The Thesis Committee for Jonathan Bryan Li
certifies that this is the approved version of the following thesis:

Specialized Solvers for Structured Convex Programs in High-Dimensional Statistics

SUPERVISING COMMITTEE:

Kevin Tian, Supervisor

Qiang Liu

Specialized Solvers for Structured Convex Programs in High-Dimensional Statistics

by

Jonathan Bryan Li

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Computer Science

The University of Texas at Austin

May 2025

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

T. S. Eliot

Every valley shall be lifted up,
and every mountain and hill be made low;
the uneven ground shall become level,
and the rough places a plain.

Isaiah 40:4 (ESV)

Acknowledgments

This thesis would not have been possible without the support, guidance, and encouragement of many incredible individuals, each of whom has contributed in meaningful ways to its completion.

First and foremost, I would like to thank my advisor, Kevin Tian. It is no exaggeration to say that I might never have pursued research had I not taken Kevin's Continuous Algorithms class. It was Kevin's first time teaching a class at UT, but his enthusiasm for the subject and his remarkable talent for conveying complex ideas in intuitive ways sparked my own curiosity for theoretical computer science. As an advisor and mentor, Kevin has been a seemingly infinite source of wisdom and patience, teaching me not only technical skills like how to write a bibliography or better understand linear algebra, but also invaluable lessons in how to communicate ideas clearly and organize research thoughtfully. There were times when I procrastinated or asked far too many trivial questions, leading me to wonder whether I truly deserved such a supportive advisor. Yet Kevin never lost patience, always encouraging me and always helping me grow.

Next, I would like to thank Arun Jambulapati for being an exceptional collaborator and mentor. Meeting Arun after having read so many of his papers was a surreal and inspiring experience — his brilliance in person matched what I had admired on paper. Arun motivated me to dive deeper into areas such as spectral graph theory and numerical linear algebra, and I continue to be awed by both the breadth

and depth of his knowledge.

I would like to thank everyone else in the Tian Group — Syamantak, Shourya, Chutong, Yusong, Jennifer, and Saloni — for a wonderful year filled with thought-provoking reading groups and research discussions. I hope to continue learning about diffusion models and differential privacy in the future.

I am deeply grateful to Qiang Liu for serving as a reader for this thesis and for introducing me to the field of machine learning theory. I would also like to thank the rest of UT Statistical Learning & AI Group and the entire GDC 4^{th} floor for fostering such an intellectually stimulating and supportive environment.

To Nathaniel and Gracie, two of my closest friends throughout college — thank you for your constant advice, encouragement, and companionship. From Seoul to Austin to Tromsø, we have been on countless adventures together, and I cannot wait to embark on our next one. I would also like to thank Arthur, Steph, Simmy, Baltej, and Jin for all the laughter and great memories we have shared over the years.

To Minshin, Toby, Gloria, Jaeni, Erin, Faith, James, Lydia, Dogyu, Hannah, and Jay — thank you for being the best life group I could have asked for. We experienced many ups and downs together, and I am grateful for how we grew as a result. Your support provided me the encouragement and motivation to complete this thesis, and I will always associate my thesis-writing days with the memories that we made this year. I would also like to thank everyone else at AKPC for being part of such a compassionate and faithful community.

To my childhood friends — Ben, Sharan, Kev, and Matthew — thank you for helping me recognize my potential in math and encouraging me to pursue great ambitions. I would not be where I am today without your influence and support.

Min Joo, thank you for being such a great friend and role model over the years. From our silly antics during the pandemic to your down-to-earth advice, you have always been someone I have learned from and admired. You are, in so many ways, truly an inspiration.

It is difficult to express in words the gratitude that I have for my parents and sister. They recognized and nurtured my interest in math from an early age and have always been an unwavering source of love and support. Knowing that I can always rely on them brings me comfort during challenging times — something I often take for granted but deeply appreciate.

Finally, I would like to thank the almighty God for bringing each of these wonderful people into my life and for His constant grace and love. *Soli Deo Gloria.*

Preface

This thesis was written during my fifth and final year at The University of Texas at Austin, as part of the Integrated Five-Year Bachelor's and Master's Program in Computer Science. It represents the culmination of my graduate studies, shaped by an interest in the interplay between optimization, machine learning, and theoretical computer science.

The goal of this thesis is to develop and analyze fast algorithms for solving problems in high-dimensional statistics that can be formulated as convex programs. By leveraging the structure of specific problems, we provide algorithms that significantly outperform black-box convex programming solvers in both theory and practice, particularly in the high-dimensional regime.

This thesis is organized into two main chapters, each corresponding to a distinct research problem and time period in my graduate work.

Chapter 2 roughly corresponds to work conducted between July and October 2024. It focuses on the analysis of an algorithm for solving a specific class of packing linear programs where the constraint is given by the Ky Fan k-norm. We also discuss an application of Ky Fan packing to fair principal component analysis, a concept that may be of interest to the trustworthy machine learning community.

Chapter 3 roughly corresponds to work conducted from November 2024 to April 2025. It is a simplified version of a paper written with Arun Jambulapati and

Kevin Tian, focusing on a fast algorithm for transforming datasets into a strong geometric condition known as radial isotropic position. The algorithm is based on an implicit, box-constrained Newton's method, where a sparsification subroutine is used to speed up the computation of matrix-vector products. We omit the details of the sparsification algorithm but give a comprehensive analysis of the main algorithm in both the well-conditioned and smoothed analysis settings.

Abstract

Specialized Solvers for Structured Convex Programs in High-Dimensional Statistics

Jonathan Bryan Li, MSCompSci
The University of Texas at Austin, 2025

SUPERVISOR: Kevin Tian

Convex programming is a fundamental tool of modern optimization, with applications in areas such as machine learning, combinatorial optimization, and graph theory. Despite their broad utility, general-purpose convex programming solvers such as cutting-plane methods and interior-point methods often struggle to scale efficiently in high-dimensional or large-scale settings. To address these limitations, we develop specialized algorithms that bypass the computational bottlenecks of black-box solvers. Specifically, we give efficient algorithms for solving Ky Fan packing linear programs and for computing approximate Forster transforms — two problems with a plethora of applications — and show that our algorithms significantly improve upon state-of-the-art runtimes based on standard convex optimization techniques.

Contents

Chapter 1

Introduction

Modern data science and statistical learning theory are characterized by increasingly complex computational problems. As datasets grow in both size and dimensionality, traditional methods often struggle to provide efficient, robust, or fair algorithms for statistical primitives. The goal of this thesis is to address several algorithmic challenges at the intersection of theoretical computer science and practical data analysis, with a particular focus on structured convex programs.

In recent decades, convex programming has emerged as a powerful technique in mathematical optimization, with widespread applications ranging from control theory to machine learning to combinatorial optimization. Although black-box convex programming solvers have been known for decades, these traditional algorithms often fail to capitalize on the structure of problems in high-dimensional statistics, leading to poor scaling behavior for modern datasets.

In this thesis, we will specialize our discussion to the following problems.

Packing semidefinite programs. A subclass of convex programs, these problems, as well as their dual covering semidefinite programs, investigate optimal convex combinations of vectors and matrices under various norm constraints. Solvers for packing

semidefinite programs can often be used as a black-box subroutine in practical algorithms for, e.g., sparse recovery, clustering, and principal component analysis.

Radial isotropic position. In many data-driven applications, it is desirable that the dataset satisfies certain regularity conditions after a preprocessing step. Radial isotropic position is a strong geometric condition that combines the well-known regularizations of normalization and isotropic position, and it has found applications in functional analysis, communication complexity, coding theory, and the design of learning algorithms. Previous algorithms for radial isotropic position were based on applying black-box convex optimization techniques, and we give a significantly faster algorithm by designing a custom solver.

1.1 Black-box convex programming

Semidefinite programming (SDP) is a generalization of linear programming (LP) and quadratic programming (QP) that optimizes a linear objective function over a spectrahedron, i.e. the intersection of the positive semidefinite cone with an affine space. The standard form of an SDP is

$$\max_{\substack{\langle \mathbf{A}_i, \mathbf{X} \rangle \leq \mathbf{b}_i \ \forall i \in [n] \\ \mathbf{X} \in \mathbb{S}^{d \times d}_{\succeq \mathbf{0}}}} \langle \mathbf{C}, \mathbf{X} \rangle \,, \tag{1.1}$$

[1]

where $\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{C} \subset \mathbb{S}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^n$. A standard duality result shows that the dual of (1.1) is

$$\min_{\substack{\sum_{i \in [n]} \mathbf{y}_i \mathbf{A}_i \succeq \mathbf{C} \\ \mathbf{y} \in \mathbb{R}^n_{\geq 0}}} \mathbf{b}^\top \mathbf{y}. \tag{1.2}$$

Both the primal and dual forms of SDPs have been well-studied in the literature, with applications in areas such as machine learning [LCB+04, dGJL07, AW08, RSL18,

[1] See Section 1.3 for notation used throughout the thesis.

JLT20, Zha20], combinatorial optimization [GW94, KMS98, ARV09], and graph theory [MS16, LS17, GSW22]. Polynomial-time high-accuracy SDP solvers have existed for decades, and there are long lines of work dedicated to cutting-plane methods [Kha80, GLS81, Vai96, GV02, BV04a, LSW15, JLSW20] and interior-point methods [Kar84, NN94, Ans00, JKL+20, HJS+22] for solving SDPs.

Cutting-plane methods, and sometimes interior-point methods, extend to general convex programs, including our approach to radial isotropic position. However, even state-of-the-art cutting-plane methods are far too slow for practical applications, which motivates the development of specialized solvers.

1.2 Overview

This thesis is divided into two main chapters. In Chapter 2, we investigate Ky Fan packing LPs and an application to fair principal component analysis. In Chapter 3, we give a fast algorithm for computing approximate Forster transforms.

1.2.1 Primary objectives

Our primary objective in Chapter 2 is to provide an algorithm that solves a Ky Fan packing LP, a natural variant of the well-studied ℓ_∞ packing LP. The problem can roughly be stated as follows: given $\mathbf{A} \in \mathbb{R}^{d\times n}_{\geq 0}$, find primal solution $\mathbf{x} \in \Delta^n$ with $\|\mathbf{A}\mathbf{x}\| \leq 1+\epsilon$ or dual solution $\mathbf{y} \in \mathcal{Y}$ with $\mathbf{A}^\top \mathbf{y} \geq (1-\epsilon)\mathbf{1}$, where $\|\cdot\|$ is a suitable norm and $\mathcal{Y}$ is a suitable dual set.

Our primary objective in Chapter 3 is to design a fast algorithm for computing approximate Forster transforms, which can be thought of as finding an invertible $\mathbf{R} \in \mathbb{R}^{d\times d}$ such that the unit vectors $\{(\mathbf{R}\mathbf{a}_i)\,\|\mathbf{R}\mathbf{a}_i\|_2^{-1}\}_{i\in[n]}$ are in approximate isotropic position, where $\{\mathbf{a}_i\}_{i\in[n]} \subset \mathbb{R}^d$ is the given dataset. Here, our notion of approximate isotropic position means that the second moment matrix is within a $\exp(\pm\epsilon)$ factor of $\mathbf{I}_d$.

1.2.2 Main contributions

Our main contributions consist of the following four theorems, where we give brief, informal statements. Formal statements and detailed discussions of prior and related work can be found in the respective chapters.

Theorem 1.1 (Informal, see Theorem 2.1)**.** *There is an algorithm that solves the ϵ-approximate Ky Fan k-norm packing LP for $\mathbf{A} \in \mathbb{R}^{d \times n}_{\geq 0}$ in time*

$$\widetilde{O}\left(\mathrm{nnz}(\mathbf{A}) \cdot \frac{k}{\epsilon^2}\right).$$

Theorem 1.2 (Informal, see Theorem 3.1)**.** *In the well-conditioned setting, there is an algorithm that computes an ϵ-approximate Forster transform, with probability $\geq 1 - \delta$, in time*

$$O\left(nd^{\omega-1}\left(\frac{n}{\delta\epsilon}\right)^{o(1)}\right).$$

Theorem 1.3 (Informal, see Theorem 3.2)**.** *Let $\mathbf{\Pi} := \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. There is an algorithm that takes a matrix-vector product oracle $\mathcal{O}$ for a graph Laplacian $\mathbf{L}$ and returns a graph Laplacian $\widetilde{\mathbf{L}}$ satisfying*

$$\mathbf{L} + \Delta\mathbf{\Pi} \preceq \widetilde{\mathbf{L}} \preceq \left(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta}\right)^{o(1)} (\mathbf{L} + \Delta\mathbf{\Pi}) \text{ and } \mathrm{nnz}(\widetilde{\mathbf{L}}) = n \cdot \left(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta}\right)^{o(1)}$$

using $(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta})^{o(1)}$ queries to $\mathcal{O}$ and $n \cdot (\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta})^{o(1)}$ additional time.

Theorem 1.4 (Informal, see Theorem 3.3)**.** *In the smoothed analysis setting, there is an algorithm that computes an ϵ-approximate Forster transform, with probability $\geq 1 - \delta$, in time*

$$O\left(nd^{\omega}\left(\frac{n}{\delta\epsilon}\right)^{o(1)}\right).$$

To our knowledge, all of these runtimes are state-of-the-art for the problems that they solve. We conjecture that the runtime in Theorem 1.1 can be improved, which we discuss in more detail in Section 2.2.1.

1.3 Notation

Throughout this thesis, we will use the following notation.

General notation. For $d \in \mathbb{N}$, $[d] := \{1, 2, \ldots, d\} = \{i \in \mathbb{N} \mid i \leq d\}$. Vectors are denoted in lowercase boldface, and matrices are denoted in capital boldface. $\widetilde{O}$ hides polylogarithmic factors in the problem parameters. $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ denote a norm and inner product, respectively. In particular, $\|\cdot\|_p$ denotes the ℓ_p or Schatten p-norm, and $\|\cdot\|_{\mathrm{op}}$, $\|\cdot\|_{\mathrm{tr}}$, and $\|\cdot\|_{\mathrm{F}}$ denote the Schatten ∞-, 1-, and 2-norms, respectively. For a norm $\|\cdot\|$, $\mathbf{x} \in \mathbb{R}^d$, and $r > 0$, $\mathbb{B}_{\|\cdot\|}(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{y}\| \leq r\}$ denotes the ball of radius r around $\mathbf{x}$; if unspecified, $\|\cdot\| = \|\cdot\|_2$ and $\mathbf{x} = \mathbf{0}$ is assumed. ∇^k denotes the k^{th} partial derivative tensor of a k-times differentiable multivariate function. $\mathbf{e}_i$ denotes the i^{th} standard basis vector of the vector space over $\mathbb{R}$ of the appropriate dimension. $\mathbb{I}_{\mathcal{E}}$ denotes the indicator function of event $\mathcal{E}$. $\mathrm{conv}(S)$ denotes the convex hull of a set S.

