \leq r[b]_i$ for all $i \in [n]$.

Matrix notation. For $A \in \mathbb{R}^{n \times d}$ of rank r, we let $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_r(A) \geq \sigma_{r+1}(A) = \cdots = \sigma_d(A) = 0$ denote its singular values in descending order. 0_d and $0_{m \times n}$ denote the d-dimensional vector of zeros and $m \times n$ matrix of zeros respectively and I_d denotes the $d \times d$ identity matrix; we may drop the subscript d when it is clear from context. For $v \in \mathbb{R}^d$, $\mathrm{diag}(v) \in \mathbb{R}^{d \times d}$ denotes the diagonal matrix whose (i, i)-th entry is $[v]_i$.

We use $\mathbb{S}^d := \{A \in \mathbb{R}^{d \times d} : A = A^\top\}$ to denote the set of $d \times d$ symmetric matrices. For $A \in \mathbb{S}^d$, we say that A is positive semi-definite (PSD), denoted $A \succeq 0_d$, if for all $x \in \mathbb{R}^d$, $x^\top A x \geq 0$; if, moreover, $x^\top A x > 0$ for all $x \in \mathbb{R}^d_{\neq 0}$, we say that A is positive definite (PD), denoted $A \succ 0_d$. Correspondingly, we define $\mathbb{S}^d_+ := \{A \in \mathbb{S}^d : A \succeq 0_d\}$ and $\mathbb{S}^d_{++} := \{A \in \mathbb{S}^d : A \succ 0_d\}$.

For $M \in \mathbb{S}^n_{++}$ and $x \in \mathbb{R}^n$, we let $\|x\|_M := \sqrt{x^\top M x}$ denote the M-norm of x. For $A, B \in \mathbb{S}^d$, $A \succeq B$ (respectively $A \succ B$) denotes that $A - B \succeq 0_d$ (respectively $A - B \succ 0_d$) and for $c \geq 1$, $A \approx_c B$ denotes that $c^{-1} A \preceq B \preceq cA$. For any matrix $A \in \mathbb{S}^n_+$, we use $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ to denote its smallest and largest eigenvalues, respectively and $\kappa(A) := \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ to denote its *condition number*.

We use the following standard definition of a linear system solver for a PD matrix.

Definition 4 (Linear system solutions and solvers) *For $M \in \mathbb{S}^d_{++}$, $\epsilon \geq 0$, $\hat{x} \in \mathbb{R}^d$ is an ϵ-*approximate solution *to the linear system $Mx = b$ if $\|\hat{x} - x\|_M \leq \epsilon \|x\|_M$. Moreover, a (randomized) oracle $\mathcal{O}$ is an (ϵ, δ)-*linear system solver *for M if for any $b \in \mathbb{R}^d$, $x \leftarrow \mathcal{O}(b)$ is, with probability $1 - \delta$, an ϵ-approximate solution to $Mx = b$.*

3. Technical overview

In this section, we describe our approach and motivate our techniques based on prior work.

3.1. Reduction to ℓ_2-ℓ_1 matrix games

The first step to obtain our results is to leverage that the ρ-separating hyperplane problem (Definition 1) can be reduced to a class of problems known as ℓ_2-ℓ_1 *matrix games* (henceforth referred to simply as *games*, for brevity) (Carmon et al., 2019, 2020b, 2024a; Kornowski and Shamir, 2025a; Karmarkar et al., 2026).

Definition 5 ((ℓ_2-ℓ_1 matrix) game) *In an* ϵ-(ℓ_2-ℓ_1 matrix) game*, we are given a matrix* $A \in \mathbb{R}^{n \times d}$ *such that* $\|A\|_{2\to\infty} \coloneqq \max_{i\in[n]} \|A_{i,:}\|_2 \le 1$ *and* $\epsilon > 0$ *and must compute* $\hat{x} \in \mathbb{B}^d$ *such that*

$$\max_{y\in\Delta^n} y^\top A\hat{x} \le \min_{x\in\mathbb{B}^d} \max_{y\in\Delta^n} y^\top Ax + \epsilon.$$

We call such an $\hat{x}$ *a (primal) solution of the* ϵ*-game of* A*.*[2]

The following Lemma 6 shows that the separating hyperplane problem (Definition 1) can be reduced to solving an appropriate ρ-game (Definition 5); this result is well-known, however, we include a proof in Appendix D for completeness.

Lemma 6 (Reduction to ℓ_2-ℓ_1 matrix game)
Let $\mathcal{D} = \{\phi_i \in \mathbb{B}^d, l_i \in \{+1, -1\}\}_{i\in[n]}$*. Any solution* $\hat{w}$ *to the* ρ*-game (Definition 5) of* $-\Phi_{\mathcal{D}}$ *induces a* ρ*-separating hyperplane (Definition 1) for* $\mathcal{D}$*.*

The algorithmic techniques we develop in this paper apply to ℓ_2-ℓ_1 matrix games more broadly. Consequently, in the remainder of the paper we focus on designing efficient algorithms for the more general problem of solving an ϵ-game. In the remainder of this section, we fix $\epsilon > 0$, $A \in \mathbb{R}^{n\times d}$ and discuss our approach for solving the corresponding ϵ-game (Definition 5).

3.2. Motivation

Our work is motivated in part by recent algorithmic advancements of Karmarkar et al. (2025, 2026), which culminated in a near-optimal (Kornowski and Shamir, 2025a) $\tilde{O}(\epsilon^{-2/3})$-matvec complexity algorithm for solving an ϵ-game (Definition 5). At a high level, Karmarkar et al. (2025, 2026) develop a primal-dual analog of accelerated ball-constrained optimization (Carmon et al., 2020a, 2024a, 2023; Asi et al., 2021b) to reduce solving an ϵ-game to a sequence of $\tilde{O}(\epsilon^{-2/3})$ constrained, regularized sub-problems of the form

$$\min_{x\in\mathbb{B}^d} \max_{y\in\Delta^n} y^\top Ax + \alpha^{(t)} \sum_{x'\in\mathcal{U}_{\mathsf{x}}} \frac{1}{2}\|x - x'\|_2^2 - \alpha^{(t)} \sum_{y'\in\mathcal{U}_{\mathsf{y}}} \mathrm{KL}(y||y') \tag{2}$$

to high accuracy, for carefully chosen finite (multi)sets of *centers* $\mathcal{U}_{\mathsf{x}} \subset \mathcal{X}, \mathcal{U}_{\mathsf{y}} \subset \mathcal{Y}$ and radii r. Here, $\mathrm{KL}(y||y') \coloneqq \sum_{i\in[n]} [y]_i \log \frac{[y]_i}{[y']_i}$ denotes the KL divergence.

To solve (2) deterministically and query-efficiently (i.e., with few matvecs to A), (Karmarkar et al., 2025, 2026) show that each subproblem of the form (2) can be further reduced to $\tilde{O}(1)$ carefully constructed *constrained subproblems*, each corresponding to a constrained version of (2). They show that within the constrained region, the primal-dual objective is approximated by a quadratic up to a multiplicative constant. Their algorithm leverages this property to either build a low-rank approximation to A or make optimization progress on (2). The former enables the latter to be achieved with fewer matvecs, and ultimately the algorithm of Karmarkar et al. (2026) uses (amortized) $\tilde{O}(1)$ matvecs to A per iteration across $\tilde{O}(\epsilon^{-2/3})$ iterations.

2. For simplicity, as our focus is on the separating hyperplane problem (Definition 1), which only requires finding a hyperplane $\hat{w}$, we focus on finding a near-optimal primal variable $\hat{x}$ for the minimax problem $\min_{x\in\mathbb{B}^d} \max_{y\in\Delta^n} y^\top Ax$. However, using the techniques of (Carmon et al., 2024b), our method could be extended to extract near-optimal dual variables $\hat{y}$.

More concretely, their methods carefully build and maintain an explicit, deterministic $\tilde{O}(\epsilon^{-2/3})$-rank *model* M_c for the matrix $Y_c^{1/2}A$, where Y_c is a carefully chosen diagonal matrix, so that $Y_c^{1/2}A - M_c$ has a smaller operator norm than $Y_c^{1/2}A$, enabling more efficient optimization progress when combined with composite proximal methods. By using matvec queries to A to maintain this model as the center y_c changes, their method converges to a high-accuracy solution of (2) using an amortized $\tilde{O}(1)$ matvecs to A per subproblem of the form (2).

Importantly, although the model-maintenance procedure in (Karmarkar et al., 2026) achieves the optimal matvec complexity in A and avoids the use of randomness, they require additional computation. Hence, while their method enables near-optimal information theoretic complexity, it does not immediately imply improved work or depth. Moreover, the models M maintained in their method may be *dense* even if the original A is sparse, and consequently, even reading the model M may naively require $\tilde{O}((n+d)\epsilon^{-2/3})$-work. Thus it is perhaps unclear how to directly implement their method and obtain total work complexities scaling with $\mathrm{nnz}(A)$.

Our key insight is that if one is willing to use randomness, there is a natural alternative to the (Karmarkar et al., 2026, 2025)'s low-rank model-building procedure described above. Indeed, (Karmarkar et al., 2025) allude to a version of this alternative approach in their introduction, albeit in a different context and towards a worse complexity bound. To motivate this idea, recall from above that in the method of (Karmarkar et al., 2026, 2025), the objective of each constrained sub-problem behaves essentially like a quadratic function over the constrained region. While (Karmarkar et al., 2025, 2026) leverage this property to build a deterministic algorithm, an alternative approach would be to leverage randomized linear-algebraic techniques such as oblivious subspace embeddings (Sarlos, 2006; Cohen et al., 2015; Clarkson and Woodruff, 2017; Nelson and Nguyên, 2013) and preconditioned iterative linear system solvers (Golub and Varga, 2007; Golub and Overton, 1988) to efficiently solve these constrained sub-problems (2). Moreover, as we discuss in Section 3.5 (and Section C), this approach is amenable to the *sample reuse framework* of (Jin et al., 2026), which allows us to reuse the same subspace embeddings accross all constrained sub-problems. Below, we formalize how this intuition motivates a straightforward *primal-only* approach to the problem that matches the matvec complexity of (Karmarkar et al., 2025, 2026) while improving over the total work complexities of the prior art (recall Table 1), albeit using randomness.

In the following sections, we discuss this approach in greater detail, and provide a sketch of the analysis which enables Theorems 2 and 3. As our method requires several linear-algebraic reductions, to anchor the discussion we include Figure 1 which summarizes the end-to-end implementation of our approach and analysis and the various reductions which enable it.

3.3. Reducing the ϵ-game to linear-system solving

Recall from Definition 5 that our goal is to compute an ϵ-minimizer of f, defined as $f(x) := \max_{y \in \Delta^n} y^\top Ax$, over $\mathbb{B}^d$. In Appendix A we show how to reduce this problem to a sequence of approximate linear-system solves, culminating in Lemma 7 below. While care is needed to handle approximations and mild extensions not handled in prior work, our overarching approach in this reduction is similar to that in prior works related to ball acceleration, e.g., Carmon et al. (2020a, 2021); Jambulapati et al. (2022).

To define these systems, for $x \in \mathbb{B}^d$ and $\eta > 0$, we let $p_{x,\eta} \in \Delta^n$ be defined via

$$[p_{x,\eta}]_i := \frac{\exp([Ax]_i/\eta)}{\sum_{j\in[n]} \exp([Ax]_j/\eta)} \text{ for all } i \in [n], \text{ and further define } P_{x,\eta} := \mathrm{diag}(p_{x,\eta}) \,. \quad (3)$$

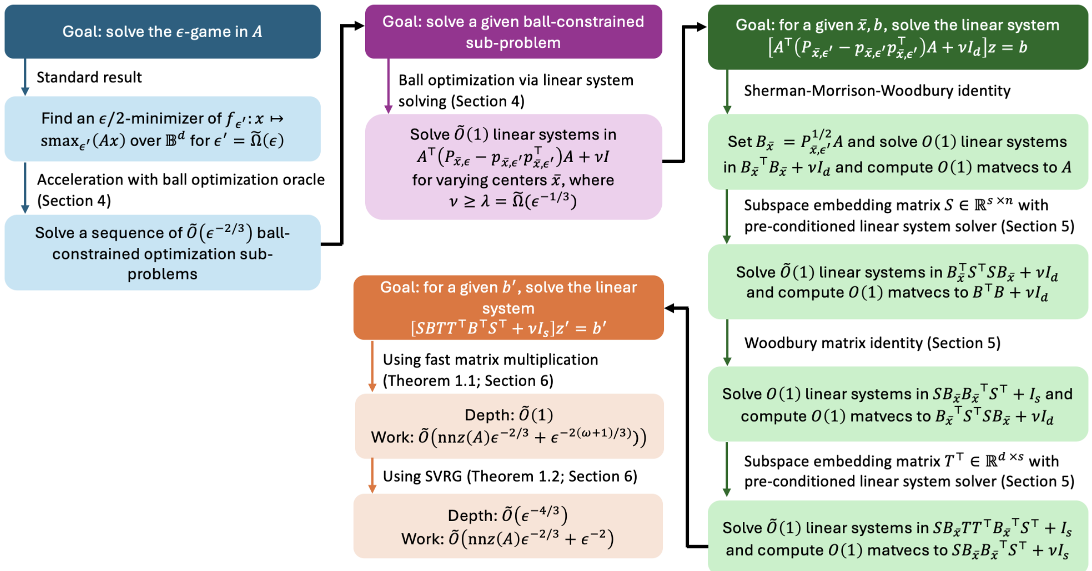

Figure 1: **Overview of approach.** Dark-colored boxes highlight the main problems and subproblems in the method. Solid colored arrows indicate reduction steps within a single problem, while solid black arrows denote transitions between nested problems. Dotted arrows represent two alternative algorithmic approaches for solving the resulting linear systems. Here, $S \in \mathbb{R}^{s\times n}$ and $T^\top \in \mathbb{R}^{s\times d}$ are oblivious subspace embeddings of size $s = \tilde{O}(\epsilon^{-2/3})$. Using a naive analysis, a fresh S, T would need to be sampled for each $\bar{x}$, however, using the sample reuse framework of (Jin et al., 2026), we show that the same T can be used across all iterations (Section 3.5).

Then informally, Lemma 7 says that to achieve the work, depth, and matvec complexities of Theorem 2 and Theorem 3, it suffices to solve $\tilde{O}(\epsilon^{-2/3})$ linear systems as needed in (4) efficiently in these complexity metrics. We note that Lemma 7 exposes the role of the random seed χ in order to enable application of the sample reuse framework of (Jin et al., 2026), as we discuss further in Section 3.5.

Lemma 7 *For $\epsilon > 0$, there is an algorithm that, with probability at least $3/4$ over the draw of a random seed $\chi \sim \mathcal{P}_{\text{seed}}$, returns $\tilde{x} \in \mathbb{B}^d$ such that $f(\tilde{x}) - \min_{x\in\mathbb{B}^d} f(x) \leq \epsilon$. The computational cost of the method is dominated by the cost of $\tilde{O}(\epsilon^{-2/3})$-iterations of performing the following:*

- *computing $x' \in \mathbb{R}^d$ which is a $(\text{poly}(\epsilon^{-1}, n))^{-1}$-approximate solution to $(H + \nu I)^{-1}y = \hat{g}$, where*

$$H = \frac{1}{\epsilon'} A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'}p_{\bar{x},\epsilon'}^\top)A \tag{4}$$

for some $\bar{x} \in \mathbb{B}^d$, $\nu = \tilde{\Omega}(\epsilon^{-1/3})$, and $\hat{g} \in \mathbb{R}^d$ (which may vary each time) with $\epsilon' := \frac{\epsilon}{2\log n}$,

- *$\tilde{O}(1)$-additional matvecs to A,*
- *$\tilde{O}(n + d)$-additional work, and $\tilde{O}(1)$-additional depth.*

Moreover, $\mathcal{P}_{\text{seed}}$ is independent of A, and $\chi \sim \mathcal{P}_{\text{seed}}$ can be sampled in $\tilde{O}(\epsilon^{-2/3})$-work and $\tilde{O}(1)$-depth.

To illustrate how Lemma 7 reduces our problem to approximately solving linear systems, recalling that $\text{nnz}(A) = \Omega(n+d)$, to prove Theorem 2 using Lemma 7 it suffices to implement (4) in $\tilde{O}(\text{nnz}(A) + \epsilon^{-2\omega/3})$-work and $\tilde{O}(1)$-depth, using $\tilde{O}(1)$ matvecs to A. Likewise, to prove Theorem 3 using Lemma 7, it suffices to implement (4) in $\tilde{O}(\text{nnz}(A) + \epsilon^{-4/3})$-work, $\tilde{O}(\epsilon^{-2/3})$-depth, and using $\tilde{O}(1)$ matvecs to A.

Let us now discuss how we prove Lemma 7 in Appendix A. Our first step is to leverage a standard smoothing technique (Nesterov, 2005; Carmon et al., 2020a, 2021; Asi et al., 2021a) where the nonsmooth objective function f is replaced with a smooth proxy objective. Concretely for $\alpha > 0$, we define the *softmax* function $\text{smax}_\alpha : \mathbb{R}^n \to \mathbb{R}$ via

$$\text{smax}_\alpha(z) \coloneqq \max_{y' \in \Delta^n} z^\top y' - \alpha \cdot e(y') = \alpha \log \left(\sum_{i \in [n]} \exp([z]_i/\alpha) \right), \tag{5}$$

where $e(y) \coloneqq \sum_{i \in [n]} y_i \log y_i$ is the negative entropy function. Then, defining $f_\alpha \coloneqq \text{smax}_\alpha(Ax)$, it is a standard result (see Lemma 21) that f_α is an additive $\alpha \log n$ approximation of f, i.e., $|f_\alpha(x) - f(x)| \le \alpha \log n$ for all $x \in \mathbb{R}^d$, and furthermore f_α is $1/\alpha$-smooth in $\|\cdot\|_2$. Thus, defining $\epsilon' \coloneqq \epsilon/(2\log(n))$, any $\epsilon/2$-minimizer of $f_{\epsilon'}$ over $\mathbb{B}^d$ is an ϵ-minimizer of f over $\mathbb{B}^d$, and therefore we focus on obtaining the former.

To (approximately) minimize $f_{\epsilon'}$, we leverage the *ball-oracle acceleration* framework developed in a series of prior works Carmon et al. (2020a, 2021); Asi et al. (2021b); Carmon et al. (2022); Carmon and Hausler (2022); Carmon et al. (2023, 2024a). For $r \in (0,1)$, this framework reduces minimizing a convex and Lipschitz function $h : \mathbb{R}^d \to \mathbb{R}$ to minimizing a regularized version of h within a ball of radius r at most $\tilde{O}(r^{-2/3})$ times (see Proposition 23 for the formal statement). By choosing $r \leftarrow \tilde{\Theta}(\epsilon)$ and using the fact that f'_ϵ is 1-Lipschitz, we are able to reduce obtaining an $\epsilon/2$-minimizer of f'_ϵ to $\tilde{O}(\epsilon^{-2/3})$ computations, for $g_{\nu,\bar{x}}(x) \coloneqq f_{\epsilon'}(x) + \frac{\nu}{2}\|x - \bar{x}\|_2^2$, of $x' \in \mathbb{B}^d_r(\bar{x}) \cap \mathbb{B}^d$ such that

$$g_{\nu,\bar{x}}(x') - \min_{x \in \mathbb{B}^d_r(\bar{x}) \cap \mathbb{B}^d} g_{\nu,\bar{x}}(x) \le \tilde{O}(\nu \epsilon^{8/3}) \tag{6}$$

for some $\bar{x} \in \mathbb{B}^d$ and $\nu = \tilde{\Omega}(\epsilon^{-1/3})$ (which may vary each iteration); see Appendix A.3 for details. Furthermore, each of the $\tilde{O}(\epsilon^{-2/3})$ iterations requires no additional access to A, $\tilde{O}(d)$-additional work, and $\tilde{O}(1)$ additional depth, and therefore to prove Lemma 7 it suffices to implement (6) using $\tilde{O}(1)$ linear system solves of the form (4), $\tilde{O}(n+d)$-additional work, $\tilde{O}(1)$ additional matvecs to A, and $\tilde{O}(1)$-additional depth.

To achieve this, we leverage that $f_{\epsilon'}$ is $\tilde{O}(\epsilon^{-1})$-*quasi-self-concordant* Bach (2010); Sun and Tran-Dinh (2019); Karimireddy et al. (2018); Carmon et al. (2020a); Doikov (2023), which implies it is *c-Hessian stable* Carmon et al. (2020a); Karimireddy et al. (2018) for an absolute constant $c > 1$ in any ball of radius r, i.e., $c^{-1}\nabla^2 f_{\epsilon'}(y) \preceq \nabla^2 f_{\epsilon'}(x) \preceq c\nabla^2 f_{\epsilon'}(y)$ for any $x, y \in \mathbb{R}^d$ such that $\|x - y\|_2 \le r$. As a result, the smoothness and Hessian-stability of $f_{\epsilon'}$ imply $g_{\nu,\bar{x}}$ is also smooth and Hessian-stable. Combining these properties with the strong convexity of $g_{\nu,\bar{x}}$, we leverage an accelerated ball-constrained Newton algorithm due to (Carmon et al., 2020a, Alg. 3) to reduce (6) to solving $\tilde{O}(1)$ ball-constrained quadratic optimization problems, along with $\tilde{O}(1)$-additional matvecs to A, $\tilde{O}(d)$-additional work, and $\tilde{O}(1)$-additional depth.

More formally, the ball-constrained quadratics subproblems take the form

$$\underset{x\in\mathbb{B}_r(\bar{x})\cap\mathbb{B}^d}{\text{minimize}} -g^\top x + \frac{1}{2}x^\top(H+\nu I)x \tag{7}$$

for $H = \nabla^2 f_{\epsilon'}(\bar{x})$ as in (4). If (7) had a single ball constraint $x \in \mathbb{B}_r(\bar{x})$, it would be the well-studied *trust-region problem* Conn et al. (2000). In particular, Carmon et al. (2020a) show how to reduce approximately minimizing the version of (7) with a single ball constraint $x \in \mathbb{B}_r(\bar{x})$ to *exact* linear system solves in matrices of the form $H + \lambda I$ for $\lambda \geq \nu$ by binary searching on the (one-dimensional) Lagrange multiplier associated with the constraint $x \in \mathbb{B}_r(\bar{x})$. In Appendix A.1, we generalize this approach to (7) by performing a cutting plane method in the two-dimensional dual space of Lagrange multipliers, and carefully show that approximate linear system solves of the form (4) suffice to solve (7) to high accuracy.

Note that for any $\bar{x} \in \mathbb{B}^d$, one matvec to H (recall (4)) can be computed using $O(1)$ matvecs to A (recall also (3)). Thus, it remains to show how to implement an efficient linear system solver for (4). Next, we discuss how we leverage modern numerical linear algebra techniques to solve such linear systems with low work and depth.

3.4. Implementing linear system solvers using subspace embeddings

Here, we focus on solving linear systems of the form (4), by adapting techniques of (Derezinski et al., 2025; Derezinski and Sidford, 2026).

Rank-one reduction. First, note that $H = \nabla^2 f_{\epsilon'}(\bar{x}) = \frac{1}{\epsilon'}A^\top(P_{\bar{x}} - p_{\bar{x}}p_{\bar{x}}^\top)A$ is PSD (because $f_{\epsilon'}$ is convex) and is simply a *rank-one update* of $B_{\bar{x}}^\top B_{\bar{x}}$ for $B_{\bar{x}} := \frac{1}{\sqrt{\epsilon'}}P_{\bar{x},\epsilon'}^{1/2}A$. Consequently, using the Sherman-Morison-Woodbury identity, we show that in order to build a linear-system solver for $H + \nu I$ it suffices to build a linear-system solver for

$$\frac{1}{\epsilon'}A^\top P_{\bar{x},\epsilon'}A + \nu I_d = \frac{1}{\epsilon'}A^\top P_{\bar{x},\epsilon'}^{1/2}P_{\bar{x},\epsilon'}^{1/2}A + \nu I_d = B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d.$$

Concretely, we show (Lemma 37) that any linear system in $H + \nu I_d$ can be solved by instead solving one linear system in $B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d$ plus $O(d)$-additional work and using $\tilde{O}(1)$-additional depth.

Consequently, we turn our attention to building a linear-system solver for $B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d$. Here, as in prior work (Carmon and Hausler, 2022; Carmon et al., 2024a, 2020a; Carmon and Hausler, 2022), we observe that

$$\|B_{\bar{x}}\|_F^2 = \frac{1}{\epsilon'}\mathrm{tr}\left(A^\top P_{\bar{x},\epsilon'}A\right) = \frac{1}{\epsilon'}\sum_{i\in[n]}[p_{\bar{x},\epsilon'}]_i\|A_{i,:}\|_2^2 \leq \frac{1}{\epsilon'}\|A\|_{2\to\infty}^2 = \tilde{O}\left(\epsilon^{-1}\right). \tag{8}$$

As a result, we can leverage a numerical linear algebra tool called *oblivious random subspace embeddings* to build a good *preconditioner* for the matrix $B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d$.

Oblivious random subspace embeddings. There are several distributions of matrices $S \in \mathbb{R}^{s\times n}$ such that for any matrix $C \in \mathbb{R}^{n\times d}$ and $\nu > 0$, $SC \in \mathbb{R}^{s\times d}$ can be computed in nearly linear time and low parallel depth and, with high probability, for $s \leq d$ sufficiently large, $C^\top S^\top SC + \nu I_d \approx_2$

$C^\top C + \nu I_d$ (Chenakkod et al., 2024a,b; Cohen et al., 2015; Cohen, 2016; Nelson and Nguyên, 2013); that is, $C^\top S^\top SC + \nu I_d$ is a *low-rank* 2-approximation for $C^\top C + \nu I_d$. In particular, in this paper we use that there is a distribution of sparse matrices $S \in \mathbb{R}^{s\times n}$ (Definition 30) such that $s = \tilde{O}(\|C\|_F^2/\nu)$ suffices (Derezinski and Sidford, 2026; Chenakkod et al., 2024b) (Corollary 33). Consequently, in light of (8) and noting that $\nu = \tilde{\Omega}(\epsilon^{1/3})$ in (4), there is a distribution of matrices $S \in \mathbb{R}^{s\times n}$ such that $B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d \approx_2 B_{\bar{x}}^\top S^\top SB_{\bar{x}} + \nu I_d$ for $s = \tilde{O}(\epsilon^{-2/3})$ and moreover, $SB_{\bar{x}}$ can be computed with just $\tilde{O}(\epsilon^{-2/3})$-matvecs to A.

Preconditioned linear system solvers. Next, we use that for any $M \in \mathbb{S}^{d\times d}_{++}$, in order to build a linear system solver for M, it suffices to build a linear system solver for some $N \approx_2 M$ and apply preconditioned methods such as preconditioned Richardson iteration (Theorem 34) (Derezinski and Sidford, 2026; Golub and Overton, 1988). Preconditioned Richardson iteration runs $\tilde{O}(1)$ iterations, where each iteration makes one call to a linear system solver for N, one matvec to M, and performs $\tilde{O}(n + d)$-additional work using $\tilde{O}(1)$-additional depth. Consequently, in order to build a linear-system solver for $B_{\bar{x}}^\top B_{\bar{x}} + \nu I_d$, it suffices to build a linear-system solver for $B_{\bar{x}}^\top S^\top SB_{\bar{x}} + \nu I_d$, where $S \in \mathbb{R}^{s\times n}$ is sampled from an appropriate distribution of matrices with $s = \tilde{O}(\epsilon^{-2/3})$.

Our goal is now to efficiently implement a linear system solver for $B_{\bar{x}}^\top S^\top SB_{\bar{x}} + \nu I_d$. Note that this is a $d \times d$ linear system, involving the $s \times d$ matrix $SB_{\bar{x}} \in \mathbb{R}^{s\times d}$ (namely, the coefficient matrix is the regularized Gram matrix of $SB_{\bar{x}}$). To get improved complexities, we further reduce to solving a system involving an $s \times s$ matrix (as opposed to an $s \times d$ matrix). To achieve this, we use a Woodbury matrix identity, preconditioning, and a subspace embedding (similar to (Derezinski and Sidford, 2026; Derezinski et al., 2025)). We describe this approach in greater detail in the following paragraphs.

Changing the order of operations and applying a second subspace embedding. Using the Woodbury matrix identity, in order to build a linear system solver for $B_{\bar{x}}^\top S^\top SB_{\bar{x}} + \nu I_d$, it suffices to build a linear system solver for $SB_{\bar{x}}B_{\bar{x}}^\top S^\top + \nu I_s$ (Lemma 38). We then further observe that $\|SB_{\bar{x}}\|_F^2 \leq \tilde{O}(\epsilon^{-2/3})$ and consequently, we can once again sample another oblivious random subspace embedding matrix $T^\top \in \mathbb{R}^{s\times d}$ with $s = \tilde{O}(\epsilon^{-2/3})$, so that $SB_{\bar{x}}TT^\top B_{\bar{x}}^\top S^\top + \nu I_s \approx_2 SB_{\bar{x}}B_{\bar{x}}^\top S^\top + \nu I_s$. Now, once again using preconditioned Richardson iteration, in order to build linear system in $SB_{\bar{x}}B_{\bar{x}}^\top S^\top + \nu I_s$, it suffices to build a linear system solver for $SB_{\bar{x}}TT^\top B_{\bar{x}}^\top S^\top + \nu I_s$.

Note that we can compute $SB_{\bar{x}}T \in \mathbb{R}^{s\times s}$ explicitly with $\tilde{O}(\epsilon^{-2/3})$-matvecs to A in $\tilde{O}(\epsilon^{-2/3}\mathrm{nnz}(A))$-work and $\tilde{O}(1)$-depth.[3] Moreover, $SB_{\bar{x}}T$ is a $\tilde{O}(\epsilon^{-2/3})$-dimensional square linear system. Consequently, using FMM (Lemma 35) we can implement a $\tilde{O}(1)$-depth linear system solver for $SB_{\bar{x}}TT^\top B_{\bar{x}}^\top S^\top + \nu I_s$ which does $\tilde{O}(\epsilon^{-2\omega/3})$-work. This yields a total work of

$$\tilde{O}(\epsilon^{-2/3}) \cdot (s\,\mathrm{nnz}(A) + s^\omega) = \tilde{O}(\epsilon^{-4/3}\mathrm{nnz}(A) + \epsilon^{-2/3(\omega+1)}) \tag{9}$$

and a depth of $\tilde{O}(\epsilon^{-2/3})$.

Alternatively, at the cost of increased depth, there is an even faster approach for implementing a linear system solver for $SB_{\bar{x}}TT^\top B_{\bar{x}}^\top S^\top + \nu I_s$ using stochastic gradient methods, such as stochastic variance reduced gradient descent (SVRG) (Frostig et al., 2015; Lin et al., 2015; Johnson and Zhang, 2013). Using SVRG, we can implement a linear system solver for $SB_{\bar{x}}TT^\top B_{\bar{x}}^\top S^\top + \nu I_s$ in

3. If A is known explicitly, then $SB_{\bar{x}}T$ is computable in just $\tilde{O}(\mathrm{nnz}(A))$-time. However, in this paper we often assume A is accessible only through a matvec oracle, as our results apply to this setting with nearly no loss in complexity.

$\tilde{O}(\mathrm{nnz}(B_{\bar{x}}) + s\|B_{\bar{x}}\|_F^2/\nu) = \tilde{O}(\mathrm{nnz}(A) + s^2)$ work. Hence, recalling (8), we can reduce the work from (9) to

$$\tilde{O}(\epsilon^{-2/3}) \cdot \left(s\,\mathrm{nnz}(A) + s^{-2}\right) = \tilde{O}(\epsilon^{-4/3}\mathrm{nnz}(A) + \epsilon^{-2}); \tag{10}$$

however, the depth increases to $\tilde{O}(\epsilon^{-4/3})$ as SVRG requires sequential computation.

The above bounds in (9) and (10) almost match the total work claimed in Theorem 2 and Theorem 3, except that the leading term $\epsilon^{-4/3}\mathrm{nnz}(A)$ is too large by an $\epsilon^{-2/3}$ factor. Likewise, (9) and (10) correspond to a matvec complexity of $\tilde{O}(\epsilon^{-4/3})$, which is an $\epsilon^{-2/3}$ factor worse than prior work (Karmarkar et al., 2026). The bottleneck here is that, naively, for each ball-constrained subproblem, we must construct a *new* $SB_{\bar{x}}T \in \mathbb{R}^{s\times s}$ for freshly sampled oblivious random subspace embeddings S, T. Correspondingly, each subproblem naively requires $\tilde{O}(s)$-matvecs. Fortunately, we show that it is possible to reduce this overhead in the following section.

3.5. Applying the sample reuse framework

As mentioned above, naively, for each center $\bar{x}$ in the outer ball acceleration loop, computing

$$SB_{\bar{x}}T = S\frac{1}{\sqrt{\epsilon'}}P_{\bar{x}}^{1/2}AT \in \mathbb{R}^{s\times s}.$$

requires $\tilde{O}(\epsilon^{-2/3})$-matvecs to A, and then applying the diagonal rescaling $P_{\bar{x}}$. Because $\bar{x}$ is chosen adaptively by the outer ball-acceleration method, naive analysis suggests that each $\bar{x}$ (i.e., each linear system in Lemma 7) requires sampling a *fresh* oblivious random subspace embeddings S, T and then reconstructing $SB_{\bar{x}}T$.

Fortunately, by leveraging recent work on sample reuse (Jin et al., 2026), a more careful analysis indicates that this can be improved. As each linear system of the form (4) is solved to *high accuracy* and S, T are sampled *obliviously* (i.e., from a distribution which is independent of the particular $\bar{x}$), it is in fact possible to reuse the same subspace embeddings T for all ball acceleration centers $\bar{x}$ (i.e., each linear system in Lemma 7) at the cost of at most polylogarithmic overheads in work.

This allows us to by *pre-compute* AT using $\tilde{O}(\epsilon^{-2/3})$-matvecs to A and $\tilde{O}(\mathrm{nnz}(A)\epsilon^{-2/3})$-work and $\tilde{O}(1)$-depth (Fact 41). Then, for each iteration in Lemma 7, we can randomly sample a fresh S and build $SB_{\bar{x}}T$ in only $\tilde{O}(\mathrm{nnz}(A))$-work and $\tilde{O}(1)$-depth (Lemma 42). This lets us reduce the work in (9) down to the claimed runtime bound in Theorem 2. Analogously, using SVRG in place of FMM, we can reduce (10) down to the claimed runtime bound in Theorem 3.

We note that the recent sample reuse framework of (Jin et al., 2026) was provided quite generally and hence we use it blackbox. Consequently, the main novelty in our work is in the sequence of careful reductions which bring the original binary classification problem into a form where the framework of (Jin et al., 2026) immediately applies and yields our improved work bounds.

4. Conclusion

In this work, we obtain faster randomized algorithms for approximately computing maximum-margin separating hyperplanes for binary data classification. Our algorithms match the matvec complexity of prior work and offer improved parallel computational depth. A natural question left open by our work is whether it is possible to achieve the depth of Theorem 2 while matching the total work of Theorem 3, perhaps using very recent work of Karmarkar et al. (2026). Another natural problem is to try to extend our techniques to obtain improved work complexities or depth for other important classes of matrix games, such as zero-sum games. Lastly, it would be interesting to explore whether the work complexities or depth obtained in Theorems 2 or 3 can be obtained deterministically.

Acknowledgements

Thank you to anonymous reviewers for their helpful feedback and thank you to LLMs for various writing advice, latex formatting assistance, and help with proof-reading algebraic derivations. Aaron Sidford was supported in part by a Microsoft Research Faculty Fellowship, NSF Grant CCF1955039, and a PayPal research award. Liam was supported by an Amazon AI PhD Fellowship.

References

Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2005–2039. SIAM, 2025.

Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. *Advances in Neural Information Processing Systems*, 34:10810–10822, 2021a.

Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021b. Curran Associates Inc.

David S Atkinson and Pravin M Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69(1-3):1–43, 1995.

Francis Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4: 384–414, 2010.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. ISSN 0167-6377.

Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. *Complexity of highly parallel non-smooth convex optimization*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Yair Carmon and Danielle Hausler. Distributionally robust optimization via ball oracle acceleration. In *arXiv preprint arXiv:2203.13225*, 2022.

Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020a.

Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix games. In *61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2020b.

Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Thinking inside the ball: Near-optimal minimization of the maximal loss. In *34th Annual Conference on Computational Learning Theory (COLT)*, 2021.

Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive monteiro-svaiter acceleration. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

Yair Carmon, Arun Jambulapati, Yujia Jin, Yin Tat Lee, Daogao Liu, Aaron Sidford, and Kevin Tian. Resqueing parallel and private stochastic convex optimization. In *64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 2031–2058. IEEE, 2023.

Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. A whole new ball game: A primal accelerated method for matrix games and minimizing the maximum of smooth functions. In *35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024a.

Yair Carmon, Arun Jambulapati, Liam O'Carroll, and Aaron Sidford. Extracting dual solutions via primal optimizers. *arXiv preprint arXiv:2412.02949*, 2024b.

Shabarish Chenakkod, Michal Derezinski, and Xiaoyu Dong. Optimal oblivious subspace embeddings with near-optimal sparsity. *arXiv preprint arXiv:2411.08773*, 2024a.

Shabarish Chenakkod, Michal Derezinski, Xiaoyu Dong, and Mark Rudelson. Optimal embedding dimension for sparse subspace embeddings. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1106–1117, 2024b.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

Kenneth L Clarkson, Elad Hazan, and David P Woodruff. Sublinear optimization for machine learning. In *Journal of the ACM (JACM)*, 2012.

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

Michael B Cohen, Jelani Nelson, and David P Woodruff. Optimal approximate matrix product in terms of stable rank. *arXiv preprint arXiv:1507.02268*, 2015.

Michael B Cohen, Jonathan Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B Rao, and Aaron Sidford. Solving directed laplacian systems in nearly-linear time through sparse lu factorizations. In *2018 IEEE 59th annual symposium on foundations of computer science (FOCS)*, pages 898–909. IEEE, 2018.

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

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2000.

Michal Derezinski and Aaron Sidford. Approaching optimality for solving dense linear systems with low-rank structure. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 925–938. SIAM, 2026.