Matrices and tensors. $\mathbf{0}$ and $\mathbf{1}$ denote the all-zeroes and all-ones tensors, respectively, and $\mathbf{I}$ denotes the identity matrix; when the dimensions are not specified in a subscript, the appropriate dimensions are assumed. Vectors in $\mathbb{R}^d$ are assumed to be $d \times 1$ matrices when appropriate. $\mathrm{Span}(\mathbf{A})$, $\mathrm{rank}(\mathbf{A})$, $\mathrm{nnz}(\mathbf{A})$, $\mathbf{A}^\top$, and $\mathbf{A}^\dagger$ denote the span, rank, number of nonzero entries, transpose, and Moore–Penrose inverse of a matrix $\mathbf{A}$, respectively. $\mathrm{Tr}(\mathbf{A})$ denotes the trace of $\mathbf{A} \in \mathbb{R}^{d \times d}$. For $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $k \in [\min(n, d)]$, $\boldsymbol{\sigma}_k(\mathbf{A})$ denotes the k^{th} largest singular value of $\mathbf{A}$. For real matrices $\mathbf{A}$ and $\mathbf{B}$ with the same dimensions, $\langle \mathbf{A}, \mathbf{B} \rangle := \mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$ denotes the Frobenius inner product. $\circ$ denotes the entrywise (Hadamard) product. ω denotes the matrix multiplication exponent, currently known to be approximately 2.37134 [ADV+25]. $\mathcal{T}_{\mathrm{mv}}(\mathbf{A})$ denotes the time required to compute $\mathbf{A}\mathbf{v}$ for an arbitrary $\mathbf{v}$. For a k-way tensor $\mathbf{T}$ operating on a set of $\ell \in [k]$ inputs $\{\mathbf{v}_1, \ldots, \mathbf{v}_\ell\}$, $\mathbf{T}[\mathbf{v}_1, \ldots, \mathbf{v}_\ell]$ denotes the resulting $(k - \ell)$-way tensor, e.g. $\mathbf{M}[\mathbf{u}, \mathbf{v}] = \mathbf{u}^\top \mathbf{M} \mathbf{v}$ when $\mathbf{M}$ is a matrix. $\mathbf{A}_{i:}$ and $\mathbf{A}_{:j}$ denote the i^{th} row and j^{th} column of a matrix $\mathbf{A}$, respectively; for sets of indices I and J,

$\mathbf{A}_{I:}$, $\mathbf{A}_{:J}$, and $\mathbf{A}_{I:J}$ denote the submatrices $(\mathbf{A}_{i:})_{i\in I}$, $(\mathbf{A}_{:j})_{j\in J}$, and $(\mathbf{A}_{ij})_{i\in I, j\in J}$, respectively. For $\mathbf{A} \in \mathbb{R}^{d\times d}$, $I \subseteq [d]$, and $J := [d] \setminus I$, $\mathrm{SC}(\mathbf{A}, I) := \mathbf{A}_{I:I} - \mathbf{A}_{I:J}\mathbf{A}_{J:J}^{\dagger}\mathbf{A}_{J:I}$ denotes the Schur complement onto $\mathbf{A}_{I:I}$.

Symmetric matrices. $\mathbb{S}^{d\times d}$, $\mathbb{S}^{d\times d}_{\succ \mathbf{0}}$, and $\mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ denote the sets of real symmetric, positive definite, and positive semidefinite (PSD) $d \times d$ matrices, respectively, and $\preceq$ denotes the Loewner order. For $\mathbf{w} \in \mathbb{R}^d$, $\mathbf{diag}\,(\mathbf{w})$ denotes the diagonal matrix with diagonal entries given by $\mathbf{w}$. For $\mathbf{A} \in \mathbb{S}^{d\times d}$ with spectrum $\boldsymbol{\lambda}$ and eigendecomposition $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ (as guaranteed by Fact A.1) and a function $f : \boldsymbol{\lambda} \to \mathbb{R}$, $f(\mathbf{A})$ denotes the matrix $\mathbf{U}f(\boldsymbol{\Lambda})\mathbf{U}^\top$, where $f(\boldsymbol{\Lambda})$ is applied on the diagonal entrywise. For $\mathbf{A}, \mathbf{B} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ and $\epsilon > 0$, $\mathbf{A} \approx_\epsilon \mathbf{B}$ denotes the chain of inequalities $\exp(-\epsilon)\mathbf{A} \preceq \mathbf{B} \preceq \exp(\epsilon)\mathbf{A}$. We will also use the obvious extension of this notion to nonnegative scalars and vectors. $\boldsymbol{\Pi}_E \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ denotes the projection matrix onto a subspace $E \subseteq \mathbb{R}^d$. For $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, $\|\cdot\|_{\mathbf{A}}$ denotes the Mahalanobis seminorm induced by $\mathbf{A}$, i.e. $\|\mathbf{v}\|_{\mathbf{A}} = \sqrt{\mathbf{v}^\top \mathbf{A}\mathbf{v}}$.

Probability. $\Delta^n := \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \mid \|\mathbf{x}\|_1 = 1\}$ denotes the n-dimensional probability simplex. $\sim_{\text{i.i.d.}}$ denotes that a collection of random variables is independent and identically distributed according to a given probability distribution. For $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \in \mathbb{S}^{d\times d}_{\succ \mathbf{0}}$, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the d-variate Gaussian distribution, i.e. the probability distribution on $\mathbb{R}^d$ with probability density function given by

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right).$$

For probability measures P and Q on Ω that are absolutely continuous with respect to a measure μ,

$$D_{\mathrm{KL}}(P \parallel Q) := \int_\Omega p(x) \log\left(\frac{p(x)}{q(x)}\right) \mu(\mathrm{d}x)$$

denotes the Kullback–Leibler divergence between P and Q.

Chapter 2

Ky Fan Packing

In this chapter, we extend the work of [MRWZ16, JLT20, DKK+21] to obtain a fast algorithm for solving Ky Fan packing linear programs. In Section 2.3, we then discuss potential applications to fairness in data analysis through a variant of the standard principal component analysis technique.

2.1 Packing linear programs

In its standard form, a *packing linear program* is given by

$$\max_{\substack{\mathbf{x}\in\mathbb{R}^n_{\geq 0}\\ \mathbf{A}\mathbf{x}\leq\mathbf{1}}} \mathbf{1}^\top\mathbf{x}, \tag{2.1}$$

where $\mathbf{A}\in\mathbb{R}^{d\times n}_{\geq 0}$. The dual of (2.1) is the *covering linear program*

$$\min_{\substack{\mathbf{y}\in\mathbb{R}^d_{\geq 0}\\ \mathbf{A}^\top\mathbf{y}\geq\mathbf{1}}} \mathbf{1}^\top\mathbf{y}. \tag{2.2}$$

Packing and covering LPs, as well as their generalizations to packing and covering SDPs, have been well-studied in the literature [LN93, PST95, You01, DJ07, Nes08, JY11, AHK12, ALO16, PTZ16, AO19], with applications in areas such as combinatorial optimization, machine learning, and robust statistics [CMY20, JLT20, DKK+21].

[JLL+20] observed that by a standard binary search, solving (2.1) is equivalent to solving the following problem.

Problem 2.1 (ℓ_∞ packing linear program)**.** *Given* $\mathbf{A} \in \mathbb{R}^{d\times n}_{\geq 0}$*, find primal solution* $\mathbf{x} \in \Delta^n$ *with* $\|\mathbf{Ax}\|_\infty \leq 1$ *or dual solution* $\mathbf{y} \in \Delta^d$ *with* $\mathbf{A}^\top \mathbf{y} \geq \mathbf{1}$*.*

Intuitively, a primal solution is a convex combination of the columns of $\mathbf{A}$ so that no coordinate exceeds 1, while a dual solution is a convex combination of the rows of $\mathbf{A}$ so that every coordinate is at least 1. In many applications, we are interested in an approximate solution to Problem 2.1 in the following sense.

Problem 2.2 (Approximate ℓ_∞ packing linear program)**.** *Given* $\mathbf{A} \in \mathbb{R}^{d\times n}_{\geq 0}$ *and* $\epsilon \in [0, \frac{1}{2}]$*, find primal solution* $\mathbf{x} \in \Delta^n$ *with* $\|\mathbf{Ax}\|_\infty \leq 1+\epsilon$ *or dual solution* $\mathbf{y} \in \Delta^d$ *with* $\mathbf{A}^\top \mathbf{y} \geq (1-\epsilon)\mathbf{1}$*.*

[MRWZ16] gives a solver for Problem 2.2, which runs in time

$$O\left(\mathrm{nnz}(\mathbf{A}) \cdot \frac{\log(d)\log(nd/\epsilon)}{\epsilon^2}\right).$$

As analyzed by [JLT20], the [MRWZ16] algorithm can be interpreted as implementing approximate entropic mirror descent, which is the approach that we will take for our analysis.

A natural variant of Problem 2.2 considers the constraint $\|\mathbf{Ax}\| \leq 1+\epsilon$ for some other norm $\|\cdot\|$, where the dual set is adjusted accordingly. [JLT20] gives a modified version of the [MRWZ16] solver for ℓ_p norms, as well as a generalized solver for Schatten p-norm packing SDPs in the cases where p is an odd integer.

2.1.1 The Ky Fan k-norm

In this chapter, we work with the *Ky Fan k-norm*, which is defined as follows.

Definition 2.1 (Ky Fan k-norm)**.** For $\mathbf{x} \in \mathbb{R}^d$ with entries $\left|\mathbf{x}_{(1)}\right| \geq \left|\mathbf{x}_{(2)}\right| \geq \cdots \geq \left|\mathbf{x}_{(d)}\right|$ and $k \in [d]$, the *Ky-Fan k-norm* of $\mathbf{x}$ is defined as

$$\|\mathbf{x}\|_{(k)} := \sum_{i\in[k]} \left|\mathbf{x}_{(i)}\right|.$$

For $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $k \in [\min(n, d)]$, the Ky-Fan k-norm of $\mathbf{A}$ is defined as the Ky-Fan k-norm of the singular values of $\mathbf{A}$.

The Ky Fan k-norm is a well-studied and useful concept of independent interest, with applications such as matrix completion [CT10], robust statistics [WGR+09], and low-rank approximation [Wat93, DV22].

Remark 2.1. When $k = 1$ or $k = d$, we recover the familiar norms $\|\cdot\|_{(1)} = \|\cdot\|_\infty$ and $\|\cdot\|_{(d)} = \|\cdot\|_1$. The Ky Fan k-norm can thus be seen as an alternative to the ℓ_p norm as an interpolation between $\|\cdot\|_1$ and $\|\cdot\|_\infty$. Additionally, the Ky Fan k-norm of $\mathbf{x} \in \mathbb{R}^d$ can be computed in $O(d)$ time by using a linear-time selection algorithm to find the k^{th} largest absolute value and then summing the $|\mathbf{x}_i|$ above this threshold.

Throughout this chapter, we let $k \in [d]$ and $\|\cdot\|_*$ denote the dual of the Ky Fan k-norm. We establish a closed form for $\|\cdot\|_*$ through the following lemma.

Lemma 2.1. $\|\mathbf{x}\|_* = \max\left(\frac{\|\mathbf{x}\|_1}{k}, \|\mathbf{x}\|_\infty\right)$.

Proof. Let $\|\mathbf{x}\| = \max\left(\frac{\|\mathbf{x}\|_1}{k}, \|\mathbf{x}\|_\infty\right)$. Then the dual norm of $\|\cdot\|$ is

$$\max_{\|\mathbf{x}\|_1 \le k, \|\mathbf{x}\|_\infty \le 1} \langle \mathbf{x}, \cdot \rangle ,$$

which is clearly the Ky Fan k-norm, so the conclusion follows by the uniqueness of the dual norm (Fact A.12). □

Lemma 2.1 suggests that when extending Problem 2.2 to the Ky Fan k-norm, the dual set for $\mathbf{y}$ should be

$$\mathcal{Y} := \left\{\mathbf{y} \in \mathbb{R}^d_{\ge 0} \mid \|\mathbf{y}\|_1 = k, \|\mathbf{y}\|_\infty \le 1\right\}. \tag{2.3}$$

We can now define the following problem.

Problem 2.3 (Approximate Ky Fan packing linear program)**.** *Given* $\mathbf{A} \in \mathbb{R}^{d \times n}_{\ge 0}$ *and* $\epsilon \in [0, \frac{1}{2}]$*, find primal solution* $\mathbf{x} \in \Delta^n$ *with* $\|\mathbf{A}\mathbf{x}\|_{(k)} \le 1 + \epsilon$ *or dual solution* $\mathbf{y} \in \mathcal{Y}$ *with* $\mathbf{A}^\top \mathbf{y} \ge (1 - \epsilon)\mathbf{1}$.

2.2 Ky Fan packing algorithm

Throughout this section, we follow notation (2.3) and define the following regularizer, patterned off [DKK+21]:

$$\phi(\mathbf{y}) := \langle \mathbf{y}, \log \mathbf{y} \rangle - \|\mathbf{y}\|_1 ,$$

where log is taken entrywise. We first present several facts about ϕ.

Lemma 2.2. *ϕ is 1-strongly convex on $\mathcal{Y}$ with respect to $\|\cdot\|_*$.*

Proof. By Fact A.13, it suffices to show that $\nabla^2\phi(\mathbf{y})[\mathbf{x},\mathbf{x}] \geq \|\mathbf{x}\|_*^2$ for all $\mathbf{y} \in \mathcal{Y}$ and $\mathbf{x} \in \mathbb{R}^d$. Note that $\nabla^2\phi(\mathbf{y}) = \mathbf{diag}\,(\mathbf{y}^{-1})$, where inversion is entrywise.

Suppose $\|\mathbf{x}\|_* = \frac{\|\mathbf{x}\|_1}{k}$. Then

$$\nabla^2\phi(\mathbf{y})[\mathbf{x},\mathbf{x}] = \sum_{i\in[d]} \frac{\mathbf{x}_i^2}{\mathbf{y}_i} = \frac{1}{k}\left(\sum_{i\in[d]} \frac{\mathbf{x}_i^2}{\mathbf{y}_i}\right)\left(\sum_{i\in[d]} \mathbf{y}_i\right) \geq \frac{1}{k}\left(\sum_{i\in[d]} |\mathbf{x}_i|\right)^2 = \frac{\|\mathbf{x}\|_1^2}{k} \geq \|\mathbf{x}\|_*^2 ,$$

where the first inequality uses Titu's lemma (Fact A.16).

Otherwise, $\|\mathbf{x}\|_* = \|\mathbf{x}\|_\infty$. Then

$$\nabla^2\phi(\mathbf{y})[\mathbf{x},\mathbf{x}] = \sum_{i\in[d]} \frac{\mathbf{x}_i^2}{\mathbf{y}_i} \geq \sum_{i\in[d]} \mathbf{x}_i^2 \geq \|\mathbf{x}\|_\infty^2 = \|\mathbf{x}\|_*^2 ,$$

where the first inequality uses $\mathbf{y} \leq \mathbf{1}$ entrywise. □

Fact 2.1 (Lemma 7.3, [CMY20])**.** *For $\mathbf{x} \in \mathbb{R}^d$, let*

$$\tau(\mathbf{x}) := \max\left\{ \tau \;\middle|\; \tau > 0, \frac{\exp(\tau)}{\sum_{j\in[d]} \exp(\min(\tau, \mathbf{x}_j))} \leq \frac{1}{k} \right\}. \tag{2.4}$$

Then for all $i \in [d]$,

$$[\nabla\phi^*(\mathbf{x})]_i = \frac{k\exp(\min(\tau(\mathbf{x}), \mathbf{x}_i))}{\sum_{j\in[d]} \exp(\min(\tau(\mathbf{x}), \mathbf{x}_j))}, \tag{2.5}$$

where $\phi^*(\mathbf{x}) := \max_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{y}, \mathbf{x} \rangle - \phi(\mathbf{y})$ is the convex conjugate of ϕ.

Remark 2.2. For any $\mathbf{x} \in \mathbb{R}^d$, $\tau(\mathbf{x})$ can be efficiently computed by binary searching for the number of thresholded $\mathbf{x}_j$ in (2.4) in $O(\log(d))$ time and then solving for τ. Filling out $\nabla\phi^*(\mathbf{x})$ using (2.5) then takes $O(d)$ additional time.

Fact 2.2. *For $\mathbf{x} \in \mathbb{R}^d$, $\phi^*(\mathbf{x})$ is a $k\log(\frac{d}{k})$-additive approximation of $\|\mathbf{x}\|_{(k)}$, i.e.*

$$\|\mathbf{x}\|_{(k)} \leq \phi^*(\mathbf{x}) \leq \|\mathbf{x}\|_{(k)} + k\log\left(\frac{d}{k}\right).$$

We now present our algorithm for solving Problem 2.3.

Algorithm 1: KyFanPackingLP$(\mathbf{A}, \epsilon)$

Input: $\mathbf{A} \in \mathbb{R}^{d\times n}_{\geq 0}$, $\epsilon \in [0, \frac{1}{2}]$

$K \leftarrow \frac{3k\log(d)}{\epsilon}$, $\eta \leftarrow \frac{1}{2K}$, $T \leftarrow \frac{27k\log(d)\log(nd/\epsilon)}{\epsilon^2}$

$[\mathbf{w}_0]_i \leftarrow \frac{\epsilon}{n^2 d}$ for all $i \in [n]$, $\mathbf{z} \leftarrow \mathbf{0}$, $t \leftarrow 0$

while $\|\mathbf{A}\mathbf{w}_t\|_{(k)} \leq K$, $\|\mathbf{w}_t\|_1 \leq K$ **do**

  $\mathbf{v}_t \leftarrow \nabla\phi^*(\mathbf{A}\mathbf{w}_t)$

  $\mathbf{g}_t \leftarrow \max(0, \mathbf{1} - \mathbf{A}^\top\mathbf{v}_t)$ entrywise

  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t \circ (1 + \eta\mathbf{g}_t)$, $\mathbf{z} \leftarrow \mathbf{z} + \mathbf{v}_t$, $t \leftarrow t + 1$

  if $t \geq T$ **then**

    return $\mathbf{y} \leftarrow \frac{1}{T}\mathbf{z}$

return $\mathbf{x} \leftarrow \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|_1}$

Theorem 2.1. *Algorithm 1 solves Problem 2.3 in $O\left(\mathrm{nnz}(\mathbf{A}) \cdot \frac{k\log(d)\log(nd/\epsilon)}{\epsilon^2}\right)$ time.*

To prove Theorem 2.1, we follow the potential argument and mirror descent interpretation of [JLT20]. Here, we define the potential at time t to be

$$\Phi_t := \phi^*(\mathbf{A}\mathbf{w}_t) - \|\mathbf{w}_t\|_1.$$

We start by showing that the potential is monotonically nonincreasing.

Lemma 2.3. *For all $0 \leq t < T$, $\Phi_{t+1} \leq \Phi_t$.*

Proof. For any $0 \leq t < T$, let $\mathbf{x} := \mathbf{w}_t$, $\mathbf{x}' := \mathbf{w}_{t+1}$, and $\mathbf{g} := \mathbf{g}_t$. For $s \in [0, 1]$, define $\mathbf{x}_s := s\mathbf{x}' + (1 - s)\mathbf{x}$,

$$Q(s) := \nabla^2\phi^*(\mathbf{A}\mathbf{x}_s)[\mathbf{A}(\eta\mathbf{g} \circ \mathbf{x}), \mathbf{A}(\eta\mathbf{g} \circ \mathbf{x})],$$

and

$$R(s) := \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}_s) - \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle,$$

where $\mathbf{g}^2 := \mathbf{g} \circ \mathbf{g}$. By nonnegativity and $\mathbf{g} \le \mathbf{1}$ entrywise,

$$\begin{aligned}
\int_0^1 (1-s)R(s)\mathrm{d}s &= \int_0^1 (1-s) \int_0^s \nabla^2\phi^*(\mathbf{A}\mathbf{x}_u)[\mathbf{A}(\eta\mathbf{g} \circ \mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x})]\mathrm{d}u\mathrm{d}s \\
&\le \int_0^1 (1-s) \int_0^s Q(u)\mathrm{d}u\mathrm{d}s \\
&\le \int_0^1 \int_0^s Q(u)\mathrm{d}u\mathrm{d}s \\
&= \int_0^1 (1-s)Q(s)\mathrm{d}s,
\end{aligned} \tag{2.6}$$

where the fourth line used the Fubini–Tonelli theorem. Thus

$$\begin{aligned}
&\phi^*(\mathbf{A}\mathbf{x}') - \phi^*(\mathbf{A}\mathbf{x}) - \langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g} \circ \mathbf{x}) \rangle \\
&\qquad = \int_0^1 (1-s)Q(s)\mathrm{d}s \\
&\qquad \le \int_0^1 (1-s) \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}_s), (\mathbf{A}(\eta\mathbf{g} \circ \mathbf{x})) \circ (\mathbf{A}(\eta\mathbf{g} \circ \mathbf{x})) \right\rangle \mathrm{d}s \\
&\qquad \le \eta K \int_0^1 (1-s) \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}_s), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle \mathrm{d}s \\
&\qquad = \frac{\eta K}{2} \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle + \eta K \int_0^1 (1-s)R(s)\mathrm{d}s \\
&\qquad \le \frac{\eta K}{2} \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle + \eta K \int_0^1 (1-s)Q(s)\mathrm{d}s,
\end{aligned} \tag{2.7}$$

where the third line used

$$\nabla^2\phi^*(\mathbf{A}\mathbf{x}_s) \preceq \mathbf{diag}\left(\nabla\phi^*(\mathbf{A}\mathbf{x}_s)\right),$$

the fourth line used the Cauchy–Schwarz inequality and $\mathbf{A}\mathbf{x} \le K\mathbf{1}$ entrywise, and the sixth line used (2.6). Rearranging (2.7) and using $\eta K = \frac{1}{2}$ gives

$$\begin{aligned}
\phi^*(\mathbf{A}\mathbf{x}') - \phi^*(\mathbf{A}\mathbf{x}) - \langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g} \circ \mathbf{x}) \rangle &\le \frac{1}{2} \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle \\
&\le \left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}(\eta\mathbf{g}^2 \circ \mathbf{x}) \right\rangle.
\end{aligned}$$

Now, we can conclude

$$\begin{aligned}\Phi_{t+1} - \Phi_t &= \phi^*(\mathbf{A}\mathbf{x}') - \phi^*(\mathbf{A}\mathbf{x}) - \|\eta\mathbf{g} \circ \mathbf{x}\|_1 \\ &\leq \eta\left(\left\langle \nabla\phi^*(\mathbf{A}\mathbf{x}), \mathbf{A}((\mathbf{g}+\mathbf{g}^2)\circ\mathbf{x})\right\rangle - \langle \mathbf{g}, \mathbf{x}\rangle\right) \\ &= \eta\left\langle \mathbf{A}^\top\mathbf{v} \circ (\mathbf{g}+\mathbf{g}^2) - \mathbf{g}, \mathbf{x}\right\rangle \leq 0,\end{aligned}$$

where $\mathbf{v} := \nabla\phi^*(\mathbf{A}\mathbf{x})$ and the third line used Fact A.19 entrywise for $\mathbf{A}^\top\mathbf{v}$. □

Lemma 2.4. $\Phi_0 \leq 2k\log(d)$.

Proof. Since Φ_t is a $k\log(\frac{d}{k})$-additive approximation of $\|\mathbf{A}\mathbf{w}_t\|_{(k)} - \|\mathbf{w}_t\|_1$ by Fact 2.2 and the entries of $\mathbf{A}$ can be assumed to be bounded by $\frac{n}{\epsilon}$,[1]

$$\Phi_0 \leq \|\mathbf{A}\mathbf{w}_0\|_{(k)} - \|\mathbf{w}_0\|_1 + k\log\left(\frac{d}{k}\right) \leq \frac{k}{d} - \frac{\epsilon}{nd} + k\log\left(\frac{d}{k}\right) \leq 2k\log(d).$$

□

Lemma 2.5. *For all* $0 \leq t < T$, *if* $\Phi_t \leq 2k\log(d)$ *and* $\|\mathbf{A}\mathbf{w}_t\|_{(k)} > K$ *or* $\|\mathbf{w}_t\|_1 > K$, *then* $\mathbf{x} := \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|_1}$ *satisfies* $\|\mathbf{A}\mathbf{x}\|_{(k)} \leq 1+\epsilon$.

Proof. Let $0 \leq t < T$. By Fact 2.2, Lemma 2.3, and Lemma 2.4,

$$\|\mathbf{A}\mathbf{w}_t\|_{(k)} - \|\mathbf{w}_t\|_1 \leq \Phi_t \leq 2k\log(d). \tag{2.8}$$

Suppose $\|\mathbf{w}_t\|_1 > K$. Rearranging (2.8) and dividing by $\|\mathbf{w}_t\|_1$ gives

$$\|\mathbf{A}\mathbf{x}\|_{(k)} \leq \frac{\|\mathbf{w}_t\|_1 + 2k\log(d)}{\|\mathbf{w}_t\|_1} \leq 1 + \frac{2k\log(d)}{\|\mathbf{w}_t\|_1} \leq 1+\epsilon.$$

Otherwise, let $\|\mathbf{A}\mathbf{w}_t\|_{(k)} > K$. Since

$$\left(\frac{3}{\epsilon} - 2\right)k\log(d) = K - 2k\log(d) < \|\mathbf{A}\mathbf{w}_t\|_{(k)} - 2k\log(d) \leq \|\mathbf{w}_t\|_1,$$

rearranging (2.8) and dividing by $\|\mathbf{w}_t\|_1$ gives

$$\|\mathbf{A}\mathbf{x}\|_{(k)} \leq \frac{\|\mathbf{w}_t\|_1 + 2k\log(d)}{\|\mathbf{w}_t\|_1} \leq 1 + \frac{2k\log(d)}{\left(\frac{3}{\epsilon}-2\right)k\log(d)} \leq 1+\epsilon.$$

□

[1] cf. Lemma 16, [JLT20].

Lemma 2.6. *If* $\|\mathbf{w}_T\|_1 \le K$, *then* $\mathbf{y} := \frac{1}{T}\sum_{0\le t<T} \mathbf{v}_t \in \mathcal{Y}$ *satisfies* $\mathbf{A}^\top \mathbf{y} \ge (1-\epsilon)\mathbf{1}$.