Michal Derezinski, 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 (SODA)*, pages 1972–2004. SIAM, 2025.

Nikita Doikov. Minimizing quasi-self-concordant functions by gradient regularization of newton method, 2023.

John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

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

Gene H Golub and Michael L Overton. The convergence of inexact chebyshev and richardson iterative methods for solving linear systems. *Numerische Mathematik*, 53(5):571–593, 1988.

Gene H Golub and Richard S Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Milestones in Matrix Computation-Selected Works of Gene H. Golub, with Commentaries*, pages 45–67, 2007.

Michael D Grigoriadis and Leonid G Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. In *Operations Research Letters*, 1995.

Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Improved iteration complexities for overconstrained p-norm regression. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 529–542, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648.

Arun Jambulapati, Aaron Sidford, and Kevin Tian. Closing the computational-query depth gap in parallel stochastic convex optimization. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2608–2643. PMLR, 2024a.

Arun Jambulapati, Aaron Sidford, and Kevin Tian. Closing the computational-query depth gap in parallel stochastic convex optimization. In *Annual Conference Computational Learning Theory*, 2024b.

Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 944–953, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384284.

Yujia Jin, Ishani Karmarkar, Aaron Sidford, and Jiayi Wang. Reusing samples in variance reduction. In *Algorithmic Learning Theory*, 2026.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of newton's method without strong-convexity or lipschitz gradients. In *arXiv preprint arXiv:1806.00413*, 2018.

Ishani Karmarkar, Liam O'Carroll, and Aaron Sidford. Solving zero-sum games with fewer matrix-vector products. In *66th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2025.

Ishani Karmarkar, Liam O'Carroll, and Aaron Sidford. Solving matrix games with near-optimal matvec complexity. *58th Annual ACM Symposium on Theory of Computing (STOC)*, pages arXiv–2601, 2026.

Guy Kornowski and Ohad Shamir. The oracle complexity of simplex-based matrix games: Linear separability and nash equilibria. In *38th Annual Conference on Computational Learning Theory (COLT)*, 2025a.

Guy Kornowski and Ohad Shamir. The oracle complexity of simplex-based matrix games: Linear separability and nash equilibria. In *arXiv preprint 2412.06990 [v3]*, 2025b.

Yin Tat Lee, Aaron Sidford, and Sam Chiu-Wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065, 2015. doi: 10.1109/FOCS.2015.68.

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

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. In *The bulletin of mathematical biophysics*, 1943.

MN Murty and Rashmi Raghava. Kernel-based svm. In *Support vector machines and perceptrons: Learning, optimization, classification, and application to social networks*, pages 57–67. Springer, 2016.

Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 ieee 54th annual symposium on foundations of computer science*, pages 117–126. IEEE, 2013.

Arkadi Nemirovski. On parallel complexity of nonsmooth convex optimization. *Journal of Complexity*, 10(4):451–463, 1994.

Arkadi Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. In *Wiley-Interscience*, 1983.

Yu Nesterov. Smooth minimization of non-smooth functions. In *Mathematical programming*, 2005.

Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344, Mar 2007.

Balamurugan Palaniappan and Francis Bach. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

Victor Y. Pan. Complexity of parallel matrix computations. *Theoretical Computer Science*, 54:65–85, 1987.

Victor Y. Pan and John H. Reif. Efficient parallel solution of linear systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 143–152. ACM, 1985.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 993–1019. PMLR, 2013.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. In *Psychological review*, 1958.

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

Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.

Pravin M. Vaidya. A new algorithm for minimizing convex functions over convex sets (extended abstract). In *FOCS*, pages 338–343, 1989.

Jan Van Den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and l1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021.

Appendix A. Reducing ℓ_2-ℓ_1-games to linear system solving

In this section, we reduce solving an ϵ-game to approximately solving a sequence of linear systems. In Appendix A.1, we show how to reduce solving a quadratic (constrained to two balls) to approximate linear system solves. Then in Appendix A.2, we combine the results of Appendix A.1 with an accelerated ball-constrained Newton method due to Carmon et al. (2020a) to reduce optimizing a ball-constrained Hessian-stable function to approximate linear system solves. Finally, in Appendix A.3, we reduce solving an ϵ-game to approximately solving a sequence of linear systems via the ball oracle acceleration framework developed in Carmon et al. (2020a, 2021); Asi et al. (2021b); Carmon et al. (2022); Carmon and Hausler (2022); Carmon et al. (2023, 2024a).

Notation for Appendix A. For a closed convex set $S \subseteq \mathbb{R}^d$ and $x \in \mathbb{R}^d$, $\mathrm{proj}_S(x) := \mathrm{argmin}_{y\in S}\|y-x\|_2$ denotes the Euclidean projection of x onto S. We define the Euclidean distance between x and S via $\mathrm{dist}(x, S) := \|x - \mathrm{proj}_S(x)\|_2$.

A.1. Ball-constrained quadratics via linear system solves

In this section, we give an algorithm for approximately solving the optimization problem

$$\begin{aligned} \underset{x\in\mathbb{R}^d}{\text{minimize}} \quad & -g^\top x + \frac{1}{2}x^\top Hx \\ \text{subject to} \quad & \|x-u_1\|_2 \le r_1 \text{ and } \|x-u_2\|_2 \le r_2 \end{aligned} \tag{11}$$

for $g, u_1, u_2 \in \mathbb{R}^d$, $r_1, r_2 > 0$, and $H \in \mathbb{S}^{d\times d}_{++}$ where $\mu I \preceq H \preceq LI$ for some $0 < \mu \le L$. We let $\kappa := L/\mu$ and $f(x) := -g^\top x + \frac{1}{2}x^\top Hx$ denote the objective function and $C_1 := \{x' \in \mathbb{R}^d : \|x' - u_1\|_2 \le r_1\}$ and $C_2 := \{x' \in \mathbb{R}^d : \|x' - u_2\|_2 \le r_2\}$ denote the constraint balls. Additionally, we assume that $u_2 \in C_1$ (the second center is contained within the first ball). Thus, Slater's condition is satisfied, strong duality holds, and we let $x^\star$ denote the unique minimizer of (11). For $\lambda = (\lambda_1, \lambda_2)$, the Lagrangian (after squaring both sides of the constraints) is given by

$$\mathcal{L}(x,\lambda) = -g^\top x + \frac{1}{2}x^\top Hx + \lambda_1(\|x-u_1\|_2^2 - r_1^2) + \lambda_2(\|x-u_2\|_2^2 - r_2^2)\,,$$

and we can express the dual optimization problem as

$$\underset{\lambda\ge 0}{\text{maximize}}\{Q(\lambda) := \min_{x\in\mathbb{R}^d} \mathcal{L}(x,\lambda)\}\,. \tag{12}$$

Letting $x^\star(\lambda) := \mathrm{argmin}_{x\in\mathbb{R}^d}\mathcal{L}(x,\lambda)$ denote the best response x for some fixed choice of $\lambda \ge 0$, note

$$x^\star(\lambda) = (H + 2(\lambda_1+\lambda_2)I)^{-1}(g + 2\lambda_1 u_1 + 2\lambda_2 u_2)\,. \tag{13}$$

Furthermore, the uniqueness of $x^\star(\lambda)$ implies by the envelope theorem that for $\lambda \ge 0$,

$$\nabla_\lambda Q(\lambda) = \nabla_\lambda \mathcal{L}(x^\star(\lambda),\lambda) = (\|x^\star(\lambda)-u_1\|_2^2 - r_1^2,\ \|x^\star(\lambda)-u_2\|_2^2 - r_2^2)\,. \tag{14}$$

At a high level, our algorithm to solve (11), whose guarantee is given at the end of this section in Proposition 15, implements a cutting plane method in the two-dimensional dual space. Indeed, note that a dual gradient (14) can be computed via a linear system solve (13). Once we obtain an (approximately) optimal dual solution, we can perform an additional (approximate) linear system solve (13) to obtain an (approximately) optimal primal solution. In the remainder of this section, we first prove a variety of regularity properties to enable these approximations. To start, we bound the maximum size of the dual optimal variables in the following lemma.

Lemma 8 *Any optimal pair of dual variables* $\lambda^\star \in \operatorname{argmax}_{\lambda \geq 0} Q(\lambda)$ *satisfies*

$$\max\{\lambda_1^\star, \lambda_2^\star\} \leq \frac{L\|x^\star\|_2 + \|g\|_2}{2\min\{r_1, r_2\}} \leq \frac{L(\|u_1\|_2 + r_1) + \|g\|_2}{2\min\{r_1, r_2\}} =: R_Q\,. \tag{15}$$

Proof First, note that

$$\|\nabla f(x^\star)\|_2 = \|Hx^\star - g\|_2 \leq \|H\|_2\|x^\star\|_2 + \|g\|_2 \leq L\|x^\star\|_2 + \|g\|_2\,. \tag{16}$$

Furthermore, the KKT stationary condition for this problem can be expressed as

$$-\nabla f(x^\star) = 2\lambda_1^\star n_1 + 2\lambda_2^\star n_2 \tag{17}$$

for $n_1 := x^\star - u_1$ and $n_2 := x^\star - u_2$. We now consider cases based on which of the two constraints are tight at $x^\star$. If neither constraint is tight, then $\lambda_1^\star = \lambda_2^\star = 0$ and we are done. If the first constraint in (11) is tight and the second is loose so that $\|n_1\|_2 = r_1$ and $\lambda_2^\star = 0$, then taking the norm of both sides of (17) yields $\lambda_1^\star = \|\nabla f(x^\star)\|_2/(2r_1)$, in which case we conclude by (16). Similarly, if the first constraint is loose and the second is tight, we have $\lambda_1^\star = 0$ and $\lambda_2^\star = \|\nabla f(x^\star)\|_2/(2r_2)$.

Now suppose that both constraints are tight at $x^\star$, implying $\|n_1\|_2 = r_1$ and $\|n_2\|_2 = r_2$. Taking the norm of both sides of (17) and squaring, we obtain

$$\|\nabla f(x^\star)\|_2^2 = (2\lambda_1^\star r_1)^2 + (2\lambda_2^\star r_2)^2 + 8\lambda_1^\star\lambda_2^\star(n_1^\top n_2)\,.$$

Then by (16), it suffices to show $8\lambda_1^\star\lambda_2^\star(n_1^\top n_2) \geq 0$ (as that would imply $\max\{(2\lambda_1^\star r_1)^2, (2\lambda_2^\star r_2)^2\} \leq \|\nabla f(x^\star)\|_2^2$), in which case it suffices to show $n_1^\top n_2 \geq 0$. Consider the triangle formed by the points $x^\star, u_1, u_2$, and letting θ denote the angle at the vertex $x^\star$, we have $\cos\theta = \frac{n_1^\top n_2}{\|n_1\|\|n_2\|}$, so it suffices to show $\cos\theta \geq 0$. By the law of cosines,

$$\cos\theta = \frac{r_1^2 + r_2^2 - \|u_1 - u_2\|_2^2}{2r_1 r_2} \geq 0$$

since the fact that u_2 is within the first constraint by assumption implies $\|u_1 - u_2\|_2^2 \leq r_1^2$. ■

Next, we show that the best primal response to any dual variable is bounded.

Lemma 9 *For all* $\lambda \geq 0$, *we have*

$$\|x^\star(\lambda)\|_2 \leq \Upsilon \;\; \textit{where} \;\; \Upsilon := \max\{\|g\|_2/\mu, \|u_1\|_2, \|u_2\|_2\}\,. \tag{18}$$

Proof Letting $\Lambda := \lambda_1 + \lambda_2$, note $\|(H + 2\Lambda I)^{-1}\|_2 \leq \frac{1}{\mu + 2\Lambda}$, in which case (13) gives

$$\|x^\star(\lambda)\|_2 \leq \|(H + 2\Lambda I)^{-1}\|_2 \cdot \|g + 2\lambda_1 u_1 + 2\lambda_2 u_2\|_2 \leq \frac{\|g\|_2 + 2\Lambda\max\{\|u_1\|_2, \|u_2\|_2\}}{\mu + 2\Lambda} \leq \Upsilon\,.$$

The last inequality follows because for any $a, b, c \geq 0$, the function $h(t) := \frac{a+bt}{c+2t}$ is monotonic for $t \geq 0$, and thus can be bounded by the values it approaches at the extremes ($t = 0$ and $t \to \infty$). ■

In the following lemma, we show that the dual objective Q is Lipschitz.

Lemma 10 *Q is β_Q-Lipschitz in $\|\cdot\|_2$ over $\lambda \geq 0$ where, for Υ defined in (18),*

$$\beta_Q := \sqrt{2} \max_{i \in \{1,2\}} \left\{ (\Upsilon + \|u_i\|_2)^2 + r_i^2 \right\}. \tag{19}$$

Proof It suffices to show $\|\nabla Q(\lambda)\|_2 \leq \beta_Q$ for all $\lambda \geq 0$. Then recalling in general that $\|v\|_2 \leq \sqrt{\ell}\|v\|_\infty$ for $v \in \mathbb{R}^\ell$, it suffices (by applying the inequality with $\ell \leftarrow 2$) to show $|[\nabla Q(\lambda)]_i| \leq (\Upsilon + \|u_i\|_2)^2 + r_i^2$ for $i \in \{1, 2\}$. This follows by combining (14), (18), and a triangle inequality. ■

The next lemma shows that an approximate linear system solve, namely obtaining x such that $\|x - x^\star(\lambda)\|_2 \leq \gamma$, suffices to obtain an approximate dual gradient $\tilde{q}$.

Lemma 11 *For any $\lambda \in \mathbb{R}^2_{>0}$, let $B_\lambda := H + 2(\lambda_1 + \lambda_2)I$. Suppose that x is a γ-approximate solution to the linear system (Definition 4)*

$$(H + 2(\lambda_1 + \lambda_2)I)y = (g + 2\lambda_1 u_1 + 2\lambda_2 u_2). \tag{20}$$

Then, for Υ as defined in (18), let $\bar{x} := \mathrm{proj}_{\{z : \|z\|_2 \leq \Upsilon\}}(x)$. Then,

$$\tilde{q} := \left(\|\bar{x} - u_1\|_2^2 - r_1^2,\ \|\bar{x} - u_2\|_2^2 - r_2^2 \right) \text{ satisfies } \|\tilde{q} - \nabla Q(\lambda)\|_2 \leq 4\gamma\Upsilon^2\sqrt{2\kappa}.$$

Proof Recall from (13) that $x^\star(\lambda)$ is the exact solution to (20). Therefore, since x is a γ-approximate solution to this linear system (Definition 4), letting $B_\lambda := (H + 2(\lambda_1 + \lambda_2)I)$, we have

$$\|x - x^\star(\lambda)\|_{B_\lambda} \leq \gamma \|x^\star(\lambda)\|_{B_\lambda}.$$

Since $B_\lambda \succeq \lambda_{\min}(B_\lambda)I$, it follows from Fact 48 that

$$\|x - x^\star(\lambda)\|_2 \leq \gamma\sqrt{\kappa(B_\lambda)}\|x^\star(\lambda)\|_2.$$

Since

$$\lambda_{\max}(B_\lambda) \leq L + 2(\lambda_1 + \lambda_2) \qquad \text{and} \qquad \lambda_{\min}(B_\lambda) \geq \mu + 2(\lambda_1 + \lambda_2),$$

and $\mu \leq L$, we have

$$\frac{\lambda_{\max}(B_\lambda)}{\lambda_{\min}(B_\lambda)} \leq \frac{L + 2(\lambda_1 + \lambda_2)}{\mu + 2(\lambda_1 + \lambda_2)} \leq \frac{L}{\mu} = \kappa.$$

Using $\|x^\star(\lambda)\|_2 \leq \Upsilon$ from (18), we obtain

$$\|x - x^\star(\lambda)\|_2 \leq \gamma\sqrt{\kappa}\,\Upsilon.$$

Next, since $x^\star(\lambda)$ lies in the ℓ_2-ball of radius Υ and $\bar{x}$ is the Euclidean projection of x onto this ball, projection can only decrease the distance to $x^\star(\lambda)$ in the ℓ_2-norm (due to the nonexpansiveness of projections onto nonempty, closed, convex sets). Consequently,

$$\|\bar{x} - x^\star(\lambda)\|_2 \leq \|x - x^\star(\lambda)\|_2 \leq \gamma\sqrt{\kappa}\,\Upsilon$$

Recalling (14), for each $i \in \{1, 2\}$,

$$\begin{aligned}[\tilde{q} - \nabla Q(\lambda)]_i &= \left(\|\bar{x} - u_i\|_2^2 - r_i^2\right) - \left(\|x^\star(\lambda) - u_i\|_2^2 - r_i^2\right) \\ &= \|\bar{x} - u_i\|_2^2 - \|x^\star(\lambda) - u_i\|_2^2 \\ &= \langle \bar{x} - x^\star(\lambda),\, \bar{x} + x^\star(\lambda) - 2u_i \rangle\,.\end{aligned}$$

By the Cauchy-Schwarz inequality and the triangle inequality,

$$\begin{aligned}|[\tilde{q} - \nabla Q(\lambda)]_i| &\le \|\bar{x} - x^\star(\lambda)\|_2 \|\bar{x} + x^\star(\lambda) - 2u_i\|_2 \\ &\le \gamma\sqrt{\kappa}\,\Upsilon\left(\|\bar{x}\|_2 + \|x^\star(\lambda)\|_2 + 2\|u_i\|_2\right).\end{aligned}$$

By definition of $\bar{x}$, we have $\|\bar{x}\|_2 \le \Upsilon$. Moreover, by the choice of Υ, we have $\|x^\star(\lambda)\|_2 \le \Upsilon$ and $\|u_i\|_2 \le \Upsilon$. Therefore,

$$|[\tilde{q} - \nabla Q(\lambda)]_i| \le \gamma\sqrt{\frac{L}{\mu}}\,\Upsilon\left(\Upsilon + \Upsilon + 2\Upsilon\right) = 4\gamma\sqrt{\kappa}\,\Upsilon^2.$$

Thus, $\|\tilde{q} - \nabla Q(\lambda)\|_\infty \le 4\gamma\sqrt{\frac{L}{\mu}}\,\Upsilon^2$. Since $\tilde{q} - \nabla Q(\lambda) \in \mathbb{R}^2$,

$$\|\tilde{q} - \nabla Q(\lambda)\|_2 \le \sqrt{2}\|\tilde{q} - \nabla Q(\lambda)\|_\infty \le 4\sqrt{2\kappa}\,\gamma\,\Upsilon^2.$$

∎

In the next lemma, we show that if we are able to obtain an approximately optimal dual solution λ, then the primal best response $x^\star(\lambda)$ is close to the true primal optimum $x^\star$.

Lemma 12 *Letting $\zeta > 0$ and $\lambda \ge 0$ be such that $\max_{\lambda' \ge 0} Q(\lambda') - Q(\lambda) \le \zeta$, we have $\|x^\star(\lambda) - x^\star\|_2 \le \sqrt{2\zeta/\mu}$.*

Proof Note that $x \mapsto \mathcal{L}(x, \lambda)$ is μ-strongly convex and minimized at $x^\star(\lambda)$, and hence for all $x \in \mathbb{R}^d$,

$$\mathcal{L}(x, \lambda) \ge \mathcal{L}(x^\star(\lambda), \lambda) + \frac{\mu}{2}\|x - x^\star(\lambda)\|_2^2\,.$$

Plugging in $x \leftarrow x^\star$ and $\mathcal{L}(x^\star(\lambda), \lambda) = Q(\lambda)$ and rearranging, we obtain the result via

$$\frac{\mu}{2}\|x^\star - x^\star(\lambda)\|_2^2 \le \mathcal{L}(x^\star, \lambda) - Q(\lambda) \le \max_{\lambda' \ge 0} \mathcal{L}(x^\star, \lambda') - Q(\lambda) = \max_{\lambda' \ge 0} Q(\lambda') - Q(\lambda) \le \zeta$$

by strong duality. ∎

Since our goal is to obtain an exactly feasible approximately optimal solution to (11) and we only have access to an approximate linear system solver, it is necessary to show that projecting onto the feasible region does not increase the distance to the optimum too much. We prove this in the following lemma.

Lemma 13 *Let $\alpha_1, \alpha_2 \ge 0$ and $x, y \in \mathbb{R}^d$ be such that $\max_{i \in \{1,2\}} \operatorname{dist}(y, C_i) \le \alpha_1$ and $\|x - y\|_2 \le \alpha_2$. Then defining $x' := \operatorname{proj}_{C_1}(x)$ and $x'' := \operatorname{proj}_{C_2}(x')$, we have $x'' \in C_1 \cap C_2$ and $\|x'' - y\|_2 \le 2\alpha_1 + \alpha_2$.*

Proof Note $x'' \in C_2$ by definition, and $x'' \in C_1$ since x'' is on the line segment connecting $x' \in C_1$ and $u_2 \in C_1$. We obtain the second result, recalling the non-expansiveness of the projection operator, by combining

$$\|x'' - y\|_2 \le \|x'' - \text{proj}_{C_2}(y)\|_2 + \|\text{proj}_{C_2}(y) - y\|_2 \le \|x' - y\|_2 + \alpha_1$$

with

$$\|x' - y\|_2 \le \|x' - \text{proj}_{C_1}(y)\|_2 + \|\text{proj}_{C_1}(y) - y\|_2 \le \|x - y\|_2 + \alpha_1 \le \alpha_2 + \alpha_1 \,.$$

■

In the next lemma, we state a guarantee for a cutting plane method due to Sidford and Zhang (2023) which allows for approximate gradients.

Proposition 14 *For $R > 0$ and $S := \{x \in \mathbb{R}^n_{\ge 0} : \|x\|_\infty \le R\}$, let $h : S \to \mathbb{R}$ be convex, differentiable, and β-Lipschitz in $\|\cdot\|_2$. Suppose we can only access h through an oracle $\mathcal{O}_{h,\delta} : S \to \mathbb{R}^n$, which, for any query point $x \in S$, satisfies $\|\mathcal{O}_{h,\delta}(x) - \nabla h(x)\|_2 \le \delta$. Then there is an algorithm which obtains $x \in S$ such that $h(x) - \min_{z \in S} h(z) \le \epsilon$, and makes $\tilde{O}(n)$ queries to $\mathcal{O}_{h,\delta}$ for $\delta \leftarrow \tilde{\Theta}(\frac{\epsilon}{R\sqrt{n}})$ with $\mathsf{poly}(n)$ additional work, where $\tilde{O}(\cdot)$ hides polylog factors in $R, \beta, n, \epsilon^{-1}$.*

Proof The query bound to $\mathcal{O}_{h,\delta}$ follows from (Sidford and Zhang, 2023, Sec. 4), which gives a generic analysis under such an oracle, which they define in (Sidford and Zhang, 2023, Def. 5). (Namely, their analysis holds for a variety of standard cutting plane algorithms.) Whereas they do not state their guarantees in terms of this oracle directly (they primarily study a stochastic setting where $\mathcal{O}_{h,\delta}$ is an intermediate oracle implemented with high probability), it is straightforward to obtain this bound from the proofs of (Sidford and Zhang, 2023, Prop. 1) and (Sidford and Zhang, 2023, Prop. 2). As for the additional work, this is achieved by a variety of deterministic cutting plane methods, e.g., Vaidya (1989); Atkinson and Vaidya (1995). (Note that the more recent cutting plane methods of Lee et al. (2015); Jiang et al. (2020) are stochastic.) ■

Finally, we state our main guarantee for this section below. As discussed above, we obtain our guarantee by applying a cutting plane method in the two-dimensional dual space, using an approximate linear system solver to obtain approximate dual gradients, and then obtaining an approximately optimal primal solution via an additional approximate linear system solve. Finally, we project onto the feasible region of (11) and argue that doing so does not increase the suboptimality too much.

Proposition 15 *For $\Delta > 0$, there is an algorithm which obtains $\tilde{x} \in \mathbb{R}^d$ which is feasible for (11) and satisfies $\|\tilde{x} - x^\star\|_2 \le \Delta$. It requires $\tilde{O}(1)$ computations of $x' \in \mathbb{R}^d$ such that, for R_Q defined in (15), x' is a*

$$\Theta\left(\Delta \cdot \min\left\{1, \sqrt{\frac{\mu}{R_Q}}, \frac{\Delta\mu^2\sqrt{\kappa}}{R_Q(\|g\|_2 + \mu(\|u_1\|_2 + \|u_2\|_2))^2}\right\}\right)\text{-approximate solution}$$

to (20) for some $\lambda \ge 0$ (which may vary each time), $\tilde{O}(d)$ additional work, $\tilde{O}(1)$ additional depth, and no additional matvecs to H, where $\tilde{O}(\cdot)$ hides polylog factors in r_1, r_2, μ, L, Δ and the ℓ_2-norms of u_1, u_2, g.

Proof We first apply the cutting plane method of Proposition 14 on the dual optimization problem (12) with $h \leftarrow -Q$ (since Q is concave), $n \leftarrow 2$, $\beta \leftarrow \beta_Q$ given by (19) (which is a valid choice by Lemma 10), and $R \leftarrow R_Q$ (guaranteeing an optimal dual solution is within S by Lemma 8). With $\delta \leftarrow \tilde{\Theta}(\epsilon/R_Q)$ as in Proposition 14 (for $\epsilon > 0$ to be chosen momentarily), a query to $\mathcal{O}_{Q,\delta}(\lambda)$ for $\lambda \geq 0$ can be implemented by computing $x' \in \mathbb{R}^d$ which is an

$$\left(\frac{\epsilon}{R_q} \cdot \frac{\sqrt{2\kappa}}{4\Upsilon^2}\right)\text{-approximate solution to (20)} \tag{21}$$

and $O(d)$ additional work by Lemma 11, with Υ defined in (18). Thus, since $n = 2$, the cutting plane method of Proposition 14 obtains $\tilde{\lambda} \geq 0$ such that $\max_{\lambda' \geq 0} Q(\lambda') - Q(\tilde{\lambda}) \leq \epsilon$ having computed at most $\tilde{O}(1)$ points x' satisfying (21) with $\tilde{O}(d)$ additional work and no additional matvecs to H.

Then, we compute $z \in \mathbb{R}^d$ such that $\|z - x^\star(\tilde{\lambda})\|_2 \leq \Delta/4$, and, choosing $\epsilon = \Theta(\mu\Delta^2)$, obtain $\|x^\star(\tilde{\lambda}) - x^\star\|_2 \leq \Delta/4$ by Lemma 12. Since $x^\star \in C_1 \cap C_2$, we have $\max_{i \in \{1,2\}} \mathrm{dist}(x^\star(\tilde{\lambda}), C_i) \leq \Delta/4$, and thus applying Lemma 13 with $y \leftarrow x^\star(\tilde{\lambda})$ and $x \leftarrow z$ gives that $z' := \mathrm{proj}_{C_1}(z)$ and $\tilde{x} := \mathrm{proj}_{C_2}(z')$ satisfy $\|\tilde{x} - x^\star(\tilde{\lambda})\|_2 \leq 3\Delta/4$. Furthermore, $\tilde{x}$ is feasible and can be computed in $O(d)$ work given z and with no additional matvecs. Finally, $\|\tilde{x} - x^\star\|_2 \leq \Delta$ by triangle inequality. ■

A.2. Ball optimization oracles via linear system solves

Carmon et al. (2020a) showed how to implement a *ball optimization oracle* (Carmon et al., 2020a, Def. 1) via exact linear system solves for a Hessian-stable function. In this section, we extend their guarantees to approximate linear system solves. First, we recall the definition of a ball optimization oracle from Carmon et al. (2020a). We note that due to our application where we seek to obtain a minimizer over $\mathbb{B}^d$ (Carmon et al. (2020a) seek to obtain an unconstrained minimizer), it is necessary to add an additional constraint to $\mathbb{B}^d$, which (Carmon et al., 2020a, Def. 1) does not add. However, otherwise the definition is the same.

Definition 16 ((Carmon et al., 2020a, Def. 1)) *We call* $\mathcal{O}_{\text{ball}}$ *a* (ζ, r)-ball optimization oracle *for* $f : \mathbb{R}^d \to \mathbb{R}$ *if for any* $\bar{x} \in \mathbb{B}^d$*, it outputs* $y = \mathcal{O}_{\text{ball}}(\bar{x}) \in \mathbb{B}^d_r(\bar{x}) \cap \mathbb{B}^d$ *such that* $\|y - x_{\bar{x},r}\|_2 \leq \zeta$ *for some* $x_{\bar{x},r} \in \mathrm{argmin}_{x \in \mathbb{B}^d_r(\bar{x}) \cap \mathbb{B}^d} f(x)$.

Next, we restate, for completeness, the definition of a Hessian-stable function. We state it for general norms as it appears in Carmon et al. (2020a), although in the following we will only instantiate it for the ℓ_2-norm.

Definition 17 ((Carmon et al., 2020a, Def. 7)) *A twice-differentiable function* $f : \mathbb{R}^d \to \mathbb{R}$ *is* (r, c)-Hessian stable *for* $r, c \geq 0$ *with respect to a norm* $\|\cdot\|$ *if for all* $x, y \in \mathbb{R}^d$ *with* $\|x - y\| \leq r$ *we have* $c^{-1}\nabla^2 f(y) \preceq \nabla^2 f(x) \preceq c\nabla^2 f(y)$.

We state our main guarantee for this section in the following lemma. This guarantee follows from combining the guarantee of Proposition 15 from the previous section with the accelerated ball-constrained Newton method due to Carmon et al. (2020a). In particular, this accelerated ball-constrained Newton method reduces implementing a ball optimization oracle to solving quadratics of the form (23) in our specific application, where there are two ball constraints. Carmon et al. (2020a) show how to implement a quadratic constrained to a single ball (22) via exact linear system solves by binary searching on the one-dimensional dual Lagrange multiplier. The necessity of the previous section comes from the fact that our application results in two ball constraints, and furthermore we must solve the linear systems approximately.

Lemma 18 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth, μ-strongly convex, and (r, c)-Hessian stable, all in $\|\cdot\|_2$, and let $\bar{x} \in \mathbb{B}^d$ be such that $\|\nabla f(\bar{x})\|_2 \le G$. Then there is an algorithm which implements a (ζ, r)-ball optimization oracle for query point $\bar{x}$. Additionally, letting $\tilde{O}(\cdot)$ hides polylog factors in L, μ, G, c, r, ζ, each of the $\tilde{O}(c)$ iterations of the algorithm requires*

- *computing $x' \in \mathbb{R}^d$ such that which is a $(\mathsf{poly}(L, \mu^{-1}, G, c, r, \zeta^{-1}))^{-1}$-approximate solution to $(\nabla^2 f(\bar{x}) + \lambda I)y = \hat{g}$, for some $\lambda \ge 0$ and $\hat{g} \in \mathbb{R}^d$ (which may vary each time),*
- *$\tilde{O}(1)$ additional matvecs to $\nabla^2 f(\bar{x})$,*
- *$\tilde{O}(1)$ additional evaluations of $\nabla f(\cdot)$,*
- *does not access f in any other way outside of the above,*
- *$\tilde{O}(d)$ additional work (outside of the above), and $\tilde{O}(1)$ additional depth.*

Proof Note that by strong convexity, we have $\|\bar{x} - \hat{x}\|_2 \le G/\mu$ for the unconstrained minimizer $\hat{x} := \operatorname{argmin}_{x \in \mathbb{R}^d} f(x)$. Then, the proof of correctness follows the same steps as the proof of (Carmon et al., 2020a, Theorem 9) with $D \leftarrow G/\mu$ and $M \leftarrow I$, except replacing the call to (Carmon et al., 2020a, Algorithm 7) in Line 9 of (Carmon et al., 2020a, Algorithm 3) with a call to the cutting plane method of Proposition 15. (Carmon et al., 2020a, Algorithm 7) solves the trust-region problem

$$\underset{x \in \mathbb{B}_r(\bar{x})}{\text{minimize}} \; -g^\top x + \frac{1}{2} x^\top H x \tag{22}$$

for $H \in \mathbb{S}_{++}^{d \times d}$ with $\mu I \preceq H \preceq LI$ (in particular, $H = \nabla^2 f(\bar{x})$); specifically by returning $\tilde{x} \in \mathbb{B}_r(\bar{x})$ such that $\|\tilde{x} - x_{g,H}\|_2 \le \Delta$, where $x_{g,H}$ is the minimizer of (22). Instead, we replace this subroutine with a call to the algorithm of Proposition 15 to obtain an approximate minimizer of

$$\underset{x \in \mathbb{B}_r(\bar{x}) \cap \mathbb{B}^d}{\text{minimize}} \; -g^\top x + \frac{1}{2} x^\top H x \,, \tag{23}$$

namely, $\tilde{x}'$ such that $\|\tilde{x}' - x'_{g,H}\|_2 \le \Delta$ for $x'_{g,H}$ the minimizer of (23). In particular, we instantiate $u_1 \leftarrow 0$, $r_1 \leftarrow 1$, $u_2 \leftarrow \bar{x}$, and $r_2 \leftarrow r$.

The proof of correctness of (Carmon et al., 2020a, Algorithm 3) treats (Carmon et al., 2020a, Algorithm 7) as a black-box; it only requires that the output satisfies $\|\tilde{x} - x_{g,H}\|_2 \le \Delta$ by (Carmon et al., 2020a, Prop. 8). It does not exploit the specific geometry of the domain. Thus, substituting (Carmon et al., 2020a, Prop. 8) with Proposition 15 in the proof of (Carmon et al., 2020a, Theorem 9) yields the desired correctness result, up to requiring a polynomial bound on $\|g\|_2$ whenever (23) is invoked, which we do next. (Note that when instantiating Proposition 15, we choose $\Delta \leftarrow \frac{\mu\zeta^2}{4Lc(5r+D)}$ as in Line 5 of (Carmon et al., 2020a, Algorithm 3).)

To conclude the proof of correctness, we must obtain a polynomial bound on $\|g\|_2$ each time Line 9 of (Carmon et al., 2020a, Algorithm 3) solves an instance of (23), due to the $\|g\|_2$-dependence in Proposition 15. To obtain this bound, note that it is clear from the pseudocode of (Carmon et al., 2020a, Algorithm 3) that g always takes the form $\nabla f(y) - H(\alpha y + (1-\alpha)z)$ for some $\alpha \in (0,1)$ and $y, z \in \mathbb{B}^d$. Thus, we can bound

$$\|\nabla f(y) - H(\alpha y + (1-\alpha)z)\|_2 \leq \|\nabla f(y)\|_2 + \|H(\alpha y + (1-\alpha)z)\|_2 \leq G + 3L$$

since $\|\nabla f(y)\|_2 = \|\nabla f(y) - \nabla f(\bar{x}) + \nabla f(\bar{x})\|_2 \leq L\|y - \bar{x}\|_2 + \|\nabla f(\bar{x})\|_2 \leq 2L + G$, and $\|H(\alpha y + (1-\alpha)z)\|_2 \leq L$ since $0 \preceq H \preceq LI$ in particular.

Finally, as for the complexity bounds, the number of iterations of (Carmon et al., 2020a, Algorithm 3) is $\tilde{O}(c)$ per the proof of (Carmon et al., 2020a, Thm. 9). Each iteration calls the algorithm of Proposition 15 (previously (Carmon et al., 2020a, Algorithm 7)) a single time and makes a single additional matvec to $\nabla^2 f(\bar{x})$ and query to $\nabla f(\cdot)$. Thus, the bound on the number of approximate linear system solves follows from Proposition 15. The additional work bound follows since each iteration of (Carmon et al., 2020a, Algorithm 3) uses $O(d)$ additional work. ■

A.3. ℓ_2-ℓ_1-games via linear system solves

In this section, we put everything together to reduce solving an ϵ-game to approximate linear system solves, culminating in the proof of Lemma 7. In particular, this is achieved by reducing solving an ϵ-game to implementing a series of ball optimization oracles, in which case we may apply Lemma 18 from the previous section. This reduction is performed using the ball oracle acceleration framework developed in Carmon et al. (2020a, 2021); Asi et al. (2021b); Carmon et al. (2022); Carmon and Hausler (2022); Carmon et al. (2023, 2024a). First, however, we set up notation and give some preliminary technical lemmas.

Throughout this section, we fix an ϵ-game per Definition 5, and define $f(x) := \max_{y\in\Delta^n} y^\top Ax$. Recall that our goal is to obtain an ϵ-minimizer of f over $\mathbb{B}^d$, and we assume $\|A\|_{2\to\infty} = \max_{i\in[n]}\|A_{i,:}\|_2 \leq 1$. For $\alpha > 0$, we define $\mathrm{smax}_\alpha : \mathbb{R}^n \to \mathbb{R}$ via

$$\mathrm{smax}_\alpha(y) := \max_{y'\in\Delta^n} y^\top y' - \alpha \cdot e(y') = \alpha \log\left(\sum_{i\in[n]} \exp(y_i/\alpha)\right)$$

where $e(y) := \sum_{i\in[n]} y_i \log y_i$ is the negative entropy function.

Then, we recall the following definition of a *quasi-self-concordant function*, for which we use the formulation of Carmon et al. (2020a), restated for completeness. See also Bach (2010); Sun and Tran-Dinh (2019); Karimireddy et al. (2018); Carmon et al. (2020a); Doikov (2023) for other work on optimizing such functions.

Definition 19 (Quasi-self-concordance (Carmon et al., 2020a, Def. 10)) *We say that a thrice-differentiable $h : \mathbb{R}^d \to \mathbb{R}$ is M-quasi-self-concordant (QSC) with respect to some norm $\|\cdot\|$, for $M \geq 0$, if for all $u, w, x \in \mathbb{R}^d$,*

$$|\nabla^3 h(x)[u,u,w]| \leq M\|w\|\|u\|^2_{\nabla^2 h(x)},$$

i.e., the restriction of the third-derivative tensor of h to any direction is bounded by a multiple of its Hessian norm.

We collect some properties of $\mathrm{smax}_\alpha(y)$ in the following lemma. For the latter two properties, we cite Carmon et al. (2020a).