Proof. Let $\mathbf{u} \in \Delta^n$. For $0 \le t < T$, where $\mathbf{x}_t := \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|_1} \in \Delta^n$,

$$\begin{aligned} D_{\mathrm{KL}}(\mathbf{x}_{t+1} \parallel \mathbf{u}) - D_{\mathrm{KL}}(\mathbf{x}_t \parallel \mathbf{u}) &= \sum_{i\in[n]} \mathbf{u}_i \log\left(\frac{[\mathbf{x}_t]_i}{[\mathbf{x}_{t+1}]_i}\right) \\ &= \sum_{i\in[n]} \mathbf{u}_i \left(\log\left(\frac{\|\mathbf{w}_{t+1}\|_1}{\|\mathbf{w}_t\|_1}\right) + \log\left(\frac{1}{1+\eta[\mathbf{g}_t]_i}\right)\right) \qquad (2.9) \\ &\le \log\left(\frac{\|\mathbf{w}_{t+1}\|_1}{\|\mathbf{w}_t\|_1}\right) - \eta(1-\eta)\left\langle \mathbf{g}_t, \mathbf{u}\right\rangle, \end{aligned}$$

where the inequality uses $\mathbf{g}_t \le \mathbf{1}$ entrywise and Fact A.20. Summing (2.9) across all T iterations, telescoping, and rearranging,

$$\begin{aligned} \eta(1-\eta)\sum_{0\le t<T} \left\langle \mathbf{g}_t, \mathbf{u}\right\rangle &\le \log\left(\frac{\|\mathbf{w}_T\|_1}{\|\mathbf{w}_0\|_1}\right) + D_{\mathrm{KL}}(\mathbf{x}_0 \parallel \mathbf{u}) - D_{\mathrm{KL}}(\mathbf{x}_T \parallel \mathbf{u}) \\ &\le \log\left(\frac{\|\mathbf{w}_T\|_1}{\|\mathbf{w}_0\|_1}\right) + D_{\mathrm{KL}}(\mathbf{x}_0 \parallel \mathbf{u}). \end{aligned}$$

Since $\|\mathbf{w}_0\|_1 = \frac{\epsilon}{nd}$, $\|\mathbf{w}_T\|_1 \le K$, $D_{\mathrm{KL}}(\mathbf{x}_0 \parallel \mathbf{u}) \le \log(n)$, and $\frac{1}{\eta(1-\eta)} \le 3K$,

$$\begin{aligned} \sum_{0\le t<T} \left\langle \mathbf{g}_t, \mathbf{u}\right\rangle &\le \frac{1}{\eta(1-\eta)}\left(\log\left(\frac{ndK}{\epsilon}\right) + \log(n)\right) \\ &= \frac{1}{\eta(1-\eta)} \log\left(\frac{n^2 dK}{\epsilon}\right) \le 3K \cdot 3\log\left(\frac{nd}{\epsilon}\right). \end{aligned}$$

Since $\mathbf{g}_t \ge \mathbf{1} - \mathbf{A}^\top \mathbf{v}_t$ entrywise for all iterations, we have

$$\left\langle \mathbf{1} - \mathbf{A}^\top \mathbf{y}, \mathbf{u}\right\rangle = \frac{1}{T}\sum_{0\le t<T} \left\langle \mathbf{1} - \mathbf{A}^\top \mathbf{v}_t, \mathbf{u}\right\rangle \le \frac{1}{T}\sum_{0\le t<T} \left\langle \mathbf{g}_t, \mathbf{u}\right\rangle \le \frac{9K\log(\frac{nd}{\epsilon})}{T} = \epsilon.$$

Taking $\mathbf{u} = \mathbf{e}_i$ for each $i \in [d]$ shows that $\mathbf{A}^\top \mathbf{y} \ge (1-\epsilon)\mathbf{1}$.

Finally, to show that $\mathbf{y} \in \mathcal{Y}$, note that each

$$\mathbf{v}_t = \nabla\phi^*(\mathbf{A}\mathbf{w}_t) \in \arg\max_{\mathbf{y}\in\mathcal{Y}} \left\langle \mathbf{y}, \mathbf{A}\mathbf{w}_t\right\rangle - \phi(\mathbf{y}) \subseteq \mathcal{Y}$$

by Fact A.15, so $\mathbf{y} \in \mathcal{Y}$ by convexity. □

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1. Correctness of primal feasibility follows from Lemma 2.5. Correctness of dual feasibility follows from Lemma 2.6. The runtime follows from Line 7, since each iteration cost is dominated by $\mathcal{T}_{\text{mv}}(\mathbf{A}) = O(\text{nnz}(\mathbf{A}))$. (Remarks 2.1 and 2.2 imply that in each iteration, everything else can be done in $O(d)$ time, and we work under the standard assumption that $\text{nnz}(\mathbf{A}) \geq \max(n, d)$.) □

2.2.1 Future work

We conjecture that it is sufficient to set

$$\eta = \Omega\left(\frac{k}{K}\right) \text{ and } T = O\left(\frac{\log(d)\log(\frac{nd}{\epsilon})}{\epsilon^2}\right)$$

in Algorithm 1, so that the resulting runtime of $O\left(\text{nnz}(\mathbf{A}) \cdot \frac{\log(d)\log(nd/\epsilon)}{\epsilon^2}\right)$ matches the runtime of the [MRWZ16] solver for Problem 2.2. The bottleneck of our analysis lies in the proof of Lemma 2.3, where our proof strategy requires $\eta K < 1$. We leave the resolution of this conjecture, as well as a generalization to Ky Fan packing SDPs, to future work.

2.3 Application to fair principal component analysis

Principal component analysis (PCA) is a widely used statistical technique for dimensionality reduction, data analysis, and pattern recognition [JC16]. The primary objective of PCA is to reduce the complexity of high-dimensional data while preserving its underlying structure. The general-purpose nature of PCA makes it particularly useful in applications ranging from machine learning [KL20] to finance [Mav22].

In its most basic form, the PCA problem (or 1-PCA problem) seeks to find

$$\arg\max_{\substack{\mathbf{v}\in\mathbb{R}^d \\ \|\mathbf{v}\|_2=1}} \left\langle \mathbf{v}\mathbf{v}^\top, \mathbf{A} \right\rangle \tag{2.10}$$

for a given $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$. It is a standard result that (2.10) is a unit eigenvector of $\mathbf{A}$ with largest eigenvalue. A natural generalization of (2.10), known as the k-PCA problem, seeks to find

$$\arg\max_{\substack{\mathbf{V}\in\mathbb{R}^{d\times k}\\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_k}} \left\langle \mathbf{V}\mathbf{V}^\top, \mathbf{A}\right\rangle \tag{2.11}$$

for a given $k \in [d]$. In many applications, $\mathbf{A} = \mathbf{X}^\top\mathbf{X}$ represents the covariance matrix of datapoints $\mathbf{X}$, where finding (2.11) intuitively translates to finding k orthogonal directions, or *principal components*, that capture the most variation in the data.

[JKL+24] showed that the following algorithm finds a solution to the k-PCA problem given an oracle for the 1-PCA problem. In fact, an approximate oracle for 1-PCA suffices to find an approximate solution to the k-PCA problem, for a suitable notion of approximation.

Algorithm 2: BlackBoxPCA$(\mathbf{A}, k, \mathcal{O})$

Input: $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, $k \in [d]$, an oracle $\mathcal{O}$ that takes $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ as input and returns a unit vector $\mathbf{v} \in \mathbb{R}^d$ in (2.10)

for $i \in [k]$ **do**
  $\mathbf{v}_i \leftarrow \mathcal{O}(\mathbf{A})$
  $\mathbf{A} \leftarrow (\mathbf{I} - \mathbf{v}_i\mathbf{v}_i^\top)\mathbf{A}(\mathbf{I} - \mathbf{v}_i\mathbf{v}_i^\top)$
return $\mathbf{V} \leftarrow \{\mathbf{v}_i\}_{i\in[k]} \in \mathbb{R}^{d\times k}$

Due to its nature as a fundamental statistical technique, the PCA problem has been studied, modified, and generalized in many ways, including robust PCA [CLMW11, JLT20], sparse PCA [Mac08, ZX18], and differentially private PCA [LKJO22]. In this section, we investigate a generalization called *fair PCA*. We mention that there are several distinct generalizations of PCA known as "fair PCA" in the literature [STM+18, KHFM22, KDRZ23]; we will work with the following definition.

Definition 2.2 (Fair k-PCA)**.** Let $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, and let $k \in [d]$. The *fair k-PCA* problem is to find a matrix in

$$\arg\max_{\substack{\mathbf{V}\in\mathbb{R}^{d\times k}\\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_k}} \min_{i\in[n]} \left\langle \mathbf{V}\mathbf{V}^\top, \mathbf{A}_i\right\rangle. \tag{2.12}$$

To see why this problem is called "fair" PCA, suppose there are m examples divided into n subgroups. Let $\mathbf{F}_i \in \mathbb{R}^{m_i \times d}$ be the feature matrix corresponding to the m_i examples in the i^{th} subgroup, and let $\overline{\mathbf{F}}_i := \mathbf{F}_i - \mathbf{1}_{m_i}\boldsymbol{\mu}_i^\top$ be the centered version of $\mathbf{F}_i$, where $\boldsymbol{\mu}_i = \frac{1}{m_i}\mathbf{F}_i^\top \mathbf{1}_{m_i}$ is the mean feature vector of the i^{th} subgroup. Then by setting

$$\mathbf{A}_i = \frac{1}{m_i}\overline{\mathbf{F}}_i^\top \overline{\mathbf{F}}_i \text{ for all } i \in [n],$$

(2.12) has the interpretation of being k orthogonal principal components that capture the most variation for *all* of the subgroups. This definition arises naturally and holds potential applications to fairness in machine learning.

Unfortunately, the natural generalization of Algorithm 2 does not compute approximate solutions to the fair k-PCA problem that are better than a constant factor, as shown in the following proposition.

Algorithm 3: BlackBoxFairPCA$(\{\mathbf{A}_i\}_{i\in[n]}, k, \mathcal{O})$

Input: $\mathbf{A}_1, \ldots, \mathbf{A}_n \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, $k \in [d]$, an oracle $\mathcal{O}$ that takes $\mathbf{A}_1, \ldots, \mathbf{A}_n \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ as input and returns a unit vector $\mathbf{v} \in \mathbb{R}^d$ in

$$\arg\max_{\substack{\mathbf{v}\in\mathbb{R}^d \\ \|\mathbf{v}\|_2=1}} \min_{i\in[n]} \left\langle \mathbf{v}\mathbf{v}^\top, \mathbf{A}_i \right\rangle$$

```
for i ∈ [k] do
    v_i ← O({A_i}_{i∈[n]})
    for j ∈ [n] do
        A_j ← (I − v_i v_i^⊤) A_j (I − v_i v_i^⊤)
return V ← {v_i}_{i∈[k]} ∈ R^{d×k}
```

Proposition 2.1. *For $n > 1$, $k > 1$, and $d > k$, Algorithm 3 is at most a $\frac{1}{2}$-approximation algorithm for fair k-PCA. Even in the smallest nontrivial case of $n = 2$, $k = 2$, and $d = 3$, Algorithm 3 is at most a $\frac{3}{4}$-approximation algorithm for fair k-PCA.*

Proof. We first show that Algorithm 3 is at most a $\frac{3}{4}$-approximation algorithm for fair k-PCA in the case $n = 2$, $k = 2$, and $d = 3$ by giving an explicit instance.

Let $\epsilon > 0$, $\mathbf{A}_1 = \mathbf{diag}(1, 0, 2+\epsilon)$, and $\mathbf{A}_2 = \mathbf{diag}(0, 1, 1+\epsilon)$. It is clear that $\mathbf{v}_1 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$. Projecting out the $\mathbf{v}_1$ direction yields the updates

$$\mathbf{A}_1 \leftarrow \mathbf{diag}(1, 0, 0) \quad \text{and} \quad \mathbf{A}_2 \leftarrow \mathbf{diag}(0, 1, 0).$$

Now $\mathbf{v}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}^\top$, so

$$\mathbf{V} = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \\ 1 & 0 \end{pmatrix},$$

which gives $\min_{i\in[n]} \left\langle \mathbf{V}\mathbf{V}^\top, \mathbf{A}_i \right\rangle = \min(\frac{5}{2}+\epsilon, \frac{3}{2}+\epsilon) = \frac{3}{2}+\epsilon$. However,

$$\mathbf{U} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

gives $\min_{i\in[n]} \left\langle \mathbf{U}\mathbf{U}^\top, \mathbf{A}_i \right\rangle = \min(2+\epsilon, 2+\epsilon) = 2+\epsilon$, so the approximation ratio of Algorithm 3 is at most $\frac{3}{4}$ in this case.

Now we extend the above instance to $n \geq 2$, $k = 2$, and $d = n+1$. For $\epsilon > 0$ and $i \in [n-1]$, let $\mathbf{A}_i = \mathbf{diag}(\delta_{1i}, \delta_{2i}, \ldots, \delta_{ni}, 2+\epsilon)$, where δ_{ij} is the Kronecker delta, and let $\mathbf{A}_n = \mathbf{diag}(0, \ldots, 0, 1, 1+\epsilon)$. Then Algorithm 3 outputs

$$\mathbf{V} = \begin{pmatrix} 0 & \frac{1}{\sqrt{n}} \\ \vdots & \vdots \\ 0 & \frac{1}{\sqrt{n}} \\ 1 & 0 \end{pmatrix},$$

which gives an objective value of $1 + \frac{1}{n} + \epsilon$, but

$$\mathbf{U} = \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

achieves an objective value of $2+\epsilon$. Taking $n \to \infty$ shows that the approximation ratio of Algorithm 3 is at most $\frac{1}{2}$. □

2.3.1 SDP formulation of fair PCA

Although Proposition 2.1 suggests that fair k-PCA cannot be solved using a fair 1-PCA oracle, we can formulate an SDP relaxation of (2.12). We first recall the following fact from convex analysis.

Fact 2.3. *For any linear function $f : \mathbb{R}^d \to \mathbb{R}$ and set $S \subseteq \mathbb{R}^d$,*

$$\min_{\mathbf{x}\in S} f(\mathbf{x}) = \min_{\mathbf{x}\in \mathrm{conv}(S)} f(\mathbf{x}).$$

We note that $\Delta^n = \mathrm{conv}(\{\mathbf{e}_i \mid i \in [n]\})$ and

$$\begin{aligned}\mathcal{F}_k^{d\times d} &:= \left\{\mathbf{Y} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}} \mid \mathrm{Tr}(\mathbf{Y}) = k, \mathbf{Y} \preceq \mathbf{I}_d\right\} \\ &= \mathrm{conv}\left(\left\{\mathbf{V}\mathbf{V}^\top \mid \mathbf{V} \in \mathbb{R}^{d\times k}, \mathbf{V}^\top\mathbf{V} = \mathbf{I}_k\right\}\right),\end{aligned}$$

where $\mathcal{F}_k^{d\times d}$ is known as the *k-Fantope*. Thus by Fact 2.3,

$$\begin{aligned}\max_{\substack{\mathbf{V}\in\mathbb{R}^{d\times k}\\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_k}} \min_{i\in[n]} \left\langle \mathbf{V}\mathbf{V}^\top, \mathbf{A}_i\right\rangle &= \max_{\substack{\mathbf{V}\in\mathbb{R}^{d\times k}\\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_k}} \min_{i\in[n]} \left\langle \mathbf{V}\mathbf{V}^\top, \sum_{j\in[n]} [\mathbf{e}_i]_j \mathbf{A}_j \right\rangle \\ &= \max_{\substack{\mathbf{V}\in\mathbb{R}^{d\times k}\\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_k}} \min_{\mathbf{w}\in\Delta^n} \left\langle \mathbf{V}\mathbf{V}^\top, \sum_{j\in[n]} \mathbf{w}_j \mathbf{A}_j \right\rangle \\ &\leq \max_{\mathbf{Y}\in\mathcal{F}_k^{d\times d}} \min_{\mathbf{w}\in\Delta^n} \left\langle \mathbf{Y}, \sum_{j\in[n]} \mathbf{w}_j \mathbf{A}_j \right\rangle,\end{aligned}$$

which can now be solved using a Ky Fan packing SDP solver.[2]

[2] As mentioned in Section 2.2.1, we do not provide a Ky Fan packing SDP solver in this work.