Lemma 20 (Properties of $\mathrm{smax}_\alpha(y)$) *For $\alpha > 0$, letting $h(y) := \mathrm{smax}_\alpha(y)$, we have that h is 1-Lipschitz, $1/\alpha$-smooth, and $2/\alpha$-QSC, all in $\|\cdot\|_\infty$.*

Proof The latter two properties are given in (Carmon et al., 2020a, Lemma 14). As for Lipschitzness, note

$$\frac{\partial h}{\partial y_k} = \frac{\exp(y_k/\alpha)}{\sum_{i\in[n]} \exp(y_i/\alpha)}\,,$$

implying $\|\nabla h(y)\|_1 \le 1$ for all $y \in \mathbb{R}^n$. ■

Next, we collect standard properties of $\mathrm{smax}_\alpha(Ax)$.

Lemma 21 (Properties of $\mathrm{smax}_\alpha(Ax)$) *For $\alpha > 0$, letting $g(x) := \mathrm{smax}_\alpha(Ax)$, we have that $|g(x) - f(x)| \le \alpha \log n$ for all $x \in \mathbb{R}^d$. Furthermore, g is 1-Lipschitz, $1/\alpha$-smooth, and $2/\alpha$-QSC, all in $\|\cdot\|_2$. Finally, defining $p_x \in \Delta^n$ via $[p_x]_i \propto \exp([Ax]_i/\alpha)$ for $i \in [n]$, we have*

$$\nabla g(x) = A^\top p_x \;\; \textit{and} \;\; \nabla^2 g(x) = \frac{1}{\alpha} A^\top(\mathrm{diag}(p_x) - p_x p_x^\top)A\,.$$

Proof The first claim follows because the negative entropy function $e(y)$ takes values in $[-\log n, 0]$. Then, defining $h(y) := \mathrm{smax}_\alpha(y)$, note $\nabla g(x) = A^\top \nabla h(Ax)$. Lipschitzness follows since $\|\nabla g(x)\|_2 \le 1$ for all $x \in \mathbb{R}^d$ by triangle inequality along with the fact that $\max_{i\in[n]}\|A_{i,:}\|_2 \le 1$ by assumption and $\|\nabla h(Ax)\|_1 \le 1$ by Lemma 20.

As for smoothness, recall in general that a twice-differentiable convex function $\phi : \mathbb{R}^d \to \mathbb{R}$ is β-smooth with respect to a norm $\|\cdot\|$ if and only if $v^\top \nabla^2 \phi(x) v \le \beta \|v\|^2$ for all $x, v \in \mathbb{R}^d$. Then note that for any $v \in \mathbb{R}^d$, we have

$$v^\top \nabla^2 g(x) v = v^\top A^\top \nabla^2 h(Ax) A v \overset{(i)}{\le} \frac{1}{\alpha}\|Av\|_\infty^2 \overset{(ii)}{\le} \frac{1}{\alpha}\|v\|_2^2$$

by (i) Lemma 20 and (ii) the fact that $\sup_{u\neq 0} \frac{\|Au\|_\infty}{\|u\|_2} \le 1$ by assumption on A, implying $\|Av\|_\infty \le \|v\|_2$ in particular.

As for quasi-self concordance, note that for all $u, w, x \in \mathbb{R}^d$,

$$|\nabla^3 g(x)[u,u,w]| = |\nabla^3 h(Ax)[Au, Au, Aw]| \overset{(iii)}{\le} \frac{2}{\alpha}\|Aw\|_\infty \|Au\|_{\nabla^2 h(Ax)}^2 \overset{(iv)}{\le} \frac{2}{\alpha}\|w\|_2 \|u\|_{\nabla^2 g(x)}^2$$

by (iii) Lemma 20 and (iv) $\|Aw\|_\infty \le \|w\|_2$ as well as the fact that

$$\|Au\|_{\nabla^2 h(Ax)}^2 = u^\top A^\top \nabla^2 h(Ax) A u = u^\top \nabla^2 g(x) u = \|u\|_{\nabla^2 g(x)}^2\,.$$

The gradient and Hessian follow via straightforward computation. ■

Recall in the previous section we introduced the formulation of a ball optimization oracle from Carmon et al. (2020a) in Definition 16. We used this formulation there as it was directly adapted to their accelerated ball-constrained Newton method, which we used to prove Lemma 18 . However, the ball oracle acceleration framework guarantee which we cite in Proposition 23 below uses a slightly different formulation of a ball optimization oracle, which involves minimizing a regularized function over a ball of radius r. In the following lemma, we show that implementing the former suffices to implement the latter.

Lemma 22 *Let $h : \mathbb{R}^d \to \mathbb{R}$ be convex, L-smooth, and β-Lipschitz in $\|\cdot\|_2$. Then for $\lambda > 0$ and $g_{\lambda,\bar{x}}(x) \coloneqq h(x) + \frac{\lambda}{2}\|x - \bar{x}\|_2^2$, the output $x' = \mathcal{O}_{\text{ball}}(\bar{x})$ of a (ζ, r)-ball optimization oracle $\mathcal{O}_{\text{ball}}(\cdot)$ for $g_{\lambda,\bar{x}}$ (Definition 16) satisfies*

$$g_{\lambda,\bar{x}}(x') - \min_{x \in \mathbb{B}_r(\bar{x}) \cap \mathbb{B}^d} g_{\lambda,\bar{x}}(x) \le (\beta + \lambda r)\zeta + (L + \lambda)\zeta^2/2\,.$$

Proof Note that $g_{\lambda,\bar{x}}$ is $(L + \lambda)$-smooth, in which case, letting $x^\star \coloneqq \operatorname{argmin}_{x \in \mathbb{B}_r(\bar{x}) \cap \mathbb{B}^d} g_{\lambda,\bar{x}}(x)$,

$$\begin{aligned} g_{\lambda,\bar{x}}(x') - g_{\lambda,\bar{x}}(x^\star) &\le \langle \nabla g_{\lambda,\bar{x}}(x^\star), x' - x^\star \rangle + \frac{L + \lambda}{2}\|x' - x^\star\|_2^2 \\ &\le \|\nabla g_{\lambda,\bar{x}}(x^\star)\|_2 \|x' - x^\star\|_2 + (L + \lambda)\zeta^2/2 \\ &\le \zeta \|\nabla h(x^\star) + \lambda(x^\star - \bar{x})\|_2 + (L + \lambda)\zeta^2/2 \\ &\le \zeta(\|\nabla h(x^\star)\|_2 + \lambda\|x^\star - \bar{x}\|_2) + (L + \lambda)\zeta^2/2 \\ &\le (\beta + \lambda r)\zeta + (L + \lambda)\zeta^2/2\,. \end{aligned}$$

■

Next we state a ball oracle acceleration guarantee due to (Carmon et al., 2023, Prop. 2), which is itself based on (Carmon and Hausler, 2022, Prop. 1) with minor modifications. In particular, their framework reducing minimizing a convex and Lipschitz function to a sequence of regularized ball-constrained subproblems of the form (24). Importantly, each regularized subproblem is guaranteed a certain minimal level of regularization, which is critical to obtaining our ultimate guarantees.

Proposition 23 ((Carmon et al., 2023, Prop. 2) and (Carmon and Hausler, 2022, Prop. 1)) *Let $\epsilon > 0$, $r \in (0, 1)$, and $h : \mathbb{R}^d \to \mathbb{R}$ be convex and β-Lipschitz in $\|\cdot\|_2$. Then there is an algorithm that, with probability at least $3/4$ over the draw of a random seed $\chi \sim \mathcal{P}_{\text{seed}}$, returns $\tilde{x} \in \mathbb{B}^d$ such that $h(\tilde{x}) - \min_{x \in \mathbb{B}^d} h(x) \le \epsilon$. Additionally, letting $\tilde{O}(\cdot)$ hide polylog factors in $\beta, \epsilon^{-1}, r^{-1}$, each of the $\tilde{O}(r^{-2/3})$ iterations of the algorithm requires*

- *computing $x' \in \mathbb{B}_r^d(\bar{x}) \cap \mathbb{B}^d$ such that, for $g_{\lambda,\bar{x}}(x) \coloneqq h(x) + \frac{\lambda}{2}\|x - \bar{x}\|_2^2$,*

$$g_{\lambda,\bar{x}}(x') - \min_{x \in \mathbb{B}_r^d(\bar{x}) \cap \mathbb{B}^d} g_{\lambda,\bar{x}}(x) \le O(\lambda r^{8/3}) \tag{24}$$

  for some $\bar{x} \in \mathbb{B}^d$ and $\lambda = \tilde{\Omega}(\epsilon r^{-4/3})$ (which may vary each time), and does not access h in any other way,

- *$\tilde{O}(d)$ additional work, and $\tilde{O}(1)$ additional depth.*

Moreover, $\mathcal{P}_{\text{seed}}$ is independent of h, and $\chi \sim \mathcal{P}_{\text{seed}}$ can be sampled in $\tilde{O}(r^{-2/3})$-work and $\tilde{O}(1)$-depth.

Proof This follows from an instantiation of (Carmon et al., 2023, Prop. 2), which is itself based on (Carmon and Hausler, 2022, Prop. 1). The setup of (Carmon et al., 2023, Prop. 2) is not constrained to the unit ball, but (Carmon and Hausler, 2022, Prop. 1) allows for an arbitrary convex constraint set $\mathcal{X}$, which we set to $\mathbb{B}^d$, so that the constraints of (24) match (Carmon and Hausler, 2022, Def. 1). The total work bound is clear from the discussion in (Carmon et al., 2023, Appendix C) as well as Algorithms 1 and 2 in Carmon and Hausler (2022). In particular, there are $\tilde{O}(r^{-2/3})$ iterations, each with $\tilde{O}(d)$ additional work. ∎

In the following lemma, we combine Proposition 23 and Lemma 18 to reduce optimizing a general convex, smooth, Lipschitz, and Hessian-stable function to approximate linear system solves in the regularized Hessian of the matrix. We note that to enable more direct application of the sample reuse framework of Jin et al. (2026) in later sections, we "redefine" a single iteration of the algorithm of Lemma 24 to correspond to a single linear system solve.

Lemma 24 *For $\epsilon > 0$ and $r \in (0, 1)$, let $h : \mathbb{R}^d \to \mathbb{R}$ be convex, L-smooth, β-Lipschitz, and (r, c)-Hessian stable (Definition 17), all in $\|\cdot\|_2$. Then there is an algorithm that, with probability at least 3/4 over the draw of a random seed $\chi \sim \mathcal{P}_{\text{seed}}$, returns $\tilde{x} \in \mathbb{B}^d$ such that $h(\tilde{x}) - \min_{x \in \mathbb{B}^d} h(x) \leq \epsilon$. Additionally, letting $\tilde{O}(\cdot)$ hide polylog factors in $L, \beta, \epsilon^{-1}, r^{-1}, c$, each of the $\tilde{O}(cr^{-2/3})$-iterations of the algorithm requires*

- *computing $x' \in \mathbb{R}^d$ which is a $(\mathsf{poly}(L, \beta, \epsilon^{-1}, r^{-1}, c))^{-1}$-approximate solution to $(\nabla^2 h(\bar{x}) + \lambda I)y = \hat{g}$ for some $\bar{x} \in \mathbb{B}^d$, $\lambda = \tilde{\Omega}(\epsilon r^{-4/3})$, and $\hat{g} \in \mathbb{R}^d$ (which may vary each time),*
- *$\tilde{O}(1)$ additional matvecs to $\nabla^2 h(\cdot)$,*
- *$\tilde{O}(1)$ additional evaluations of $\nabla h(\cdot)$,*
- *does not access h in any other way outside of the above,*
- *$\tilde{O}(d)$ additional work (outside of the above), and $\tilde{O}(1)$ additional depth.*

Moreover, $\mathcal{P}_{\text{seed}}$ is independent of h, and $\chi \sim \mathcal{P}_{\text{seed}}$ can be sampled in $\tilde{O}(r^{-2/3})$-work and $\tilde{O}(1)$-depth.

Proof This follows from instantiating Proposition 23 where each implementation of (24) is performed via the algorithm of Lemma 18. Indeed, Lemma 22 shows that (24) can be implemented via a single call to a (ζ, r)-ball optimization oracle for $g_{\lambda,\bar{x}}$ for an appropriate choice of precision ζ. Furthermore, note

$$\nabla g_{\lambda,\bar{x}}(x) = \nabla h(x) + \lambda(x - \bar{x}) \text{ and } \nabla^2 g_{\lambda,\bar{x}}(x) = \nabla^2 h(x) + \lambda I \,,$$

and therefore a gradient evaluation of $\nabla g_{\lambda,\bar{x}}(\cdot)$ requires a gradient evaluation of $\nabla h(\cdot)$ and $O(d)$ additional work. Also, a matvec to $\nabla^2 g_{\lambda,\bar{x}}(\cdot)$ requires a matvec to $\nabla^2 h(\cdot)$ and at most $O(d)$ additional work. Finally, it is straightforward to show that $g_{\lambda,\bar{x}}$ is also (r, c)-Hessian stable. ∎

Finally, we prove the main result of Appendix A below in Lemma 7, which reduces solving the ϵ-game to approximate linear system solves. The proof is a straightforward application of Lemma 24, using the properties of the softmax function derived above.

Lemma 7 *For $\epsilon > 0$, there is an algorithm that, with probability at least $3/4$ over the draw of a random seed $\chi \sim \mathcal{P}_{\text{seed}}$, returns $\tilde{x} \in \mathbb{B}^d$ such that $f(\tilde{x}) - \min_{x \in \mathbb{B}^d} f(x) \leq \epsilon$. The computational cost of the method is dominated by the cost of $\tilde{O}(\epsilon^{-2/3})$-iterations of performing the following:*

- *computing $x' \in \mathbb{R}^d$ which is a $(\mathsf{poly}(\epsilon^{-1}, n))^{-1}$-approximate solution to $(H + \nu I)^{-1} y = \hat{g}$, where*

$$H = \frac{1}{\epsilon'} A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'} p_{\bar{x},\epsilon'}^\top) A \tag{4}$$

  for some $\bar{x} \in \mathbb{B}^d$, $\nu = \tilde{\Omega}(\epsilon^{-1/3})$, and $\hat{g} \in \mathbb{R}^d$ (which may vary each time) with $\epsilon' := \frac{\epsilon}{2 \log n}$,

- *$\tilde{O}(1)$-additional matvecs to A,*

- *$\tilde{O}(n + d)$-additional work, and $\tilde{O}(1)$-additional depth.*

Moreover, $\mathcal{P}_{\text{seed}}$ is independent of A, and $\chi \sim \mathcal{P}_{\text{seed}}$ can be sampled in $\tilde{O}(\epsilon^{-2/3})$-work and $\tilde{O}(1)$-depth.

Proof We apply Lemma 24 with $h(x) := \text{smax}_\alpha(Ax)$ for $\alpha := \frac{\epsilon}{2 \log n}$ and $r \leftarrow \Theta(\epsilon / \log n)$. By Lemma 21, we have $|f(x) - h(x)| \leq \epsilon/2$ for all $x \in \mathbb{R}^d$, and therefore it suffices to obtain a $\frac{\epsilon}{2}$-minimizer of h. By Lemma 21, we have that h is 1-Lipschitz, $\frac{2 \log n}{\epsilon}$-smooth, and $\frac{4 \log n}{\epsilon}$-QSC in $\|\cdot\|_2$. The latter and (Carmon et al., 2020a, Lemma 11) (with $M \leftarrow I$) imply h is $(r, O(1))$-Hessian stable in $\|\cdot\|_2$. Finally, note that by the gradient and Hessian expressions in Lemma 21, a matvec to $\nabla^2 h(\cdot)$ as well as an evaluation of $\nabla h(\cdot)$ can both be computed in $O(1)$ matvecs to A and $O(n)$ additional work.

■

Appendix B. Linear system solving

In this section, we show how to implement an efficient (ϵ, δ)-linear system solver (Definition 4) for

$$\frac{1}{\epsilon'} A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'} p_{\bar{x},\epsilon'}^\top) A + \nu I_d \tag{25}$$

for some $\bar{x} \in \mathbb{B}^d, \epsilon, \epsilon' > 0$, $\nu > \lambda > 0$, $\delta \in (0, 1)$, and $A \in \mathbb{R}^{n \times d}$ with $\|A\|_{2 \to \infty} = 1$. Recall from (3) that for any $x \in \mathbb{B}^d$ and $\eta > 0$, we use the notation $p_{x,\eta}$ to denote the vector in Δ^n defined via $[p_{x,\eta}]_i \propto \exp([Ax]_i / \eta)$ for $i \in [n]$ and correspondingly denote $P_{x,\eta} := \text{diag}(p_{x,\eta})$.

First, in Section B.1 we discuss several linear algebraic preliminaries which will aid in our analysis. Next in Section B.2 and B.3, we discuss the sequence of reductions discussed in Section 3, which enable our method. Finally, in Section B.4 we combine the results in the preceding section to give our final linear system solver guarantees. Throughout this section, we use $\tilde{O}(\cdot)$ to hide polylogarithmic factors in the following parameters: $n, d, \epsilon^{-1}, \delta^{-1}, \lambda^{-1}, \nu^{-1}$.

Randomized algorithms and random seeds. In the remainder of the paper, on occasion, it will be helpful to expose the random seed(s) used to seed the randomness in an (ϵ, δ)-linear system solver $\mathcal{O}$ for $M \in \mathbb{S}^d_{++}$ (Definition 4). In this case, we may write $\mathcal{O}_\xi$ to make explicit that the solver's randomness is fixed by the seed ξ. That is, $\mathcal{O}_\xi : \mathbb{R}^d \to \mathbb{R}^d$ is a *deterministic* mapping corresponding to *conditioning* on a fixed seed ξ. Moreover, on any input $b \in \mathbb{R}^d$ one can view $\mathcal{O}(b) \in \mathbb{R}^d$ as a random variable whose value is given by $\mathcal{O}_\xi(b)$ for a random $\xi \sim \mathcal{P}_{\text{seed}}$ (where $\mathcal{P}_{\text{seed}}$ is the distribution of the random seed used by $\mathcal{O}$). Note that because $\mathcal{O}$ is an (ϵ, δ)-linear system solver, for any $b \in \mathbb{R}^d$, we have that with probability $1 - \delta$ over the draw of $\xi \sim \mathcal{P}_{\text{seed}}$, $\mathcal{O}_\xi(b)$ is an ϵ-solution of $Mx = b$.

B.1. Linear algebraic preliminaries

Here, we review several useful properties from numerical linear algebra. As our techniques for linear system solving are in part motivated by those discussed in (Derezinski and Sidford, 2026; Derezinski et al., 2025), our presentation often follows their notation and definitions. In the paragraphs below, we discuss several matrix identities, formally introduce subspace embeddings, discuss preconditioning for linear system solving, and introduce several standard linear system solvers.

Matrix identities. A fact we use repeatedly throughout this paper is the Woodbury matrix identity.

Fact 25 (Woodbury matrix identity) *Let $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times k}, C \in \mathbb{R}^{k \times k}, D \in \mathbb{R}^{k \times n}$ where C, A, and $C^{-1} + DA^{-1}B$ are invertible. Then,*

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}.$$

The following special case of Fact 25, where $k = 1$, is commonly known as the Sherman-Morrison matrix identity.

Fact 26 (Sherman-Morrison matrix identity) *Let $A \in \mathbb{R}^{n \times n}$ be invertible and $u, v \in \mathbb{R}^n$. Then, $A + uv^\top$ is invertible if and only if $1 + v^\top A^{-1}u \neq 0$ and moreover,*

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}.$$

Subspace embeddings. Here, we formally introduce the notion of a regularized embedding. Intuitively, an (ϵ, ν)-embedding matrix constructs a low-dimensional $(1 + \epsilon)$ spectral approximation of $A^\top A + \nu I$, in the following sense.

Definition 27 (Regularized-embedding, Definition 8 of (Derezinski et al., 2025)) *For $\epsilon, \nu > 0$, we call $S \in \mathbb{R}^{k \times n}$ an (ϵ, ν)-(regularized subspace) embedding for $A \in \mathbb{R}^{n \times d}$ if*

$$A^\top S^\top SA + \nu I \approx_{1+\epsilon} A^\top A + \nu I.$$

The property of being an (ϵ, ν)-embedding is monotonic in ν, as formalized in the following fact.

Fact 28 *Suppose that $\epsilon, \nu > 0$ and $S \in \mathbb{R}^{k \times n}$ is a (ϵ, ν)-embedding for $A \in \mathbb{R}^{n \times d}$. Then for all $\nu' \geq \nu$, S is a (ϵ, ν')-embedding for A.*

Proof This follows from $(A^\top S^\top SA + \nu I) \approx_{1+\epsilon} (A^\top A + \nu I)$ and that $(\nu' - \nu)I \approx_{1+\epsilon} (\nu' - \nu)I$. ■

In order to construct embeddings, we leverage the following notio oblivious subspace embedding (OSE) moments.

Definition 29 (Oblivious subspace embedding (OSE) moments, Definition 9 of (Derezinski and Sidford, 2026)) *A random matrix $S \in \mathbb{R}^{s\times n}$ has $(\epsilon, \delta, d, \ell)$-OSE moments if for all matrices $U \in R^{n\times d}$ with orthonormal columns, $\mathbb{E}[\|U^\top S^\top SU - I_d\|_2^\ell] < \epsilon^\ell \delta$.*

Definition 30 (Sparse embedding matrix, Definition 10 of (Derezinski and Sidford, 2026)) *We say that a random matrix $S \in \mathbb{R}^{s\times n}$ is a sparse embedding matrix with sketch size s and b non-zeros per column, if it has independent columns, and each column consists of b random $\pm 1/\sqrt{b}$ entries placed uniformly at random without replacement.*

Next, we state several helpful properties of sparse embedding matrices.

Lemma 31 (Sparse embedding matrices preserve sparsity) *Let $A \in \mathbb{R}^{n\times d}$ and $S \in \mathbb{R}^{s\times n}$ be a sparse embedding matrix with sketch size s and at most b non-zeros per column (Definition 30). Then,* $\mathrm{nnz}(SA) \leq b\,\mathrm{nnz}(A)$.

Proof Let $B = SA \in \mathbb{R}^{s\times d}$. For any $r \in [s]$ and $j \in [d]$, the (r, j)-th entry of B is

$$B_{rj} = \sum_{i=1}^{n} S_{ri} A_{ij}.$$

If $B_{rj} \neq 0$, then there exists an index $i \in [n]$ such that $S_{ri} \neq 0$ and $A_{ij} \neq 0$. Fix a nonzero entry $A_{ij} \neq 0$. Its contribution to the product SA is confined to entries B_{rj} with $S_{ri} \neq 0$. Since column i of S has at most b nonzero entries, the nonzero A_{ij} can contribute to at most b entries of B. Summing over all nonzero entries of A, each of which contributes to at most b entries of SA, we obtain the result. ■

Several prior works (Nelson and Nguyên, 2013; Chenakkod et al., 2024a,b; Cohen, 2016) study guarantees for sparse embedding matrices. We use the version stated in (Derezinski and Sidford, 2026), which was originally due to (Chenakkod et al., 2024a).

Lemma 32 (Lemma 11 of Derezinski and Sidford (2026)) *For $\epsilon, \delta \in (0, 1/2)$, a sparse embedding matrix S with sketch size $s = O((d+\log(1/(\delta\epsilon)))/\epsilon^2)$ and $b = O\left(\log^2(d/(\delta\epsilon)/\epsilon) + \log^3(d/(\delta\epsilon))\right)$ non-zeros per column (Definition 30) has $(\epsilon, \delta, d, 16\log(d/(\delta\epsilon))$-OSE moments.*

In our analysis, we will leverage the following special case of Lemma 12 of (Derezinski and Sidford, 2026), which bounds the sketch size needed to construct a constant approximation for $A \in \mathbb{R}^{n\times d}$.

Corollary 33 *Let $A \in \mathbb{R}^{n\times d}$, $\lambda > 0$, $k := \lceil \|A\|_F^2/\lambda \rceil \leq d$, and $S \in \mathbb{R}^{s\times n}$ has $(1, \delta, 2k, \ell)$-OSE moments for some $\ell \geq 2$. With probability $1 - \delta$, for every $\nu \geq \lambda$, S is an $(2, \nu)$-embedding for A.*

Proof Take $\epsilon = 1/6$. We have that by Lemma 12 of Derezinski and Sidford (2026),

$$A^\top S^\top SA + \frac{1}{k}\sum_{i>k} \sigma_i^2(A) I_d \approx_2 A^\top A + \frac{1}{k}\sum_{i>k} \sigma_i^2(A) I_d.$$

Since $\frac{1}{k}\sum_{i>k} \sigma_i^2(A) \leq \|A\|_F^2/k \leq \lambda \leq \nu$, from Fact 28, we have $A^\top S^\top SA + \nu I_d \approx_2 A^\top A + \nu I_d$. ∎

In light of the previous corollary, in the following fact, we collect several useful but elementary properties of constant approximations of PD matrices, which will aid in the later analysis.

Preconditioning. The next theorem gives a standard guarantee on preconditioned Richardson iteration for solving a linear system in $M \in \mathbb{S}^d_{++}$ given a linear system solver (Definition 4) for $N \in \mathbb{S}^d_{++}$ such that $M \approx_c N$. Such an N is commonly called a *preconditioner* for M. We note that the following guarantee is well-known (see, e.g. (Golub and Overton, 1988), Lemma 2.5 of (Cohen et al., 2018) or the discussion in (Derezinski and Sidford, 2026)).

Theorem 34 (Preconditioned Richardson) *There is an algorithm*

$$\textsc{PrecondRichardson}(M, c, \epsilon, \delta, b, \mathcal{O})$$

which takes in $M \in \mathbb{S}^d_{++}$, $c \geq 1$, $\epsilon > 0$, $\delta \in (0,1)$, $b \in \mathbb{R}^d$, and an oracle $\mathcal{O}$. Here, $\mathcal{O}$ is a $(1/(4c^2), \delta/L)$-linear system solver for a matrix $N \in \mathbb{S}^d_{++}$ such that $N \approx_c M$. Letting $L = \lceil 2c^2 \log(2c/\epsilon)\rceil$, the algorithm makes L queries to $\mathcal{O}$, L matvecs to M, performs $O(dL)$-additional work using $\tilde{O}(1)$-additional depth, and with probability $1-\delta$, outputs an ϵ-solution of $Mx = b$.

Proof Fix $b \in \mathbb{R}^d$, and let $x^\star = M^{-1}b$. We define the solver via the preconditioned Richardson iteration with step size $\eta = 1/c$: initialize $x_0 = 0$, and for $t = 0, 1, \ldots, L-1$ perform

$$r_t = b - Mx_t, \qquad y_t = \mathcal{O}(r_t), \qquad x_{t+1} = x_t + \eta y_t,$$

and output x_L. Note that $M \approx_c N$ implies

$$\frac{1}{c} I \preceq N^{-1/2} M N^{-1/2} \preceq cI.$$

In particular, all eigenvalues of $N^{-1}M$ lie in $[1/c,\, c]$. Consider the update matrix $(I - \eta N^{-1}M)$. Since $\eta = 1/c$, the eigenvalues of this matrix lie in the range $[1-\eta c,\, 1-\eta(1/c)] = [0,\, 1-1/c^2]$. Therefore, for any $v \in \mathbb{R}^d$, the operator norm contracts:

$$\|(I - \eta N^{-1}M)v\|_N \leq \left(1 - \frac{1}{c^2}\right)\|v\|_N. \tag{26}$$

Also note that $\|N^{-1}Mv\|_N \leq c\|v\|_N$.

Let $e_t := x_t - x^\star$ denote the error at iteration t. We define the oracle error ξ_t such that $y_t = N^{-1}r_t + \xi_t$. The error update becomes:

$$e_{t+1} = x_t + \eta(N^{-1}r_t + \xi_t) - x^\star$$

$$= e_t + \eta N^{-1} M(x^\star - x_t) + \eta \xi_t$$
$$= (I - \eta N^{-1} M) e_t + \eta \xi_t. \quad (27)$$

The oracle guarantees that $\|y_t - N^{-1} r_t\|_N \le \frac{1}{4c^2} \|N^{-1} r_t\|_N$. By a union bound over L steps, with probability $1 - \delta$, for all t:

$$\|\xi_t\|_N \le \frac{1}{4c^2} \|N^{-1} M e_t\|_N \le \frac{1}{4c^2} \cdot c \|e_t\|_N = \frac{1}{4c} \|e_t\|_N.$$

Plugging this into (27) and using the triangle inequality:

$$\|e_{t+1}\|_N \le \left(1 - \frac{1}{c^2}\right) \|e_t\|_N + \eta \|\xi_t\|_N \le \left(1 - \frac{1}{c^2}\right) \|e_t\|_N + \frac{1}{c} \left(\frac{1}{4c} \|e_t\|_N\right).$$

Simplifying the terms, we obtain

$$\|e_{t+1}\|_N \le \left(1 - \frac{1}{c^2} + \frac{1}{4c^2}\right) \|e_t\|_N = \left(1 - \frac{3}{4c^2}\right) \|e_t\|_N \le \left(1 - \frac{1}{2c^2}\right) \|e_t\|_N.$$

Consequently, by induction,

$$\|e_L\|_N \le \left(1 - \frac{1}{2c^2}\right)^L \|e_0\|_N \le \exp\left(-\frac{L}{2c^2}\right) \|x^\star\|_N.$$

Taking $L = \lceil 2c^2 \log(2c/\epsilon) \rceil$, this implies $\|x_L - x^\star\|_N \le \frac{\epsilon}{c} \|x^\star\|_N$. Using $M \approx_c cN$, we conclude $\|x_L - x^\star\|_M \le \epsilon \|x^\star\|_M$ by Fact 49. ■

Standard solvers. In our algorithms underlying Theorems 2 and 3, we use preconditioned Richardson iteration (Theorem 34) and consider two underlying subroutines for implementing a linear system solver for the preconditioner N. The first, is to simply apply FMM, which eventually corresponds to our final runtime claimed in Theorem 2.

Lemma 35 (FMM (Alman et al., 2025)) *Consider $C \in \mathbb{R}^{d \times d}$, ν and $\epsilon > 0$. There is a (deterministic) algorithm $\mathrm{FMM}(C, b, \nu, \epsilon)$, which takes in an explicit $C \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d, \nu, \epsilon > 0$, does $\tilde{O}(d^\omega)$-work in $\tilde{O}(1)$-depth and outputs an ϵ-solution of $(C^\top C + \nu I_d)x = b$ (Definition 4).*

The second, is to use stochastic variance-reduced gradient descent (SVRG), which will eventually correspond to our final runtime claimed in Theorem 3.

Lemma 36 (SVRG Frostig et al. (2015); Lin et al. (2015)) *There is a randomized algorithm $\mathrm{SVRG}_\xi(C, b, \nu, \epsilon, \delta)$ that takes in an explicit $C \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d, \nu, \epsilon > 0, \delta \in (0, 1)$ and a random seed $\xi \sim \mathcal{P}_{\mathrm{seed}}$ such that, with probability $1 - \delta$ over the draw of $\xi \sim \mathcal{P}_{\mathrm{seed}}$, the algorithm outputs an ϵ-solution of $(C^\top C + \nu I_d)x = b$ (Definition 4). Moreover, the algorithm can be implemented in $\tilde{O}(\mathrm{nnz}(C) + d\|C\|_F^2/\nu)$-work using $\tilde{O}(\|C\|_F^2/\nu)$-depth, and $\mathcal{P}_{\mathrm{seed}}$ is independent of C and b.*

Proof Consider

$$G = \begin{pmatrix} C \\ \sqrt{\nu} I_d \end{pmatrix} \in \mathbb{R}^{2d \times d}, \qquad b' = \begin{pmatrix} 0_d \\ \frac{1}{\sqrt{\nu}} b \end{pmatrix} \in \mathbb{R}^{2d \times 1}$$

and $g(x) := \frac{1}{2}\|Gx - b'\|_2^2$. This is a least squares problem, whose normal equations are

$$G^\top Gx = (C^\top C + \nu I_d)x = G^\top b' = b.$$

Consequently, the (unique) minimizer $x^\star \in \mathbb{R}^d$ of $g(x)$ satisfies $(C^\top C + \nu I_d)x^\star = b$. Moreover,

$$g(x) = \sum_{i \in [2d]} g_i(x), \text{ where } g_i(x) = \frac{1}{2}(G_{i,:}^\top x - [b']_i)^2,$$

and each g_i is convex and has smoothness

$$\|G_{i,:}\|_2^2 = \begin{cases} \|C_{i,:}\|_2^2, & i \leq d \\ \nu, & i > d \end{cases}.$$

Consequently, standard SVRG guarantees (Frostig et al., 2015; Lin et al., 2015) imply that SVRG converges to an ϵ-solution of $(C^\top C + \nu I_d)x = b$ using $\tilde{O}(1)$ matvecs to G, $\tilde{O}(\|C\|_F^2/\nu)$ queries to a row of G, and $\tilde{O}(\|C\|_F^2/\nu)$-depth, which completes the proof. Since a matvec to G can be implemented by a matvec to C, and a row query to G requires $\tilde{O}(d)$-work, the result follows. ■

B.2. Simplifying the target matrix via a rank-one update

Here, we follow the proof approach in Section 3 to reduce the problem of building a linear system solver for (25) into the problem of building a linear system solver for $SBTT^\top B^\top S$ where $B = \frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2} A$ and $S \in \mathbb{R}^{s \times n}, T^\top \in \mathbb{R}^{s \times d}$ are sparse embedding matrices (Definition 30) of an appropriate sketch size s.

First, the following lemma states a general result that given a linear system solver (Definition 4) for $M \in \mathbb{S}_{++}^d$ and a rank one update $uu^\top$ such that $M - uu^\top \in \mathbb{S}_{++}^d$, we can implement a linear system solver for $M - uu^\top$ with low computational overhead and parallel depth.

As alluded in Section 3, this result reduces building a linear system solver for (25) to building a linear system solver for

$$\frac{1}{\epsilon'} A^\top P_{\bar{x}} A + \nu I. \tag{28}$$

This is advantageous because $B = \frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2} A$ is simply a diagonal rescaling of A, and consequently, it is more straightforward to apply the regularized embedding techniques discussed in Section B.1 to build an efficient linear system solver for (28) $= B^\top B + \nu I$.

Lemma 37 (Solving with a symmetric rank-one update) *There is an algorithm*

$$\text{RANKONESOLVE}(M, u, \lambda, \epsilon, \delta, b, \mathcal{O})$$

which takes in $M \succeq \lambda I_d$ and $u \in \mathbb{R}^d$ such that $M - uu^\top \succeq \lambda I_d$ for $\lambda > 0$, $\epsilon \in (0, 1/2)$, $\delta \in (0,1)$, $b \in \mathbb{R}^d$, and an oracle $\mathcal{O}$. Te algorithm makes two queries to $\mathcal{O}$, performs $O(d)$-additional work using $\tilde{O}(1)$-additional depth. If $\mathcal{O}$ is a $\left(\frac{\epsilon}{9\kappa_u^2}, \frac{\delta}{2}\right)$-approximate linear system solver for M with $\kappa_u := 1 + \frac{\|u\|_2^2}{\lambda}$, then with probability $1 - \delta$, the algorithm outputs an ϵ-solution to $(M - uu^\top)x = b$.

Proof Fix $b \in \mathbb{R}^d$. Let $\tilde{M} := M - uu^\top$. We wish to approximate $x^\star := \tilde{M}^{-1}b$. Define:

$$y := M^{-1}b, \qquad z := M^{-1}u, \qquad \text{and} \qquad \alpha := 1 - u^\top z.$$

First, we establish a lower bound on α. Since $M \succeq \tilde{M} \succeq \lambda I$, we have $M^{-1} \preceq \tilde{M}^{-1}$. By the Sherman-Morrison identity (Fact 26),

$$\tilde{M}^{-1} = M^{-1} + \frac{1}{\alpha} zz^\top.$$

Since $\tilde{M}^{-1}$ is PD, $\alpha \in (0, 1)$. Furthermore, $M \succeq \lambda I + uu^\top$ implies $M^{-1} \preceq (\lambda I + uu^\top)^{-1}$. Thus:

$$u^\top M^{-1} u \le u^\top (\lambda I + uu^\top)^{-1} u = \frac{\|u\|_2^2}{\lambda + \|u\|_2^2}.$$

Consequently,

$$\alpha = 1 - u^\top M^{-1} u \ge 1 - \frac{\|u\|_2^2}{\lambda + \|u\|_2^2} = \frac{\lambda}{\lambda + \|u\|_2^2} = \frac{1}{\kappa_u}. \tag{29}$$

The exact solution satisfies $x^\star = y + \frac{u^\top y}{\alpha} z$. The algorithm computes estimates using the oracle $\mathcal{O}$ with precision $\epsilon_{\text{op}} := \frac{\epsilon}{9\kappa_u^2}$:

$$\hat{y} \leftarrow \mathcal{O}(b), \qquad \hat{z} \leftarrow \mathcal{O}(u), \qquad \hat{\alpha} := 1 - u^\top \hat{z}, \qquad \hat{x} := \hat{y} + \frac{u^\top \hat{y}}{\hat{\alpha}} \hat{z}$$

and outputs $\hat{x}$. With probability $1 - 2(\delta/2) = 1 - \delta$, the oracle guarantees:

$$\|\hat{y} - y\|_M \le \epsilon_{\text{op}} \|y\|_M \qquad \text{and} \qquad \|\hat{z} - z\|_M \le \epsilon_{\text{op}} \|z\|_M. \tag{30}$$

Condition on this event. Note that $\|z\|_M = \sqrt{u^\top M^{-1} M M^{-1} u} = \sqrt{u^\top M^{-1} u} = \sqrt{1 - \alpha}$.