Chapter 3

Fast Forster Transforms

This chapter is based on [JLT25], with Arun Jambulapati and Kevin Tian.

3.1 Radial isotropic position

Transforming a dataset $A = \{\mathbf{a}_i\}_{i\in[n]} \subset \mathbb{R}^d$ into a canonical representation enjoying a greater deal of regularity is a powerful idea that has had myriad applications throughout computer science, statistics, and related fields. Examples of common such representations include the following.

- **Normalization:** Replacing each $\mathbf{a}_i$ with the unit vector $\tilde{\mathbf{a}}_i := \mathbf{a}_i \left\|\mathbf{a}_i\right\|_2^{-1}$ in the same direction. Such a transformation exists whenever all of the $\{\mathbf{a}_i\}_{i\in[n]}$ are nonzero vectors.

- **Isotropic position:** Replacing each $\mathbf{a}_i$ with $\tilde{\mathbf{a}}_i := \mathbf{R}\mathbf{a}_i$ for an invertible $\mathbf{R} \in \mathbb{R}^{d\times d}$, such that $\sum_{i\in[n]} \tilde{\mathbf{a}}_i\tilde{\mathbf{a}}_i^\top = \mathbf{I}_d$. Such a transformation exists whenever the $\{\mathbf{a}_i\}_{i\in[n]}$ span $\mathbb{R}^d$.

Recently, a common generalization of both of these representations known as *radial isotropic position* has emerged as a desirable data processing step in many settings. Although the concept of radial isotropic position first arose in early work on

algebraic geometry [GGMS87] and functional analysis [Bar98], it has since enabled many surprising results in algorithms and complexity. For example, radial isotropic position played a pivotal role in breakthroughs spanning disparate areas such as communication complexity [For02], subspace recovery [HM13], coding theory [DSW14], frame theory [HM19], active and noisy learning of halfspaces [HKLM20, DKT21, DTK23], and robust statistics [Che24].

We now formally define the concept of radial isotropic position.

Definition 3.1 (Radial isotropic position)**.** Let $\mathbf{c} \in (0,1]^n$ satisfy $\|\mathbf{c}\|_1 = d$, and let $\epsilon \in (0,1)$. We say that $\mathbf{A} \in \mathbb{R}^{n \times d}$ with rows $\{\mathbf{a}_i\}_{i \in [n]}$ is in $(\mathbf{c}, \epsilon)$-*radial isotropic position* (or, $(\mathbf{c}, \epsilon)$-*RIP*) if

$$\exp(-\epsilon)\mathbf{I}_d \preceq \sum_{i \in [n]} \mathbf{c}_i \cdot \frac{\mathbf{a}_i \mathbf{a}_i^\top}{\|\mathbf{a}_i\|_2^2} \preceq \exp(\epsilon)\mathbf{I}_d. \tag{3.1}$$

If ϵ is omitted then $\epsilon = 0$ by default, and if $\mathbf{c}$ is omitted then $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$ by default. For an invertible matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$, we say that $\mathbf{R}$ *is a* $(\mathbf{c}, \epsilon)$-*Forster transform of* $\mathbf{A}$ if $\mathbf{A}\mathbf{R}^\top$ is in $(\mathbf{c}, \epsilon)$-RIP:

$$\exp(-\epsilon)\mathbf{I}_d \preceq \sum_{i \in [n]} \mathbf{c}_i \cdot \frac{(\mathbf{R}\mathbf{a}_i)(\mathbf{R}\mathbf{a}_i)^\top}{\|\mathbf{R}\mathbf{a}_i\|_2^2} \preceq \exp(\epsilon)\mathbf{I}_d. \tag{3.2}$$

In other words, $\mathbf{R}$ is a $\mathbf{c}$-Forster transform of $\mathbf{A} \in \mathbb{R}^{n \times d}$ representing the dataset $A = \{\mathbf{a}_i\}_{i \in [n]} \subset \mathbb{R}^d$ if the transformed-and-normalized vectors

$$\{(\mathbf{R}\mathbf{a}_i)\|\mathbf{R}\mathbf{a}_i\|_2^{-1}\}_{i \in [n]}$$

are in isotropic position. Note that $\|\mathbf{c}\|_1 = d$ in Definition 3.1 is necessary as $\epsilon \to 0$, by taking traces of (3.1). For example, $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$ induces an empirical second moment matrix with uniform weights. After applying a Forster transform, the new dataset then exhibits desirable properties that are useful in downstream applications.

To briefly demystify Definition 3.1, let $\mathbf{A} \in \mathbb{R}^{n \times d}$ with rows $\{\mathbf{a}_i\}_{i \in [n]}$ have $\mathrm{rank}(\mathbf{A}) = d$ (so $n \geq d$). Then, it is well-known that $\mathbf{A}$ can be scaled by invertible

$\mathbf{R} \in \mathbb{R}^{d \times d}$ so that

$$\mathbf{R}\mathbf{A}^\top \mathbf{diag}\left(\mathbf{c}\right) \mathbf{A}\mathbf{R}^\top = \sum_{i \in [n]} \mathbf{c}_i \left(\mathbf{R}\mathbf{a}_i\right) \left(\mathbf{R}\mathbf{a}_i\right)^\top = \mathbf{I}_d. \tag{3.3}$$

Indeed, choosing $\mathbf{R} = \left(\mathbf{A}^\top \mathbf{diag}\left(\mathbf{c}\right) \mathbf{A}\right)^{-\frac{1}{2}}$ suffices. The condition (3.3) is sometimes referred to as being scaled to be in $\mathbf{c}$-isotropic position, and there are natural ϵ-approximate generalizations.

Similarly, as long as all $\mathbf{a}_i \neq \mathbf{0}_d$, there is a diagonal scaling $\mathbf{S}$ so that $\mathbf{S}\mathbf{A}$ has unit-norm rows: let

$$\mathbf{S} = \mathbf{diag}\left(\mathbf{s}\right) \text{ where } \mathbf{s}_i = \frac{1}{\left\|\mathbf{a}_i\right\|_2} \text{ for all } i \in [n] \implies \left\|\left[\mathbf{S}\mathbf{A}\right]_{i:}\right\|_2 = 1 \text{ for all } i \in [n]. \tag{3.4}$$

Each of the transformations (3.3) and (3.4) is used in many applications to improve the regularity of a point set given by viewing the rows of $\mathbf{A}$ as points in $\mathbb{R}^d$. The purpose of $\mathbf{c}$-radial isotropic position (Definition 3.1) is to give a Forster transform matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$ inducing a scaling $\mathbf{S} = \mathbf{diag}\left(\mathbf{s}\right)$ via $\mathbf{s}_i^{-1} = \left\|\mathbf{R}\mathbf{a}_i\right\|_2$ for all $i \in [n]$, such that the left-and-right scaled matrix $\mathbf{S}\mathbf{A}\mathbf{R}^\top$ simultaneously has unit-norm rows, and is in $\mathbf{c}$-isotropic position. Our goal is to efficiently approximate $\mathbf{R}$.

3.1.1 Equivalent characterizations

Not all point sets admit a $\mathbf{c}$-Forster transform. Many of the aforementioned results and applications [GGMS87, Bar98, For02, CLL04, HM13, DSW14, HKLM20, DKT21, DTK23] leverage certain necessary and sufficient conditions for the existence of a $\mathbf{c}$-Forster transform instead of using Definition 3.1 directly. We mention several of these conditions that are relevant to our development here; a more extensive survey can be found in [AKS20].

First, [GGMS87, Bar98] (see also [CLL04, HKLM20]) show that a $\mathbf{c}$-Forster transform of $\mathbf{A}$ exists if and only if $\mathbf{c}$ belongs to the *basis polytope* of the independence matroid induced by A. A more intuitive and equivalent way of phrasing this result is given by the following proposition.

Proposition 3.1 (Lemma 4.19, [HKLM20]). *Given a point set $A := \{\mathbf{a}_i\}_{i\in[n]} \subset \mathbb{R}^d$ and $\mathbf{c} \in (0,1]^n$ satisfying $\|\mathbf{c}\|_1 = d$, the following conditions are equivalent, where $\mathbf{A} \in \mathbb{R}^{n\times d}$ has rows A.*

1. *For any $\epsilon > 0$ there exists $\mathbf{R} \in \mathbb{R}^{d\times d}$, a $(\mathbf{c}, \epsilon)$-Forster transform of $\mathbf{A}$.*
2. *For every $k \in [d]$, every k-dimensional linear subspace $V \subseteq \mathbb{R}^d$ satisfies*

$$\sum_{\substack{i\in[n]\\ \mathbf{a}_i\in V}} \mathbf{c}_i \le k. \tag{3.5}$$

One direction of Proposition 3.1 is straightforward: if a k-dimensional subspace V is too "heavy" (i.e., (3.5) is violated) then there still exists a heavy subspace under any transform $\mathbf{R}$. Taking the trace of both sides of the definition (3.2) restricted to this heavy subspace yields a contradiction for sufficiently small ϵ. In the case of $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$, (3.5) simply translates to no k-dimensional subspace containing more than $\frac{k}{d} \cdot n$ of the points. One simple way for this condition to hold is if A is in *general position*. Note also that by taking $V = \mathrm{Span}(A)$, (3.5) implies that $n \ge d$ and $\mathbf{A}$ has full rank.

The other direction is significantly more challenging, and [HKLM20] gives an iterative construction based on decompositions with respect to the *basis polytope* of $A = \{\mathbf{a}_i\}_{i\in[n]}$, which we define here.

Definition 3.2 (Basis polytope). Consider a point set $A = \{\mathbf{a}_i\}_{i\in[n]}$. Let $\mathcal{B} \subseteq 2^{[n]}$ be the set of subsets $B \subseteq [n]$ such that $\{\mathbf{a}_i\}_{i\in B}$ is a basis of $\mathbb{R}^d$, i.e., it is a linearly-independent set that spans $\mathbb{R}^d$. Letting $\mathbf{1}_B \in \{0,1\}^n$ denote the 0-1 indicator vector of each $B \in \mathcal{B}$, we let

$$\mathcal{P}(A) := \mathrm{conv}\left(\{\mathbf{1}_B\}_{B\in\mathcal{B}}\right)$$

denote the *basis polytope* corresponding to the independent set matroid induced by A.

The following result of [CLL04] shows that Proposition 3.1 is an equivalent formulation of the basis polytope characterization of radial isotropic position; see also [Bar98, DSW14, HKLM20] for interpretations of this condition.

Proposition 3.2 (Theorem 4.4, [CLL04])**.** *The conditions in Proposition 3.1 hold iff* $\mathbf{c} \in \mathcal{P}(A)$.

Another dual viewpoint on Forster transforms is from the perspective of scaling the dataset to induce certain *leverage scores*, which are defined as follows.

Definition 3.3 (Leverage scores)**.** Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ have rows $\{\mathbf{a}_i\}_{i \in [n]} \subset \mathbb{R}^d$. The *leverage scores* $\boldsymbol{\tau}$ of $\mathbf{A}$ are given by

$$\boldsymbol{\tau}_i(\mathbf{A}) := \mathbf{a}_i^\top \left(\mathbf{A}^\top \mathbf{A}\right)^\dagger \mathbf{a}_i. \tag{3.6}$$

As described in our derivation (3.3), (3.4), $\mathbf{c}$-RIP can equivalently be viewed as being induced by a pair $(\mathbf{R}, \mathbf{s})$, where the diagonal scaling $\mathbf{s}$ is an implicit function of the Forster transform $\mathbf{R}$.

We may ask if this correspondence goes the other direction; are there conditions on $\mathbf{s} \in \mathbb{R}^n_{>0}$ such that one can deduce $\mathbf{R} \in \mathbb{R}^{d \times d}$ that scales $\mathbf{A}$ to be in $\mathbf{c}$-RIP? The following observation, patterned from [DR24], shows the answer is yes: finding $\mathbf{s} \in \mathbb{R}^n_{>0}$ such that

$$\boldsymbol{\tau}\left(\mathbf{SA}\right) = \mathbf{c}, \text{ where } \mathbf{S} := \mathbf{diag}\left(\mathbf{s}\right), \tag{3.7}$$

implies that $\mathbf{R} = (\mathbf{A}^\top \mathbf{S}^2 \mathbf{A})^{-\frac{1}{2}}$ is a $\mathbf{c}$-Forster transform of $\mathbf{A}$.

Lemma 3.1. *Given a point set* $A := \{\mathbf{a}_i\}_{i \in [n]} \subset \mathbb{R}^d$ *and* $\mathbf{c} \in (0, 1]^n$ *satisfying* $\|\mathbf{c}\|_1 = d$*, suppose* (3.5) *holds. Then letting* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *have rows* A*, if some* $\mathbf{s} \in \mathbb{R}^n_{>0}$ *satisfies* $\boldsymbol{\tau}(\mathbf{SA}) = \mathbf{c}$*, where* $\mathbf{S} := \mathbf{diag}\left(\mathbf{s}\right)$*, then*

$$\sum_{i \in [n]} \mathbf{c}_i \cdot \frac{\left(\mathbf{R}\mathbf{a}_i\right)\left(\mathbf{R}\mathbf{a}_i\right)^\top}{\|\mathbf{R}\mathbf{a}_i\|_2^2} = \mathbf{I}_d \text{ for } \mathbf{R} := \left(\mathbf{A}^\top \mathbf{S}^2 \mathbf{A}\right)^{-\frac{1}{2}}. \tag{3.8}$$

More generally, for any $\epsilon > 0$*, if* $\boldsymbol{\tau}(\mathbf{SA}) \approx_\epsilon \mathbf{c}$*, then*

$$\sum_{i\in[n]} \mathbf{c}_i \cdot \frac{(\mathbf{Ra}_i)(\mathbf{Ra}_i)^\top}{\|\mathbf{Ra}_i\|_2^2} \approx_\epsilon \mathbf{I}_d \text{ for } \mathbf{R} := \left(\mathbf{A}^\top \mathbf{S}^2 \mathbf{A}\right)^{-\frac{1}{2}}. \tag{3.9}$$

Proof. We prove (3.9), which implies (3.8) by taking $\epsilon \to 0$. Indeed, by using $\boldsymbol{\tau}(\mathbf{SA}) \approx_\epsilon \mathbf{c}$,

$$\boldsymbol{\tau}_i(\mathbf{SA}) = \mathbf{s}_i^2 \|\mathbf{Ra}_i\|_2^2 \approx_\epsilon \mathbf{c}_i \implies \frac{\mathbf{c}_i}{\|\mathbf{Ra}_i\|_2^2} \approx_\epsilon \mathbf{s}_i^2 \text{ for all } i \in [n].$$

This directly implies that (3.9) holds:

$$\sum_{i\in[n]} \mathbf{c}_i \cdot \frac{(\mathbf{Ra}_i)(\mathbf{Ra}_i)^\top}{\|\mathbf{Ra}_i\|_2^2} \approx_\epsilon \sum_{i\in[n]} \mathbf{s}_i^2 (\mathbf{Ra}_i)(\mathbf{Ra}_i)^\top = \mathbf{R}\left(\sum_{i\in[n]} \mathbf{s}_i^2 \mathbf{a}_i \mathbf{a}_i^\top\right)\mathbf{R} = \mathbf{I}_d.$$

□

Thus, while Forster transforms are *right scalings* $\mathbf{R} \in \mathbb{R}^{d\times d}$ putting $\mathbf{A}$ in isotropic position, we can equivalently find a *left scaling* $\mathbf{s} \in \mathbb{R}^n_{>0}$ that balances $\mathbf{A}$'s rows to have target leverage scores $\mathbf{c}$ of our choice.

The final characterization that we discuss here was given in the seminal work [Bar98]. We fix a point set $A := \{\mathbf{a}_i\}_{i\in[n]} \subset \mathbb{R}^d$ that forms the rows of $\mathbf{A} \in \mathbb{R}^{n\times d}$. We also let $\mathbf{c} \in (0,1]^n$ satisfy $\|\mathbf{c}\|_1 = d$, such that the condition (3.5) holds. For some $\epsilon \in (0,1)$, we will use Barthe's characterization to give an algorithm for computing a $(\mathbf{c},\epsilon)$-Forster transform of $\mathbf{A}$.

To ease our exposition we fix the following notation throughout, for $\mathbf{t} \in \mathbb{R}^n$:

$$\begin{aligned}
\mathbf{Z}(\mathbf{t}) &:= \mathbf{A}^\top \mathbf{diag}(\exp(\mathbf{t}))\,\mathbf{A} = \sum_{i\in[n]} \exp(\mathbf{t}_i)\,\mathbf{a}_i\mathbf{a}_i^\top, \\
\mathbf{R}(\mathbf{t}) &:= \mathbf{Z}(\mathbf{t})^{-\frac{1}{2}} = \left(\mathbf{A}^\top \mathbf{diag}(\exp(\mathbf{t}))\,\mathbf{A}\right)^{-\frac{1}{2}}, \\
\mathbf{S}(\mathbf{t}) &:= \mathbf{diag}(\mathbf{s}(\mathbf{t})), \text{ where } \mathbf{s}(\mathbf{t}) := \exp\left(\frac{\mathbf{t}}{2}\right), \\
\tilde{\mathbf{a}}_i(\mathbf{t}) &:= \mathbf{R}(\mathbf{t})\,\mathbf{a}_i, \text{ for all } i \in [n].
\end{aligned} \tag{3.10}$$

In (3.10), we let exp be applied to a vector argument entrywise. Note that all of the matrices and vectors in (3.10) correspond to those arising in our earlier discussion, after reparameterizing the problem by $\mathbf{t} = 2\log(\mathbf{s})$ entrywise. This reparameterization becomes convenient shortly.

[Bar98] gives an algorithmic proof of Propositions 3.1 and 3.2 by explicitly characterizing the scaling $\mathbf{s} \in \mathbb{R}^n_{>0}$ in Lemma 3.1 such that $\boldsymbol{\tau}(\mathbf{SA}) = \mathbf{c}$ for $\mathbf{S} := \mathbf{diag}(\mathbf{s})$, by way of a $\mathbf{t} \in \mathbb{R}^n$ that achieves $\mathbf{s} = \mathbf{s}(\mathbf{t})$. To explain, we first define Barthe's objective:

$$f(\mathbf{t}) := -\langle \mathbf{c}, \mathbf{t} \rangle + \log\det(\mathbf{Z}(\mathbf{t})). \tag{3.11}$$

Then Barthe's result can be stated as follows.

Proposition 3.3 (Proposition 6, [Bar98])**.** *Following notation* (3.10), (3.11), $f : \mathbb{R}^n \to \mathbb{R}$ *is a convex function, and its minimizer is attained iff* $\mathbf{A}$, $\mathbf{c}$ *satisfy* $\|\mathbf{c}\|_1 = d$ *and the condition* (3.5)*. Moreover, letting* $\mathbf{t}^\star := \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$, $\mathbf{R}(\mathbf{t}^\star)$ *is a* $\mathbf{c}$*-Forster transform of* $\mathbf{A}$*.*

Proposition 3.3 can be somewhat demystified by computing the derivatives of Barthe's objective. We introduce one additional piece of notation here:

$$\mathbf{M}_i(\mathbf{t}) := \mathbf{s}_i(\mathbf{t})^2 \tilde{\mathbf{a}}_i(\mathbf{t}) \tilde{\mathbf{a}}_i(\mathbf{t})^\top = \mathbf{R}(\mathbf{t}) \left(\exp(\mathbf{t}_i)\, \mathbf{a}_i \mathbf{a}_i^\top\right) \mathbf{R}(\mathbf{t}), \text{ for all } i \in [n]. \tag{3.12}$$

Fact 3.1. *Following notation* (3.10), (3.11), *we have for all* $(i,j) \in [n] \times [n]$ *that*

$$\begin{aligned} \nabla_i f(\mathbf{t}) &= -\mathbf{c}_i + \mathrm{Tr}(\mathbf{M}_i(\mathbf{t})), \\ \nabla^2_{ij} f(\mathbf{t}) &= \mathrm{Tr}(\mathbf{M}_i(\mathbf{t}))\, \mathbb{I}_{i=j} - \mathrm{Tr}(\mathbf{M}_i(\mathbf{t})\mathbf{M}_j(\mathbf{t})). \end{aligned} \tag{3.13}$$

From Fact 3.1 we can glean several different parts of Proposition 3.3. For example, the fact that $\sum_{i\in[n]} \mathbf{M}_i(\mathbf{t}) = \mathbf{R}(\mathbf{t})\mathbf{Z}(\mathbf{t})\mathbf{R}(\mathbf{t}) = \mathbf{I}_d$, combined with Kadison's inequality [Kad52] (see also Theorem 2.3.2, [Bha07]), shows that for all vectors $\mathbf{v} \in \mathbb{R}^d$,

$$\left(\sum_{i\in[n]} \mathbf{v}_i \mathbf{M}_i(\mathbf{t})\right)^2 \preceq \sum_{i\in[n]} \mathbf{v}_i^2 \mathbf{M}_i(\mathbf{t}).$$

In particular, taking a trace of both sides above shows

$$\nabla^2 f(\mathbf{t})[\mathbf{v},\mathbf{v}] = \operatorname{Tr}\left(\sum_{i\in[n]} \mathbf{v}_i^2 \mathbf{M}_i(\mathbf{t})\right) - \operatorname{Tr}\left(\left(\sum_{i\in[n]} \mathbf{v}_i \mathbf{M}_i(\mathbf{t})\right)^2\right) \geq 0,$$

which implies that f is convex. Similarly, letting $\mathbf{t}^\star$ minimize f, we have from (3.13) that

$$\mathbf{c}_i = \operatorname{Tr}\left(\mathbf{M}_i(\mathbf{t}^\star)\right) = \boldsymbol{\tau}_i\left(\mathbf{S}\left(\mathbf{t}^\star\right)\mathbf{A}\right) \text{ for all } i \in [n]. \tag{3.14}$$

Using the characterization of $\mathbf{S}(\mathbf{t}^\star)$ in (3.14) as obtaining the leverage scores $\mathbf{c}$, and applying Lemma 3.1, we have shown that $\mathbf{R}(\mathbf{t}^\star)$ is indeed a $\mathbf{c}$-Forster transform of $\mathbf{A}$, as stated in Proposition 3.3.

For completeness, we sketch a proof of the other direction of Proposition 3.3. By Proposition 3.2, it suffices to show that if $\mathbf{c}$ is in the basis polytope of the rows of $\mathbf{A}$, then f attains a finite infimum.

Let $\mathcal{B} \subseteq 2^{[n]}$ be the set of subsets $B \subseteq [n]$ such that $\{\mathbf{a}_i\}_{i\in B}$ is a basis of $\mathbb{R}^d$. Since $\mathbf{c}$ is in the basis polytope, there exist $\{\mathbf{w}_B\}_{B\in\mathcal{B}} \subset [0,1]$ such that

$$\sum_{B\in\mathcal{B}} \mathbf{w}_B = 1 \text{ and } \mathbf{c} = \sum_{B\in\mathcal{B}} \mathbf{w}_B \mathbf{1}_B.$$

For all $S \subseteq [n]$ such that $|S| = d$ and $S \notin \mathcal{B}$, we let $\mathbf{w}_S = 0$. For all $\mathbf{t} \in \mathbb{R}^n$, by the Cauchy–Binet formula (Fact A.6),

$$\begin{aligned}
\det(\mathbf{Z}(\mathbf{t})) &= \sum_{\substack{S\subseteq[n]\\|S|=d}} \exp\left(\sum_{i\in S} \mathbf{t}_i\right) \det(\mathbf{A}_{S:})^2 \\
&= \sum_{\substack{S\subseteq[n]\\|S|=d\\ \mathbf{w}_S\neq 0}} \mathbf{w}_S \exp\left(\sum_{i\in S} \mathbf{t}_i\right) \frac{\det(\mathbf{A}_{S:})^2}{\mathbf{w}_S} + \sum_{\substack{S\subseteq[n]\\|S|=d\\ \mathbf{w}_S=0}} \exp\left(\sum_{i\in S} \mathbf{t}_i\right) \det(\mathbf{A}_{S:})^2 \\
&\geq \sum_{\substack{S\in\mathcal{B}\\ \mathbf{w}_S\neq 0}} \mathbf{w}_S \exp\left(\sum_{i\in S} \mathbf{t}_i\right) \frac{\det(\mathbf{A}_{S:})^2}{\mathbf{w}_S} \\
&\geq \prod_{\substack{S\in\mathcal{B}\\ \mathbf{w}_S\neq 0}} \left(\exp\left(\sum_{i\in S} \mathbf{t}_i\right) \frac{\det(\mathbf{A}_{S:})^2}{\mathbf{w}_S}\right)^{\mathbf{w}_S} = \exp(\langle \mathbf{c}, \mathbf{t}\rangle) \prod_{\substack{S\in\mathcal{B}\\ \mathbf{w}_S\neq 0}} \left(\frac{\det(\mathbf{A}_{S:})^2}{\mathbf{w}_S}\right)^{\mathbf{w}_S},
\end{aligned}$$

where the fourth line uses Fact A.17. Since $\mathbf{w}_S \neq 0$ implies $\det(\mathbf{A}_{S:})^2 > 0$, it follows that

$$f(\mathbf{t}) \geq \sum_{\substack{S \in \mathcal{B} \\ \mathbf{w}_S \neq 0}} \mathbf{w}_S \log \left(\frac{\det(\mathbf{A}_{S:})^2}{\mathbf{w}_S} \right) > -\infty$$

for all $\mathbf{t} \in \mathbb{R}^n$.

3.2 Prior and related work

The goal of our work is designing efficient algorithms for computing a $(\mathbf{c}, \epsilon)$-Forster transform of $\mathbf{A} \in \mathbb{R}^{n \times d}$, whenever one exists. This goal is inspired by advancements in the complexity of simpler, but related, dataset transformation problems called *matrix balancing and scaling*, for which [CMTV17, ALdOW17] achieved nearly-linear runtimes in well-conditioned regimes. Indeed, as the list of applications of radial isotropic position grows, so too does the importance of designing efficient algorithms for finding them.

Given the algorithmic significance of Forster transforms, it is perhaps surprising that investigations of their computational complexity are relatively nascent. Previous strategies for obtaining polynomial-time algorithms can largely be grouped under two categories, optimizing Barthe's objective and iterative scaling methods.

The first approach was followed by [HM13] (see also discussion in [HM19, Che24]), who proceeded via cutting-plane methods (CPMs), and [AKS20], who used first-order methods (e.g., gradient descent). However, it is somewhat challenging to quantify the accuracy needed in solving (3.11) to induce a $(\mathbf{c}, \epsilon)$-Forster transform for $\epsilon > 0$, because Barthe's objective is not strongly convex. For example, combining Lemmas B.6, B.9 of [HM13] with Corollary 4 of [HM19] gives an estimate of $\approx \epsilon \exp(-nd)$ additive error sufficing. Our work drastically improves this estimate (cf. Lemma 3.3), showing it is enough to obtain an additive error that is polynomial in ϵ and $\min_{i \in [n]} \mathbf{c}_i$.

An optimistic bound on [HM13]'s runtime scales as $\approx n^2 d^{\omega-1} + n^3$ (using

the state-of-the-art CPM [JLSW20]), where additional poly(n, d) factors are saved using our improved error bounds. Incomparably, [AKS20] gave runtimes for first-order methods depending polynomially on either the inverse target accuracy $\frac{1}{\epsilon}$ (and hence precluding high-accuracy solutions), or the inverse strong convexity of Barthe's objective, which is data-dependent but can lose $\exp(d)$ factors or worse.