We first bound the error in the scalar $\hat{\alpha}$. By Cauchy-Schwarz and (30):

$$|\hat{\alpha} - \alpha| = |u^\top (\hat{z} - z)| \le \|u\|_{M^{-1}} \|\hat{z} - z\|_M \le \epsilon_{\text{op}} \|z\|_M^2 = \epsilon_{\text{op}}(1 - \alpha).$$

Recall that $\epsilon_{\text{op}} = \frac{\epsilon}{9\kappa_u^2}$. Since $\alpha \ge 1/\kappa_u$ by (29), we have $\epsilon_{\text{op}} \le \frac{\epsilon}{9}\alpha^2$. Substituting this into the error bound:

$$|\hat{\alpha} - \alpha| \le \frac{\epsilon}{9}\alpha^2(1 - \alpha) < \frac{\alpha}{2} \cdot \left(\frac{2\epsilon\alpha(1 - \alpha)}{9} \right).$$

Since $\epsilon < 1/2$ and $\alpha(1 - \alpha) \le 1/4$, the term in the parenthesis on the right hand side of the display above, is strictly less than 1. Thus $|\hat{\alpha} - \alpha| \le \alpha/2$, which implies:

$$\hat{\alpha} \ge \alpha - \frac{\alpha}{2} = \frac{\alpha}{2}, \qquad \text{hence} \qquad \frac{1}{\hat{\alpha}} \le \frac{2}{\alpha}. \tag{31}$$

Next, we decompose the error in $\hat{x}$:

$$\begin{aligned} \|\hat{x} - x^\star\|_M &\le \|\hat{y} - y\|_M + \left\| \frac{u^\top \hat{y}}{\hat{\alpha}} \hat{z} - \frac{u^\top y}{\alpha} z \right\|_M \\ &\le \|\hat{y} - y\|_M + \frac{|u^\top \hat{y}|}{\hat{\alpha}} \|\hat{z} - z\|_M + \left| \frac{u^\top \hat{y}}{\hat{\alpha}} - \frac{u^\top y}{\alpha} \right| \|z\|_M. \end{aligned} \tag{32}$$

We now bound the two scalar factors. First, by Cauchy–Schwarz, (31), and (30):

$$\frac{|u^\top \hat{y}|}{\hat{\alpha}} \leq \frac{\|u\|_{M^{-1}}\|\hat{y}\|_M}{\hat{\alpha}} \leq \frac{2\|u\|_{M^{-1}}(\|y\|_M + \|\hat{y}-y\|_M)}{\alpha} \leq \frac{4\|u\|_{M^{-1}}\|y\|_M}{\alpha}.$$

Combining this with $\|\hat{z}-z\|_M \leq \epsilon_{\text{op}}\|z\|_M$ and $\|z\|_M = \|u\|_{M^{-1}}$, we get the second term bound:

$$\frac{|u^\top \hat{y}|}{\hat{\alpha}}\|\hat{z}-z\|_M \leq \frac{4\epsilon_{\text{op}}}{\alpha}\|u\|_{M^{-1}}^2\|y\|_M = \frac{4\epsilon_{\text{op}}(1-\alpha)}{\alpha}\|y\|_M. \tag{33}$$

Next, for the third term, we split the difference:

$$\left|\frac{u^\top \hat{y}}{\hat{\alpha}} - \frac{u^\top y}{\alpha}\right| \leq \frac{|u^\top(\hat{y}-y)|}{\hat{\alpha}} + |u^\top y|\left|\frac{1}{\hat{\alpha}} - \frac{1}{\alpha}\right|.$$

For the first part, using $1/\hat{\alpha} \leq 2/\alpha$ and (30):

$$\frac{|u^\top(\hat{y}-y)|}{\hat{\alpha}} \leq \frac{2\epsilon_{\text{op}}\|u\|_{M^{-1}}\|y\|_M}{\alpha}. \tag{34}$$

For the second part, using $|\hat{\alpha}-\alpha| \leq \epsilon_{\text{op}}(1-\alpha)$ and the bound $1/\hat{\alpha} \leq 2/\alpha$ from (31),

$$\left|\frac{1}{\hat{\alpha}} - \frac{1}{\alpha}\right| = \frac{|\hat{\alpha}-\alpha|}{\alpha\hat{\alpha}} \leq \frac{2\epsilon_{\text{op}}(1-\alpha)}{\alpha^2}.$$

Multiplying both parts by $\|z\|_M = \|u\|_{M^{-1}} = \sqrt{1-\alpha} \leq 1$ yields:

$$\left|\frac{u^\top \hat{y}}{\hat{\alpha}} - \frac{u^\top y}{\alpha}\right|\|z\|_M \leq \left(\frac{2\epsilon_{\text{op}}}{\alpha} + \frac{2\epsilon_{\text{op}}}{\alpha^2}\right)\|y\|_M. \tag{35}$$

Substituting (30), (33), and (35) into (32), and simplifing using $\alpha \leq 1$:

$$\|\hat{x}-x^\star\|_M \leq \epsilon_{\text{op}}\|y\|_M + \frac{4\epsilon_{\text{op}}}{\alpha}\|y\|_M + \left(\frac{2\epsilon_{\text{op}}}{\alpha} + \frac{2\epsilon_{\text{op}}}{\alpha^2}\right)\|y\|_M \leq \frac{9\epsilon_{\text{op}}}{\alpha^2}\|y\|_M. \tag{36}$$

We now convert this to the $\tilde{M}$-norm. Since $\tilde{M} \preceq M$, we have $\|v\|_{\tilde{M}} \leq \|v\|_M$. Also, $\|x^\star\|_{\tilde{M}}^2 = b^\top \tilde{M}^{-1} b = b^\top M^{-1} b + \frac{1}{\alpha}(b^\top z)^2 \geq \|y\|_M^2$. Thus:

$$\|\hat{x}-x^\star\|_{\tilde{M}} \leq \|\hat{x}-x^\star\|_M \leq \frac{9\epsilon_{\text{op}}}{\alpha^2}\|x^\star\|_{\tilde{M}}.$$

Substituting $\epsilon_{\text{op}} = \frac{\epsilon\alpha^2}{9}$ (which is satisfied since $\alpha \geq 1/\kappa_u$) yields $\|\hat{x}-x^\star\|_{\tilde{M}} \leq \epsilon\|x^\star\|_{\tilde{M}}$. ■

B.3. Leveraging subspace embeddings and Woodbury identity

Next, as discussed in Section 3, using the Woodbury matrix identity (Fact 25), we first prove a general result that for any $B \in \mathbb{R}^{n\times d}$, in order to solve a linear system in $B^\top B + \nu I_d$ it suffices to solve a linear system in $BB^\top + \nu I_n$ plus moderate additional compute.

As discussed in Section 3, this is helpful because, as we will see in Lemma 39, this enable us to leverage two sparse embeddings $S \in \mathbb{R}^{s\times n}, T^\top \in \mathbb{R}^{s\times d}$ (Definition 30) to reduce building a linear system solver for the $d\times d$ matrix $B^\top B + \nu I_d$ to building a linear system solver for an $s \times s$ matrix $SBTT^\top BS + \nu I_s$ for an appropriate sketch dimension $s \ll d$. The following result is similar to Lemma 25 of (Derezinski and Sidford, 2026).

Lemma 38 (Solving in the Transpose) *There is an algorithm*

$$\textsc{TransposeSolve}(B, \nu, \epsilon, \delta, b, \mathcal{O})$$

which takes in $B \in \mathbb{R}^{n \times d}$, $\nu > 0$, $\epsilon > 0$, $\delta \in (0, 1)$, $b \in \mathbb{R}^n$, and an oracle $\mathcal{O}$. If $\mathcal{O}$ is an $\left(\frac{\epsilon\nu}{\|B\|_2^2}, \delta\right)$-linear system solver for $K := B^\top B + \nu I_d$, then the algorithm outputs an ϵ-approximate solution to $(BB^\top + \nu I_n)y = b$ with probability $1 - \delta$. The algorithm makes one query to $\mathcal{O}$, one matvec to B, and performs $O(n + d)$-additional work using $\tilde{O}(1)$-depth.

Proof Fix $b \in \mathbb{R}^n$ and let $M := BB^\top + \nu I_n$. We wish to approximate $y^\star := M^{-1}b$. By the Woodbury identity (Fact 25), $M^{-1} = \frac{1}{\nu}(I_n - B(B^\top B + \nu I_d)^{-1}B^\top)$. Letting $K := B^\top B + \nu I_d$, we define:

$$z^\star := K^{-1}B^\top b, \qquad y^\star = \frac{1}{\nu}(b - Bz^\star).$$

The algorithm computes $u \leftarrow B^\top b$, queries the oracle $\hat{z} \leftarrow \mathcal{O}(u)$, and outputs $\hat{y} := \frac{1}{\nu}(b - B\hat{z})$. Condition on the event that $\mathcal{O}$ succeeds (which occurs with probability at least $1 - \delta$), providing a solution $\hat{z}$ such that $\|\hat{z} - z^\star\|_K \le \epsilon_{\text{op}}\|z^\star\|_K$ for $\epsilon_{\text{op}} := \frac{\epsilon\nu}{\|B\|_2^2}$.

Let $e_z := \hat{z} - z^\star$. Then $\hat{y} - y^\star = -\frac{1}{\nu}Be_z$. To relate the error in the M-norm to the oracle error in the K-norm, observe that for any $v \in \mathbb{R}^d$:

$$\begin{aligned}
\|Bv\|_M^2 &= v^\top B^\top MBv \\
&= v^\top B^\top(BB^\top + \nu I_n)Bv \\
&= v^\top(B^\top B)(B^\top B + \nu I_d)v \\
&\le \|B^\top B\|_2 \cdot (v^\top Kv) = \|B\|_2^2\|v\|_K^2.
\end{aligned}$$

Applying this with $v = e_z$ gives:

$$\|\hat{y} - y^\star\|_M = \frac{1}{\nu}\|Be_z\|_M \le \frac{\|B\|_2}{\nu}\|e_z\|_K \le \frac{\|B\|_2}{\nu}\epsilon_{\text{op}}\|z^\star\|_K. \tag{37}$$

Next, we bound $\|z^\star\|_K$ in terms of $\|y^\star\|_M$. Note that $z^\star = B^\top y^\star$. Using the same logic as above:

$$\|z^\star\|_K^2 = (y^\star)^\top B(B^\top B + \nu I_d)B^\top y^\star = (y^\star)^\top(BB^\top)(BB^\top + \nu I_n)y^\star \le \|B\|_2^2\|y^\star\|_M^2.$$

Substituting $\|z^\star\|_K \le \|B\|_2\|y^\star\|_M$ into (37):

$$\|\hat{y} - y^\star\|_M \le \frac{\|B\|_2^2}{\nu}\epsilon_{\text{op}}\|y^\star\|_M.$$

By our choice of $\epsilon_{\text{op}} = \frac{\epsilon\nu}{\|B\|_2^2}$, we conclude $\|\hat{y} - y^\star\|_M \le \epsilon\|y^\star\|_M$. ■

By combining this result with subspace embedding techniques, we obtain the following guarantee.

Lemma 39 *There is a randomized algorithm*

$$\textsc{SketchAndSolve}_{\xi_1,\xi_2,\xi_3}(B, \nu, k, \epsilon, \delta, b', \mathcal{O})$$

which takes in $B \in \mathbb{R}^{n\times d}$, $\nu \geq \lambda > 0$, $k \geq \lceil \|B\|_F^2/\lambda \rceil$, $\epsilon, \delta \in (0,1), b' \in \mathbb{R}^d$, and an oracle $\mathcal{O}$. Here, letting, $s = \tilde{O}((k + \log(1/\delta)))$, $b = \tilde{O}(\log^2(k/\delta) + \log^3(k/\delta))$, and $L = \lceil \log(1/\epsilon) \rceil$, $\mathcal{O}$ is an

$$\left(\frac{\epsilon}{\|B\|_2^2}, \frac{\delta}{3L^2}\right)\text{-linear system solver for } B^\top B + \nu I_d.$$

The algorithm is parameterized by three independent random seeds $\xi_1 \sim \mathcal{P}_1, \xi_2 \sim \mathcal{P}_2, \xi_3 \sim \mathcal{P}_3$ such that ξ_1 seeds the randomness used to sample a sparse embedding matrix $S \in \mathbb{R}^{s\times n}$ with sketch size s and b nonzeros per column (i.e., conditioned on ξ_1, S is fixed); ξ_2, seeds the randomness used to sample a sparse embedding matrix $T^\top \in \mathbb{R}^{s\times d}$ with sketch size s and b nonzeros per column (i.e., conditioned on ξ_2, T is fixed); and ξ_3 seeds the randomness of $\mathcal{O}$ (i.e., for any $b'' \in \mathbb{R}^d$, $\mathcal{O}(b'')$ is a random variable equal to $\mathcal{O}_{\xi_3}(b'')$ for $\xi_3 \sim \mathcal{P}_3$),[4]

The algorithm makes $\tilde{O}(1)$ queries to $\mathcal{O}_{\xi_3}$, matvec queries to $SBB^\top S^\top + \nu I_s$, matvec queries to $B^\top S^\top$ and SB, and matvec queries to $SBTT^\top B^\top S^\top + \nu I_s$, and does $\tilde{O}(s+d)$-additional work using $\tilde{O}(1)$-additional depth. Moreover, $\mathcal{P}_1, \mathcal{P}_2$ do not depend on B, b and the random seeds ξ_1, ξ_2 can be sampled in $\tilde{O}(s(n+d))$-additional work using $\tilde{O}(1)$-additional depth.

Proof By Lemma 32 and Corollary 33, with probability $1 - \delta/3$ over the draw of $\xi_1 \sim \mathcal{P}_1$,

$$B^\top S^\top SB + \nu I_d \approx_2 B^\top B + \nu I_d \,. \tag{38}$$

Conditioning on this event, by Fact 49, $\|SB\|_F^2 \leq 2\|B\|_F^2$. Again, by Lemma 32 and Corollary 33, we have that with probability $1 - \delta/3$ over the draw of $\xi_2 \sim \mathcal{P}_2$,

$$SBTT^\top B^\top S^\top + \nu I_s \approx_2 SBB^\top S^\top + \nu I_s.$$

Condition on such ξ_1, ξ_2 in the remainder of the proof and let $\epsilon_{\text{op}} := \frac{\epsilon}{\|B\|_2^2}$ and $\delta_{\text{op}} := \frac{\delta}{3L^2}$. Now, conditioning on ξ_3, suppose that the algorithm outputs $\mathcal{O}^3(b)$, where

- $\mathcal{O}^1 : b' \in \mathbb{R}^s \mapsto \textsc{PrecondRichardson}(SBB^\top S^\top + \nu I_s, 2, \epsilon_{\text{op}}, \delta_{\text{op}}, b', \mathcal{O}_{\xi_3})$;
- $\mathcal{O}^2 : b' \in \mathbb{R}^d \mapsto \textsc{TransposeSolve}(B^\top S^\top SB, \nu, \epsilon_{\text{op}}, \delta/(3L), b', \mathcal{O}^1)$; and
- $\mathcal{O}^3 : b' \in \mathbb{R}^d \mapsto \textsc{PrecondRichardson}(B^\top B + \nu I_d, 2, \epsilon, \delta/3, b', \mathcal{O}^2)$

The following statements hold with probability $1 - \delta/3$ over the draw of ξ_3.

- Note that by Theorem 34, $\mathcal{O}^1$ is an $(\epsilon_{\text{op}}, \delta/(3L))$-linear system solver for $SBB^\top S^\top + \nu I_s$.
- Consequently, by Lemma 38, $\mathcal{O}^2$ is an $(\epsilon, L\delta_{\text{op}})$-linear system solver for $B^\top S^\top SB + \nu I_d$.
- Hence, by Theorem 34, $\mathcal{O}^3$ is an (ϵ, δ)-linear system solver for $B^\top B + \nu I_d$.

By a union bound over the failure probabilities induced by ξ_1, ξ_2, ξ_3, the correctness guarantee follows. Finally, to bound the complexity, we combine the complexity guarantees of Theorem 34 and Lemma 38 as well as the sparse embedding guarantee Lemma 32. ■

4. If $\mathcal{O}$ is deterministic, ξ_3 can be omitted.

B.4. Solving the embedded linear systems efficiently

In order to apply Lemma 39, we must show how to compute an efficient linear system solver matrix $SBTT^\top B^\top S^\top$ arising in Lemma 39 as well as support matvecs to SB and SBT efficiently. Here, we discuss this in greater detail.

In the remainder of this subsection, we fix $\bar{x} \in \mathbb{R}^d$, $A \in \mathbb{R}^{n\times d}$, $\lambda > 0$, $\epsilon' > 0$, $B = \frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2} A$, and $k := \|B\|_F^2/\lambda$. Furthermore, we let $S \in \mathbb{R}^{s\times n}, T^\top \in \mathbb{R}^{s\times d}$ denote random sparse embedding matrices with sketch size $s = O((k + \log(1/\delta)))$ and $b = O(\log^2(k/\delta) + \log^3(k/\delta))$ non-zeros per column (Definition 30). We will show that we can efficiently implement matvecs to SB and AT, as well as construct SBT explicitly.

Lemma 40 (Supporting matvecs to SB) *Given an explicit realization of S, one matvec to $SB \in \mathbb{R}^{s\times d}$ can be implemented in $O(b\,\mathrm{nnz}(A))$-time, using one matvec to A, $\tilde{O}(bn)$-additional work, and $\tilde{O}(1)$-additional depth.*

Proof To support one matvec to $SB \in \mathbb{R}^{s\times d}$, consider a query $(x, y) \in \mathbb{R}^d \times \mathbb{R}^s$. To compute SBx, note that we can first compute $u \leftarrow Bx = \frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2} Ax \in \mathbb{R}^n$ in $\mathrm{nnz}(A)$-work and then compute $SBx = Su$ in $O(bn)$-work, leveraging that each column of S has at most b nonzero entries.

To compute $y^\top SB$, note that we can first compute $u^\top \leftarrow y^\top S \in \mathbb{R}^n$ in $O(bn)$-work, again leveraging that each column of S has at most b nonzero entries. Then, we can compute $u^\top B = u^\top \frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2} A \in \mathbb{R}^d$ in $\mathrm{nnz}(A)$-work. ■

Fact 41 (Constructing AT) *Given an explicit realization of T, $AT \in \mathbb{R}^{n\times s}$ can be computed explicitly in in $O(s\,\mathrm{nnz}(A))$-work, using $\tilde{O}(1)$-depth and $\tilde{O}(s)$-matvecs to A.*

Proof This follows immediately, as $T \in \mathbb{R}^{d\times s}$ has s columns. ■

Lemma 42 (Constructing SBT) *Given an explicit $AT \in \mathbb{R}^{n\times s}$, $SBT \in \mathbb{R}^{s\times s}$ can be computed explicitly in $O(b^2\,\mathrm{nnz}(A) + s^2)$-work and $\tilde{O}(1)$-depth.*

Proof Since $AT \in \mathbb{R}^{n\times s}$ is given as input and $\frac{1}{\sqrt{\epsilon'}} P_{\bar{x},\epsilon'}^{1/2}$ is diagonal, in $O(\mathrm{nnz}(AT)) \le O(b\mathrm{nnz}(A))$-time (recall Lemma 31) we can compute $BT \in \mathbb{R}^{n\times s}$. Next, to compute $SBT \in \mathbb{R}^{s\times s}$, we apply the following procedure. For each $i \in [n]$, the i-th column of S has at most b nonzeros at rows $h_1(i), \ldots, h_b(i) \in [s]$ with corresponding values $v_1(i), \ldots, v_b(i) \in \{-1/\sqrt{b}, 1/\sqrt{b}\}$. We compute $C = SBT \in \mathbb{R}^{s\times s}$ as follows: Initialize $C \leftarrow 0 \in \mathbb{R}^{s\times s}$ and for each $i \in [n]$, for each $t \in [b]$ (for which $S_{h_t(i),i} = v_t(i) \neq 0$), update

$$C_{h_t(i),:} \;\leftarrow\; C_{h_t(i),:} + v_t(i)\,(BT)_{i,:}.$$

For any $r \in [s]$ and $j \in [s]$,

$$C_{rj} \;=\; (SBT)_{rj} \;=\; \sum_{i=1}^{n} S_{ri}(BT)_{ij}.$$

The above algorithm adds $(BT)_{i,:}$ into exactly those rows r for which $S_{ri} \neq 0$, and scales by S_{ri}, hence the final C equals SBT. For the runtime, each nonzero entry $(BT)_{ij}$ participates in at most b updates (one per nonzero in column i of S), so the total work is $O(b\,\mathrm{nnz}(BT))$. Using $\mathrm{nnz}(BT) = \mathrm{nnz}(AT) \leq b\,\mathrm{nnz}(A)$ (recall Lemma 31), this is $O(b^2\,\mathrm{nnz}(A))$-work. The initialization of the dense output $C \in \mathbb{R}^{s\times s}$ costs $O(s^2)$-work. Thus the overall work is $O(b^2\,\mathrm{nnz}(A) + s^2)$. ∎

B.5. Putting it all together

Here, we combine the previous results to obtain the main results of this section.