Instead of optimizing Barthe's objective, [DTK23, DR24] recently gave alternative approaches that either iteratively refine a right-scaling $\mathbf{R} \in \mathbb{R}^{d\times d}$ to satisfy (3.2), or refine a left-scaling $\mathbf{s} \in \mathbb{R}^n_{>0}$ to satisfy (3.7). These algorithms have the advantage of running in *strongly polynomial time*, i.e., the number of arithmetic operations needed only depends on n and $\frac{1}{\epsilon}$, rather than problem conditioning notions such as bit complexity. Designing strongly polynomial time algorithms is an interesting and important goal in its own right. For instance, [DTK23] was motivated by the connection of Forster transforms to learning halfspaces with noise [DKT21], a robust generalization of linear programming, which is a basic problem for which strongly polynomial time algorithms are unknown. In a different direction, [DR24] showed that a strongly polynomial algorithm for matrix scaling by [LSW00] could be adapted to Forster transforms.

Unfortunately, the resulting runtimes from these direct iterative methods that sidestep Barthe's objective are quite large. For example, [DTK23] claim a runtime of at least $\approx n^5 d^{11} \epsilon^{-5}$, and a recent improvement in [DR24] still requires at least $\approx n^4 d^{\omega-1} \log(\frac{1}{\epsilon})$ time.

There are a few other approaches to polynomial-time computation of approximate Forster transforms based on more general formulations of the problem, see e.g., [AGL+18, SV19]. We discuss these algorithms in more detail in Section 3.2.1 but note that they appear to lack explicit runtime bounds, and we believe they are subsumed by those described thus far.

In summary, existing methods for computing $(\mathbf{c}, \epsilon)$-Forster transforms have runtimes at least $\approx n^2 d^{\omega-1} + n^3$ (weakly polynomial) or $\approx n^4 d^{\omega-1}$ (strongly poly-

nomial). On the other hand, for related problems such as matrix scaling, near-optimal runtimes are known in well-conditioned regimes, via structured optimization methods that more faithfully capture the geometry of relevant objectives [CMTV17, ALdOW17].

3.2.1 Forster transforms via maximum entropy

An alternative characterization of Forster transforms was followed by [SV19], who studied certain *maximum entropy distribution* representations of specified marginals $\mathbf{c}$ with respect to an index set $\mathcal{S}$, which we briefly explain for context. In our setting of finding a $\mathbf{c}$-Forster transform of $\mathbf{A} \in \mathbb{R}^{n \times d}$, the index set $\mathcal{S}$ consists of all $S \subseteq [n]$ with $|S| = d$, and the underlying $\pi(S)$ is the *determinantal measure* with $\pi(S) \propto \det([\mathbf{A}^\top \mathbf{A}]_{S:S})$. Then, Section 8.3 of [SV19] applies the Cauchy–Binet formula to show that

$$\begin{aligned}
&\min_{\mathbf{t} \in \mathbb{R}^n} -\langle \mathbf{c}, \mathbf{t} \rangle + \log \det \left(\sum_{i \in [n]} \exp(\mathbf{t}_i) \, \mathbf{a}_i \mathbf{a}_i^\top \right) \\
= &\min_{\mathbf{t} \in \mathbb{R}^n} \log \left(\sum_{S \in \mathcal{S}} \exp(\langle \mathbf{1}_S - \mathbf{c}, \mathbf{t} \rangle) \det\left(\left[\mathbf{A}^\top \mathbf{A} \right]_{S:S} \right) \right) \\
= &\min_{\mathbf{t} \in \mathbb{R}^n} \log \left(\sum_{S \in \mathcal{S}} \pi(S) \exp(\langle \mathbf{1}_S - \mathbf{c}, \mathbf{t} \rangle) \right) + \log \left(\sum_{S \in \mathcal{S}} \det\left(\left[\mathbf{A}^\top \mathbf{A} \right]_{S:S} \right) \right),
\end{aligned}$$

where the starting expression is Barthe's objective (3.11). This shows that computing Forster transforms falls within the framework of Section 7 in [SV19], which exactly gives polynomial-time algorithms for optimizing functions in the form of the ending expression above.

The runtime of [SV19] is not explicit, and we believe that it runs more slowly than more direct approaches such as CPMs [HM13]. Similarly, [BLNW20] develop interior-point methods for solving maximum entropy optimization problems of the above form, but with runtimes scaling polynomially in $|\mathcal{S}|$, which in our setting is $\approx$

n^d. Finally, we mention [CKYV19, CKV20],[1] which also study variants of maximum entropy problems. Interestingly, [CKYV19] also applies a box-constrained Newton's method. However, it is unclear to us whether our problem falls in their framework, and their claimed runtimes (see Theorem 4.1, [CKYV19]) depend at least on $n^{4.5}$.

3.2.2 Forster transforms via operator scaling

In another direction, [GGdOW17] discovered a nontrivial connection between computing Forster transforms (phrased in an equivalent way of computing Brascamp-Lieb constants, see Proposition 1.8, [GGdOW17]) and a related problem called *operator scaling*. This implies that polynomial-time algorithms for operator scaling [GGdOW16, GGdOW17, IQS17, AGL+18] apply to our problem as well. However, none of the aforementioned algorithms for operator scaling have an explicitly specified polynomial, and a crude analysis results in fairly substantial blowups. Also, some of these algorithms have more explicit and stronger variants analyzed in [DTK23, DR24], so we believe they are subsumed by our existing discussion.

More generally, there is an active body of research on generalizations of operator scaling and Forster transforms [GGdOW17, Fra18, BFG+18, BFG+19], for which several state-of-the-art results are via variants of Newton's method, broadly defined. It would be interesting to explore if the ideas developed in this paper could extend to those settings as well.

3.2.3 Reductions between graph primitives

Our work on implicit sparsification (Theorem 3.2) fits into a line of work that aims to characterize which fast (i.e., $n^{1+o(1)}$-time) graph primitives imply others by reduction. This theme was explicitly considered by [ACSS20] (see also related work by [Qua21]),

[1] To provide some context, [CKV20] is the conference version of an unpublished preprint [CKYV19]. The box-constrained Newton's method discussed here only appears in the preprint version.

who studied these primitives for graphs implicitly defined by low-dimensional kernels. Among the three primitives of (1) fast matrix-vector multiplication, (2) fast spectral sparsification, and (3) fast Laplacian system solving, it was known previously that (3) reduces to (1) and (2) [ST04], and that (1) reduces to (2) and (3) [ACSS20]. Our work makes progress on this reduction landscape, as it shows (2) reduces to (1) (and hence, (3) also reduces to (1)). We mention that Theorem 5 in [JLM+23] gives a related, but slower, $\approx n^2$-time reduction from (2) to (3).

Our work is also thematically connected to prior work on spectral sparsification under weak graph access, e.g., in streaming and dynamic settings [KLM+17, ADK+16]. Specifically, several of the rounding-via-sketching tools used to prove Theorem 3.2 are inspired by [KMM+20]. Their result is incomparable to ours, as we are unable to directly access a sketch of the graph, so we instead use these tools to speed up an optimization method rather than identify the sparsifier in one shot.

3.3 Our results

The state of affairs in Section 3.2 prompts the natural question: can we obtain substantially faster algorithms for computing a $(\mathbf{c}, \epsilon)$-Forster transform? Our main contribution is to design such algorithms, primarily specialized to two settings which we call the *well-conditioned* and *smoothed analysis* regimes. We note that the distinction between well-conditioned and poorly-conditioned instances is a common artifact of fast algorithms for scaling problems, see e.g., discussions in [CMTV17, ALdOW17, BLNW20]. In particular, analogous works to ours for matrix scaling and balancing [CMTV17, ALdOW17] obtain nearly-linear runtimes in well-conditioned regimes, and polynomial runtime improvements in others.

For a fixed pair $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $\mathbf{c} \in (0, 1]^n$ satisfying $\|\mathbf{c}\|_1 = d$, we use the following notion of conditioning for the associated problem of computing a $\mathbf{c}$-Forster transform of $\mathbf{A}$.

Assumption 3.1. *For f defined in (3.11), there is $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$ satisfying $\|\mathbf{t}^\star\|_\infty \le \log(\kappa)$.*

To justify this, recall that $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$ induces the optimal left scaling, in the sense of (3.7), via $\mathbf{s}(\mathbf{t}^\star) = \exp(\frac{1}{2}\mathbf{t})$, entrywise. Further, Barthe's objective is invariant to translations by $\mathbf{1}_n$:

$$\begin{aligned} f(\mathbf{t}+\alpha\mathbf{1}_n) &= -\langle \mathbf{c}, \mathbf{t}+\alpha\mathbf{1}_n\rangle + \log\det(\mathbf{Z}(\mathbf{t}+\alpha\mathbf{1}_n)) \\ &= -\langle \mathbf{c},\mathbf{t}\rangle - \alpha d + \log\det(\mathbf{Z}(\mathbf{t})) + \log\det(\exp(\alpha)\mathbf{I}_d) = f(\mathbf{t}). \end{aligned} \tag{3.15}$$

Thus, we can always shift any minimizing $\mathbf{t}^\star$ so that its extreme coordinates average to 0, which achieves the tightest ℓ_∞ bound on $\mathbf{t}^\star$ via shifts by $\mathbf{1}_n$. This shows that κ in Assumption 3.1 is the ratio of the largest and smallest entries of the optimal scaling $\mathbf{s} \in \mathbb{R}^n_{>0}$ achieving (3.7).

3.3.1 Main theorem

We now state our main result on computing Forster transforms.

Theorem 3.1. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{c} \in (0,1]^n$ satisfy Assumption 3.1, and let $\delta, \epsilon \in (0,1)$. There is an algorithm that computes $\mathbf{R}$, a $(\mathbf{c},\epsilon)$-Forster transform of $\mathbf{A}$, with probability $\ge 1-\delta$, in time*

$$O\left(nd^{\omega-1}\log(\kappa)\left(\frac{n\log(\kappa)}{\delta\epsilon\mathbf{c}_{\min}}\right)^{o(1)}\right), \text{ where } \mathbf{c}_{\min} := \min_{i\in[n]}\mathbf{c}_i.$$

In the well-conditioned regime where $\kappa = \mathrm{poly}(n)$, Theorem 3.1 improves upon the state-of-the-art runtimes for radial isotropic position by a factor of $\approx \max(n, n^2 d^{1-\omega})$, up to a subpolynomial overhead in problem parameters.[2] Moreover, Theorem 3.1 approaches natural limits for computing Forster transforms. For

[2] As discussed earlier, to our knowledge, even that CPMs [HM13] obtain runtimes of $\approx n^2 d^{\omega-1} + n^3$ for well-conditioned instances was unknown previously. This is enabled by our improved error tolerance analysis in Lemma 3.3.

example, using current techniques, it takes $\approx nd^{\omega-1}$ time to perform basic relevant operations such as evaluating Barthe's objective (3.11), or verifying that a given right scaling $\mathbf{R} \in \mathbb{R}^{d\times d}$ or left scaling $\mathbf{s} \in \mathbb{R}^n_{>0}$ places a dataset in radial isotropic position.

Interestingly, our algorithm for optimizing Barthe's objective (3.11) is a variant of the *box-constrained Newton's method* of [CMTV17, ALdOW17], originally developed for approximate matrix scaling and balancing. For this reason, it is perhaps surprising that Theorem 3.1 depends linearly on n. Indeed, merely writing down the Hessian of Barthe's objective takes n^2 time, which dominates the runtime of Theorem 3.1 for $n \gg d$.

We were inspired to use this tool by noticing similarities between the derivative structure of Barthe's objective (Fact 3.1) and the *softmax* function, which can be viewed as the one-dimensional case of Barthe's objective. Previously, the softmax function was known to be *Hessian stable* in the ℓ_∞ norm (Definition 3.4), enabling local optimization oracles that can be implemented via Newton's method [CJJ+20] over ℓ_2 or ℓ_∞ norm balls.

It is much more challenging to prove that Barthe's objective is Hessian stable, as the proof in [CJJ+20] does not naturally extend to non-commuting variables. Nonetheless, we give a different proof inspired by Kadison's inequality in operator algebra [Kad52] to establish Hessian stability of Barthe's objective in Section 3.4.1. We complement this result in Section 3.4.2 with bounds on the additive error on Barthe's objective required to obtain a $(\mathbf{c}, \epsilon)$-Forster transform, for an approximation tolerance $\epsilon > 0$. By using the leverage score characterization (3.7) of exact Forster transforms, and performing a local perturbation analysis at the optimizer, we show that $\mathrm{poly}(\epsilon, \min_{i\in[n]} \mathbf{c}_i)$ error suffices. This significantly sharpens prior error tolerance bounds from [HM13, Che24], which scaled exponentially in a polynomial of the problem parameters.

With these stability bounds in place, the rest of Section 3.4 makes small modifications to the [CMTV17] analysis. We show that by using fast matrix multipli-

cation, each Hessian can be computed in $\approx n^2 d^{\omega-2}$ time (Lemma 3.5), and that box-constrained Newton steps can be efficiently implemented using the constrained optimization methods from [CPW21]. We then obtain the stated runtime via a technical tool of potential independent interest.

We mention here that the use of this last technical tool leads to the subpolynomial overheads in Theorem 3.1. By using explicit Hessian evaluations, we obtain an alternate runtime of

$$O\left(n^2 d^{\omega-2} \log(\kappa) \operatorname{polylog}\left(\frac{n \log(\kappa)}{\delta \epsilon \mathbf{c}_{\min}}\right)\right),$$

as described more formally in Lemma 3.5 and Remark 3.1, which yields (improved) polylogarithmic dependences on $\frac{1}{\delta}$, $\frac{1}{\epsilon}$, and $\frac{1}{\mathbf{c}_{\min}}$, at the cost of a multiplicative overhead of $\approx \frac{n}{d}$.