Lemma 43 *Let $A \in \mathbb{R}^{n\times d}$ with $\|A\|_{2\to\infty} \leq 1$, $\bar{x} \in \mathbb{B}^d$, $\lambda > 0$, $\epsilon, \epsilon' > 0, \delta \in (0,1)$, and $k = 1/(\lambda\epsilon')$. For any $b \in \mathbb{R}^d, \nu > \lambda$, let*

$$\mathcal{O}''(b) = \textsc{RankOneSolve}\left(\frac{1}{\epsilon'}P_{\bar{x},\epsilon'}B, \frac{1}{\sqrt{\epsilon'}}Ap_{\bar{x},\epsilon'}, \lambda, \epsilon, \delta, b, \mathcal{O}'_{\xi_1,\xi_2,\xi_3}\right),$$

where

$$\mathcal{O}'_{\xi_1,\xi_2,\xi_3}(b') = \textsc{SketchAndSolve}_{\xi_1,\xi_2,\xi_3}\left(B, \nu, \frac{1}{\lambda\epsilon'}, \frac{\epsilon}{1+\frac{n}{\epsilon'\lambda}}, \frac{\delta}{2}, b', \mathcal{O}\right).$$

and $\mathcal{O}$ instantiates FMM (Lemma 35). Then, $\mathcal{O}''$ is an (ϵ, δ)-linear system solver for

$$\frac{1}{\epsilon'}A^\top(P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'}p_{\bar{x},\epsilon'}^\top)A + \nu I_d,$$

and each query to $\mathcal{O}''$ does $\tilde{O}(\epsilon^{-2/3}\,\mathrm{nnz}(A))$-work using $\tilde{O}(1)$-depth and $\tilde{O}(\epsilon^{-2/3})$-matvecs to A.

Proof First, let

$$B = \frac{1}{\sqrt{\epsilon'}}P_{\bar{x},\epsilon'}^{1/2}A,$$

and note that $\|B\|_F^2 \leq {\epsilon'}^{-1}$ (by the argument as in (8)) and consequently $k = \|B\|_F^2/\epsilon'$. Moreover, observe that

$$\|B\|_2^2 \leq \frac{1}{\epsilon'}\|A\|_2^2 \leq {\epsilon'}^{-1}\|A\|_F^2 \leq \frac{1}{\epsilon'}\sum_{i\in[n]}\|A_{i,:}\|_2^2 \leq \frac{n}{\epsilon'}\|A\|_{2\to\infty}^2 = \frac{n}{\epsilon'}.$$

Similarly,

$$\|Ap_{\bar{x},\epsilon'}\|_2^2 \leq \|A\|_2^2\|p_{\bar{x},\epsilon'}\|_2^2 \leq \frac{n}{\epsilon'},$$

since $\|p_{\bar{x},\epsilon'}\|_2 \leq \|p_{\bar{x},\epsilon'}\|_1 = 1$. Thus, the correctness follows directly from Lemma 39 and Lemma 37. The complexity follows directly from Lemma 40, Fact 41, Lemma 42, and Lemma 35. ∎

Lemma 44 (SVRG solver with fresh embeddings) *Let $A \in \mathbb{R}^{n\times d}$ with $\|A\|_{2\to\infty} \le 1$, $\bar{x} \in \mathbb{B}^d$, $\lambda > 0$, $\epsilon, \epsilon' > 0, \delta \in (0,1)$, and $k = 1/(\lambda\epsilon')$. For any $b \in \mathbb{R}^d, \nu > \lambda$, let*

$$\mathcal{O}''(b) = \textsc{RankOneSolve}\left(\frac{1}{\epsilon'}P_{\bar{x},\epsilon'}B, \frac{1}{\sqrt{\epsilon'}}Ap_{\bar{x},\epsilon'}, \lambda, \epsilon, \delta, b, \mathcal{O}'_{\xi_1,\xi_2,\xi_3}\right),$$

where

$$\mathcal{O}'_{\xi_1,\xi_2,\xi_3}(b') = \textsc{SketchAndSolve}_{\xi_1,\xi_2,\xi_3}\left(B, \nu, \frac{1}{\lambda\epsilon'}, \frac{\epsilon}{1+\frac{n}{\epsilon'\lambda}}, \frac{\delta}{2}, b', \mathcal{O}\right).$$

and $\mathcal{O}$ instantiates SVRG (Lemma 36). Then, $\mathcal{O}''$ is an (ϵ,δ)-linear system solver for

$$\frac{1}{\epsilon'}A^\top(P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'}p_{\bar{x},\epsilon'}^\top)A + \nu I_d,$$

and each query to $\mathcal{O}''$ does $\tilde{O}(\epsilon^{-2/3}\,\mathrm{nnz}(A))$-work using $\tilde{O}(1)$-depth and $\tilde{O}(\epsilon^{-2/3})$-matvecs to A.

Proof Correctness follows identically as the proof of Lemma 43. The complexity follows directly from Lemma 40, Fact 41, Lemma 42, and Lemma 36. ■

In order to obtain Theorems 2 and Theorems 3, in the following section we show that it is actually possible to *reuse* the same T across *every* linear system solve in Lemma 7. This enables us to pre-compute $AT^\top$ *once* in the very first iteration, using $\tilde{O}(s\,\mathrm{nnz}(A))$-work, $\tilde{O}(s)$-matvecs to A and $\tilde{O}((n+d))$-depth and reuse it in all future iterations. Correspondingly, we reduce the auxiliary work and depth in Lemmas 44 and 43.

Appendix C. Main results

In this section, we discuss how the sample reuse framework of (Jin et al., 2026) immediately enables us to reuse the same seed ξ_2 across sequential invocations of Lemmas 43 and 44 when used to instantiate the linear system solver for Lemma 7, and we use this to obtain our main results. Throughout this section, we use $\tilde{O}(\cdot)$ to hide polylogarithmic factors in the following parameters: $n, d, \epsilon^{-1}, \delta^{-1}, \lambda^{-1}, \nu^{-1}$.

In order to directly apply the sample reuse framework of (Jin et al., 2026), we need to check the three key assumptions that enable their framework.

First note that $\xi_2 \sim \mathcal{P}_2$ in Lemmas 43 and 44 is used to sample a sparse oblivious subspace embedding (Definition 30) in lemma 39, meaning that $\mathcal{P}_2$ is an oblivious sampling distribution (i.e., $\mathcal{P}_2$ depends only on the sketch size $s = \tilde{O}(\epsilon^{-2/3})$ and number of nonzeros $b = \tilde{O}(1)$.)

Second, note that the requirements of the outer loop in Lemma 7, which orchestrates the sequence of linear system solves, is ℓ_∞-robust in the sense of Definition 2.2 of (Jin et al., 2026), as illustrated by the following lemma.

Lemma 45 (ℓ_∞-robustness) *Suppose that $x' \in \mathbb{R}^d$ satisfies*

$$\left\|x' - \left(\epsilon' A^\top(P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'}p_{\bar{x},\epsilon'}^\top)A + \nu I\right)^{-1}\hat{g}\right\|_2 \le \alpha$$

for some $\epsilon', \nu, \alpha > 0, \bar{x} \in \mathbb{R}^d, A \in \mathbb{R}^{n\times d}$ and suppose that $\zeta \in \mathbb{R}^d$ satisfies $\|\zeta\|_\infty < \tau$. Then,

$$\left\|(x'+\zeta) - \left(\epsilon' A^\top(P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'}p_{\bar{x},\epsilon'}^\top)A + \nu I\right)^{-1}\hat{g}\right\|_2 \le \alpha + \tau\sqrt{d}.$$

Proof Note that for any $x \in \mathbb{R}^d$, $\|x\|_\infty \le \|x\|_2 \le \sqrt{d}\|x\|_\infty$. Consequently, we have

$$\begin{aligned}&\left\|(x' + \zeta) - \left(\epsilon' A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'} p_{\bar{x},\epsilon'}^\top) A + \nu I\right)^{-1} \hat{g}\right\|_2 \\ &\le \left\|x' - \left(\epsilon' A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'} p_{\bar{x},\epsilon'}^\top) A + \nu I\right)^{-1} \hat{g}\right\|_\infty + \|\zeta\|_2 \\ &\le \alpha + \tau\sqrt{d}.\end{aligned}$$

■

Consequently, the requirements of Lemma 7 are robust to any τ-bounded ℓ_∞ perturbation, provided that the accuracy requirement of the linear system solves is correspondingly lowered by a factor of $\tau/\sqrt{d}$.

Third, note that the complexities of Lemmas 43 and 44 depend at most *polylogarithmically* on the required solver accuracy ϵ (i.e., in the language of (Jin et al., 2026), the linear system solvers can solve to *high accuracy*). Consequently, lowering the required accuracy of the linear system solves by a factor of $\tau/\sqrt{d}$ comes at the cost of at most polylogarithmic overhead in the key problem parameters.

Thus, by a straightforward application of Theorem 2.6 of (Jin et al., 2026), we see that the same random seed ξ_2 can be used across all sequential invocations of Lemmas 43 and 44 when used to instantiate the linear system solver for Lemma 7.

Finally, we are prepared to prove our main results. First, using an FMM-based linear system solver, we prove the following guarantee.

Theorem 46 (Final FMM-based solver) *There is an algorithm which solves the* $(A \in \mathbb{R}^{n\times d}, \epsilon > 0)$*-game (Definition 5) in* $\tilde{O}(\epsilon^{-2/3}\mathrm{nnz}(A) + \epsilon^{-2(\omega+1)/3})$*-work, using* $\tilde{O}(\epsilon^{-2/3})$ *matvecs to* A *and* $\tilde{O}(\epsilon^{-2/3})$*-depth.*

Proof By Lemma 7, we see that it suffices to show that for any $\bar{x} \in \mathbb{R}^d$ and $\epsilon, \delta > 0$ we can implement an (ϵ, δ)-linear system solver for

$$\frac{1}{\epsilon'} A^\top (P_{\bar{x},\epsilon'} - p_{\bar{x},\epsilon'} p_{\bar{x},\epsilon'}^\top)^{-1} A + \nu I$$

for $\nu \ge \lambda = \tilde{\Omega}(\epsilon^{-1/3})$, which runs in $\tilde{O}(\mathrm{nnz}(A) + \epsilon^{-2\omega/3})$-work and $\tilde{O}(n+d)$-parallel depth and makes at most $\tilde{O}(1)$-matvec queries to A.

To this end, let k, s, b as in Lemma 43 and 39 and let $T_*^\top \in \mathbb{R}^{s\times d}$ be a sparse embedding matrix (Definition 30) with sketch size $s = \tilde{O}(\epsilon^{-2/3})$ and $b = \tilde{O}(1)$ non-zeros per column. Note that by Fact 41, we can compute $AT_* \in \mathbb{R}^{n\times s}$ in $\tilde{O}(s\,\mathrm{nnz}(A))$-time, $\tilde{O}(1)$-parallel-depth, and $\tilde{O}(s)$-matvecs to A.

By combining Lemma 45 and Theorem 2.2 of (Jin et al., 2026), the guarantees of Theorem 43 remain unchanged if the random seed ξ_2 is fixed across all invocations of $\tilde{\mathcal{O}}$. Consequently, by Lemmas 42 and 40, the complexity of Lemma 43 can be reduced to $\tilde{O}(b\,\mathrm{nnz}(A)) = \tilde{O}(\mathrm{nnz}(A))$-time and only $\tilde{O}(1)$ matvecs to A, by reusing the precomputed AT_* in all invocations. ■

Second, using an SVRG-based linear system solver, we prove the following guarantee.

Theorem 47 (Final SVRG-based solver) *There is an algorithm which solves the* $(A \in \mathbb{R}^{n\times d}, \epsilon > 0)$*-game (Definition 5) in* $\tilde{O}(\epsilon^{-2/3}\mathrm{nnz}(A) + \epsilon^{-2})$*-work, using* $\tilde{O}(\epsilon^{-2/3})$ *matvecs to* A *and* $\tilde{O}(\epsilon^{-2/3})$*-depth.*

Proof The proof follows identically as that of Theorem 46, except that we leverage Lemma 44 in place of Lemma 43. ■

Theorems 2 and 3 now follow immediately from Lemma 6 and Theorems 46 and 47, respectively.

Appendix D. Reduction from separating hyperplane problem to ℓ_2-ℓ_1 games

Lemma 6 (Reduction to ℓ_2-ℓ_1 matrix game)
Let $\mathcal{D} = \{\phi_i \in \mathbb{B}^d, l_i \in \{+1, -1\}\}_{i\in[n]}$*. Any solution* $\hat{w}$ *to the* ρ*-game (Definition 5) of* $-\Phi_{\mathcal{D}}$ *induces a* ρ*-separating hyperplane (Definition 1) for* $\mathcal{D}$*.*

Proof To reformulate the $(\mathcal{D}, \rho)$-separating hyperplane problem as an ℓ_2-ℓ_1 matrix game, note that (1) is equivalent to

$$\max_{w\in\mathbb{B}^d}\min_{i\in[n]} l_i\langle w, \phi_i\rangle = \max_{w\in\mathbb{B}^d}\min_{i\in[n]}[\Phi_{\mathcal{D}}w]_i = \max_{w\in\mathbb{B}^d}\min_{p\in\Delta^n} p^\top\Phi_{\mathcal{D}}w = -\min_{w\in\mathbb{B}^d}\max_{p\in\Delta^n} p^\top(-\Phi_{\mathcal{D}})w.$$

By (1), there exists $w^\star \in \mathbb{B}^d$ such that

$$\min_{p\in\Delta^n} p^\top\Phi_{\mathcal{D}}w^\star = \min_{i\in[n]}[\Phi_{\mathcal{D}}w^\star]_i = \min_{i\in[n]} l_i\langle w^\star, \phi_i\rangle \geq \gamma_{\mathcal{D}}.$$

Therefore,

$$\max_{w\in\mathbb{B}^d}\min_{p\in\Delta^n} p^\top\Phi_{\mathcal{D}}w \geq \gamma_{\mathcal{D}}, \qquad \text{and consequently} \qquad \min_{w\in\mathbb{B}^d}\max_{p\in\Delta^n} p^\top(-\Phi_{\mathcal{D}})w \leq -\gamma_{\mathcal{D}}.$$

Let $\hat{w}$ be a solution to the $(-\Phi_{\mathcal{D}}, \gamma_{\mathcal{D}}/2, \rho)$-game. Then,

$$\max_{p\in\Delta^n} p^\top(-\Phi_{\mathcal{D}})\hat{w} \leq \min_{w\in\mathbb{B}^d}\max_{p\in\Delta^n} p^\top(-\Phi_{\mathcal{D}})w + \rho \leq -\gamma_{\mathcal{D}} + \rho.$$

Consequently, we have $\max_{i\in[n]}[(-\Phi_{\mathcal{D}})\hat{w}]_i < -\gamma_{\mathcal{D}} + \rho < 0$, which implies that

$$l_i\langle \hat{w}, \phi_i\rangle = [\Phi_{\mathcal{D}}\hat{w}]_i > 0, \text{ for all } i \in [n].$$

Thus, $\hat{w}$ is a solution to the $(\mathcal{D}, \rho)$-separating hyperplane problem. ■

Appendix E. Linear algebraic properties

Fact 48 *Suppose that* $A \in \mathbb{S}^n_{++}$ *and* x' *is an* ϵ*-approximate solution to the linear system* $Ax' = b$ *(Definition 4). Then,* $\|x - x'\|_2 \leq \sqrt{\kappa(A)}\epsilon\|x\|_2$*.*

Proof We have $\|x' - x\|_A \leq \gamma\|x\|_A$. Since $A \succeq \lambda_{\min}(A)I$ it follows that

$$\|x' - x\|_2 \leq \frac{\|x' - x\|_A}{\sqrt{\lambda_{\min}(A)}} \leq \gamma\frac{\|x\|_A}{\sqrt{\lambda_{\min}(A)}}.$$

Moreover, $\|x\|_A \leq \sqrt{\lambda_{\max}(A)}\|x\|_2$. Thus, the lemma holds. ■

Fact 49 *Suppose that $A, B \in \mathbb{S}^d_{++}$ with $A \approx_c B$ for some $c \geq 1$. Then,*

- *for any $x \in \mathbb{R}^d$, $\|x\|_A \approx_{\sqrt{c}} \|x\|_B$,*
- $\|A\|_F \approx_c \|B\|_F$,
- *for any $b \in \mathbb{R}^d$, $\|B^{-1}b\|_B \approx_{\sqrt{c}} \|A^{-1}b\|_A$, and*
- *for any $b \in \mathbb{R}^d$, $\|A^{-1}b - B^{-1}b\|_B \leq (c-1)\|B^{-1}b\|_B$,*

Proof We prove the statements one-by-one.

- Since $c^{-1}A \preceq B \preceq cA$ implies $c^{-1}B \preceq A \preceq cB$, for any $x \in \mathbb{R}^d$ we have

$$c^{-1}x^\top Bx \leq x^\top Ax \leq cx^\top Bx,$$

  which yields $\|x\|_A \approx_{\sqrt{c}} \|x\|_B$.

- From $c^{-1}A \preceq B \preceq cA$, the eigenvalues of B are within a factor c of those of A (due to the Courant-Fischer theorem) Since $\|M\|_F^2 = \sum_i \lambda_i(M)^2$ for $M \succeq 0$, it follows that $\|A\|_F \approx_c \|B\|_F$.

- Inverting the Loewner order gives $c^{-1}A^{-1} \preceq B^{-1} \preceq cA^{-1}$. Therefore, for any $b \in \mathbb{R}^d$,

$$b^\top B^{-1}b \approx_c b^\top A^{-1}b,$$

  which is equivalent to $\|B^{-1}b\|_B \approx_{\sqrt{c}} \|A^{-1}b\|_A$.

- Writing $b = Bx$, we have

$$A^{-1}b - B^{-1}b = (A^{-1}B - I)x.$$

  Since the eigenvalues of $B^{1/2}A^{-1}B^{1/2}$ lie in $[c^{-1}, c]$, we have $\|A^{-1}B - I\|_B \leq c - 1$, and

$$\|A^{-1}b - B^{-1}b\|_B \leq (c-1)\|x\|_B = (c-1)\|B^{-1}b\|_B.$$

■