3.3.2 Implicit sparsification

Our fastest runtimes are obtained by using an implicit sparsifier for certain structured matrices. To explain its relevance to our setting, while the Hessian of Barthe's objective $\nabla^2 f$ is $n \times n$ and fully dense (as given in Fact 3.1), its structure is appealing in several regards. While we do not know how to compute $\nabla^2 f$ faster than in $\approx n^2 d^{\omega-2}$ time, we can access it via matrix-vector products in $O(nd^{\omega-1})$ time (cf. Lemma 3.5). In addition, $\nabla^2 f$ is actually a *graph Laplacian*, i.e., it belongs to a family of matrices that have enabled many powerful algorithmic primitives, such as *spectral sparsification*. For example, breakthroughs by [SS11, ST14] show that any $n \times n$ graph Laplacian $\mathbf{L}$ admits constant-factor spectral approximations with only $\approx n$ nonzero entries.

In this work, we add a new primitive to the graph Laplacian toolkit. We consider the following problem, which to our knowledge has not been explicitly studied before: given an (implicit) Laplacian $\mathbf{L}$ accessible only via a matrix-vector product oracle, how many queries are needed to produce an (explicit) spectral sparsifier of

$\mathbf{L}$? The sparsifier can then be used as a preconditioner, enabling faster second-order methods. Our main result to this end is the following.

Theorem 3.2. *Let $\mathbf{L}$ be an $n \times n$ graph Laplacian, and let $\mathcal{O} : \mathbb{R}^n \to \mathbb{R}^n$ be an oracle that returns $\mathbf{L}\mathbf{v}$ on input $\mathbf{v} \in \mathbb{R}^n$. Let $\delta \in (0,1)$, $\Delta \in (0, \mathrm{Tr}(\mathbf{L}))$, and let $\mathbf{\Pi} := \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ be the projection matrix to the subspace of $\mathbb{R}^n$ orthogonal to $\mathbf{1}_n$. There is an algorithm that takes as inputs $(\mathcal{O}, \delta, \Delta)$, and with probability $\geq 1 - \delta$, it returns $\widetilde{\mathbf{L}}$, an $n \times n$ graph Laplacian satisfying*

$$\mathbf{L} + \Delta\mathbf{\Pi} \preceq \widetilde{\mathbf{L}} \preceq \left(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta}\right)^{o(1)} (\mathbf{L} + \Delta\mathbf{\Pi}), \ \mathrm{nnz}(\widetilde{\mathbf{L}}) = n \cdot \left(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta}\right)^{o(1)}, \tag{3.16}$$

using $(\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta})^{o(1)}$ queries to $\mathcal{O}$, and $n \cdot (\frac{n\mathrm{Tr}(\mathbf{L})}{\Delta\delta})^{o(1)}$ additional time.

For $\delta = \mathrm{poly}(\frac{1}{n})$ and $\mathrm{poly}(n)$-well conditioned graph Laplacians, Theorem 3.2 produces a spectral sparsifier of a Laplacian $\mathbf{L}$ using $n^{o(1)}$ matrix-vector products and $n^{1+o(1)}$ additional time. The approximation quality of the sparsifier is somewhat poor, i.e., $n^{o(1)}$, but in algorithmic contexts (such as that of Theorem 3.1), this is sufficient for use as a low-overhead preconditioner.

We believe Theorem 3.2 may be of independent interest to the graph algorithms and numerical linear algebra communities, as it enhances the flexibility of existing Laplacian-based tools. We are optimistic that its use can extend the reach of fast second-order methods for combinatorially-structured optimization problems.

The proof of Theorem 3.2 extends the work of [JLM+23], which uses a reduction to packing SDPs and applies packing SDP solvers from the literature [ALO16, PTZ16, JLT20] along with a homotopy scheme. For brevity, we omit the rather technical proof here and refer instead to Section 4 of [JLT25].

3.3.3 Smoothed regime

Our third main contribution is to provide explicit bounds on the problem conditioning κ in Assumption 3.1, for "beyond worst-case" inputs $\mathbf{A}$. We specialize our result to the

smoothed analysis setting, a well-established paradigm for beyond worst-case analysis in the theoretical computer science community [ST04, Rou20]. In our smoothed setting, we perturb entries of our input by Gaussian noise at noise level $\sigma > 0$. This is a standard smoothed matrix model used in the study of linear programming algorithms [ST04, SST06].

Here, we state the basic variant of our conditioning bound in the smoothed analysis regime.

Theorem 3.3. *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ have rows $\{\mathbf{a}_i\}_{i \in [n]}$ such that $\|\mathbf{a}_i\|_2 = 1$ for all $i \in [n]$, let $\mathbf{c} := \frac{d}{n}\mathbf{1}_n$, let $\delta \in (0,1)$, and let $\sigma \in (0, \frac{\delta}{10nd})$. Let $\widetilde{\mathbf{A}} := \mathbf{A} + \mathbf{G}$, where $\mathbf{G} \in \mathbb{R}^{n \times d}$ has entries $\sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma^2)$. Then with probability $\geq 1 - \delta$, if $n \geq Cd$ where C is any constant larger than 1, Assumption 3.1 holds for Barthe's objective f defined with respect to $(\widetilde{\mathbf{A}}, \mathbf{c})$, where*

$$\log(\kappa) = O\left(d \log\left(\frac{1}{\sigma}\right)\right).$$

That is, $\mathbf{A}$ in Theorem 3.3 is a "base worst-case instance" that is smoothed into a more typical instance $\widetilde{\mathbf{A}}$, which our conditioning bound of $\kappa \approx (\frac{1}{\sigma})^{O(d)}$ applies to.

The assumption in Theorem 3.3 that $\mathbf{A}$ has unit norm rows is relatively mild; rescaling rows does not affect the (base) Forster transform problem, and our result still applies if row norms are in a $\text{poly}(n)$ multiplicative range. Further, while Theorem 3.3 is stated for uniform marginals $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$, we show in Corollary 3.1 that as long as the marginals $\mathbf{c}$ are bounded away from 1 entrywise by a constant, a similar conditioning estimate still holds for sufficiently large n. The requirement that $n \geq Cd$ for $C > 1$ is a minor bottleneck of our approach, discussed in Remark 3.2.

We are aware of few explicit conditioning bounds for Forster transforms such as Theorem 3.3, so we hope it (and techniques used in establishing it) become useful in future studies. Among conditioning bounds that exist presently, Lemma B.6 of [HM13] (cf. discussion in Corollary 4, [HM19]) shows that for $\mathbf{A}$ with rows in *general*

position, we have $\log(\kappa) = O(n \log(\frac{1}{D}))$, where D is the smallest determinant of a nonsingular $d \times d$ submatrix of $\mathbf{A}$. In particular, D can be inverse-exponential in d (or worse) for poorly-behaved instances. A crude lower bound of $D \gtrsim \exp(-d^3)$ was provided in [Che24] for essentially the smoothed model we consider in Theorem 3.3.

On the other hand, [DTK23, DR24], who respectively design strongly polynomial methods for iteratively updating an approximate Forster transform $\mathbf{R} \in \mathbb{R}^{d \times d}$ or dual scaling $\mathbf{s} \in \mathbb{R}^n_{>0}$, bound related conditioning quantities. Both papers contain results (cf. Section 5, [DTK23] and Section 4, [DR24]) showing that any iterate has a "nearby" iterate in bounded precision, that does not significantly affect some potential function of interest. These results do not appear to directly have implications for Assumption 3.1, and provide rather large bounds on the bit complexity (focusing on worst-case instances). Nonetheless, exploring connections in future work could be fruitful.

Most relatedly, Theorem 1.5 of [AKS20] proves that for target marginals $\mathbf{c}$ that are "deep" inside the basis polytope for independent sets of $\mathbf{A}$'s rows, $\log(\kappa) \lesssim d$. However, [AKS20] does not give estimates on the deepness of marginals in concrete models, and indeed our approach to proving Theorem 3.3 is to provide such explicit bounds in the smoothed analysis regime.

To prove Theorem 3.3, in Definition 3.6 we first extend an approach of [AKS20] that defines a notion of *deepness* of marginal vectors $\mathbf{c}$ inside the basis polytope induced by $\mathbf{A}$'s independent row subsets. As we recall in Section 3.5.1, [AKS20] argues that if $\mathbf{c}$ has deepness of $\eta = \Omega(1)$, then we can obtain a conditioning bound of $\log(\kappa) \approx d$ in Assumption 3.1. In the case of $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$, this roughly translates to a robust variant of (3.5) that says: for all subspaces $E \subseteq \mathbb{R}^d$ of dimension k, at most a $\approx \frac{k}{d}$ fraction of $\mathbf{A}$'s rows (after smoothing by Gaussian noise) should lie at distance $\mathrm{poly}(\frac{1}{n})$ from E. The rest of Section 3.5 proves this deepness result for smoothed matrices $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{G}$.

The key challenge is to avoid union bounding over a net of all possible sub-

spaces E; for $\dim(E) = \Theta(d)$, this naïve approach would require taking $n \gtrsim d^2$ samples, as nets of $\Theta(d)$-dimensional subspaces have cardinality $\approx \exp(d^2)$. We instead show in Lemma 3.9 that deepness is implied by appropriate submatrices of $\widetilde{\mathbf{A}}$ having large singular values, allowing us to apply union bounds to a smaller number of *data-dependent* subspaces. We combine this observation with singular value estimates from the random matrix theory literature to prove Theorem 3.3. Our argument requires some casework on the subspace dimension; we handle wide and near-square submatrices in Section 3.5.2 and tall submatrices in Section 3.5.3.

Directly combining Theorems 3.1 and 3.3 shows that for smoothed instances, the complexity of computing an approximate Forster transform is at most $\approx nd^\omega$, up to a subpolynomial factor. While it is worse than our well-conditioned runtime, our method in Theorem 3.1 still improves upon state-of-the-art algorithms based on CPMs by a factor of $\approx \max(\frac{n}{d}, n^3 d^{-\omega})$ in this regime.

3.3.4 Computational model

We briefly mention that we work in the real RAM model, where we bound the number of basic arithmetic operations. Prior work on optimizing Barthe's objective [HM13, AKS20] also worked in this model. A detailed investigation of the numerical stability of Forster transforms is an important direction for future work, but it is outside of our scope.

3.4 Optimizing Barthe's objective via Newton's method

In this section, we give our algorithm for computing approximate Forster transforms.

We first define the following notion of multiplicative stability for analyzing Newton's method, patterned off [CMTV17, KSJ18, CJJ+20].

Definition 3.4 (Hessian stability)**.** We say that twice-differentiable $f : \mathbb{R}^n \to \mathbb{R}$ is

(r, ϵ)-*Hessian stable with respect to norm* $\|\cdot\|$ if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with $\|\mathbf{x} - \mathbf{y}\| \leq r$,

$$\nabla^2 f(\mathbf{x}) \approx_\epsilon \nabla^2 f(\mathbf{y}).$$

[CMTV17] called this property "second-order robustness." Our algorithm is a variant of the box-constrained Newton's method of [CMTV17], which solves box-constrained quadratics to optimize a Hessian-stable function in $\|\cdot\|_\infty$ to high precision.

We first make our key technical observation in Section 3.4.1: that Barthe's objective is Hessian-stable with respect to $\|\cdot\|_\infty$. We then give a termination condition in Section 3.4.2 that suffices for $\mathbf{t} \in \mathbb{R}^n$ to induce an ϵ-Forster transform $\mathbf{R}(\mathbf{t})$. In Section 3.4.3, we leverage our implicit Laplacian sparsification algorithm from Theorem 3.2 to implement the iteration of the [CMTV17] Newton's method. We put all the pieces together in Section 3.4.4 to give our main result.

Throughout this section, we fix a pair $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $\mathbf{c} \in (0, 1]^n$ satisfying $\|\mathbf{c}\|_1 = d$ and (3.5). We follow the notation outlined in Section 3.1, in particular, (3.10), (3.11), and (3.12). We also will state our results under the diameter bound in Assumption 3.1.

3.4.1 Hessian stability of Barthe's objective

In this section, we prove the following key structural result enabling our approach.

Proposition 3.4. *For all $r > 0$, f is $(r, 2r)$-Hessian stable with respect to $\|\cdot\|_\infty$.*

A similar result to Proposition 3.4 was previously established for the softmax objective

$$\mathbf{t} \to \log \left(\sum_{i \in [n]} \exp(\mathbf{t}_i) \right),$$

in Lemma 14, [CJJ+20], using more elementary techniques, i.e., directly establishing that the softmax satisfies the following third-order regularity property.

Definition 3.5 (Quasi-self-concordance)**.** We say that thrice-differentiable and convex $f : \mathbb{R}^n \to \mathbb{R}$ is *M-quasi-self-concordant (QSC) with respect to norm $\|\cdot\|$* if for all $\mathbf{u}, \mathbf{h}, \mathbf{x} \in \mathbb{R}^n$,

$$\left|\nabla^3 f(\mathbf{x})[\mathbf{u}, \mathbf{u}, \mathbf{h}]\right| \leq M \|\mathbf{h}\| \|\mathbf{u}\|^2_{\nabla^2 f(\mathbf{x})}.$$

We provide an alternative proof of the following result in [CJJ+20], which is of independent interest.

Proposition 3.5. *The softmax function*

$$g(\mathbf{t}) = \log\left(\sum_{i \in [n]} \exp(\mathbf{t}_i)\right)$$

is 2-QSC with respect to $\|\cdot\|_\infty$.

Proof. For $\mathbf{x} \in \mathbb{R}^n$, define $P(\mathbf{x})$ as the distribution over $\mathbf{e}_i$ given by

$$\Pr_{\mathbf{y} \sim P(\mathbf{x})}[\mathbf{y} = \mathbf{e}_i] = \exp(\mathbf{x}_i - g(\mathbf{x})).$$

Observe that $\sum_{i \in [n]} \exp(\mathbf{x}_i - g(\mathbf{x})) = 1$, so $P(\mathbf{x})$ is indeed a probability distribution. By straightforward calculation, where $\otimes$ denotes the outer product,

$$\begin{aligned}
\nabla g(\mathbf{x}) &= \mathbb{E}_{\mathbf{y} \sim P(\mathbf{x})}[\mathbf{y}] \\
\nabla^2 g(\mathbf{x}) &= \mathbb{E}_{\mathbf{y} \sim P(\mathbf{x})}[(\mathbf{y} - \nabla g(\mathbf{x})) \otimes (\mathbf{y} - \nabla g(\mathbf{x}))] \\
\nabla^3 g(\mathbf{x}) &= \mathbb{E}_{\mathbf{y} \sim P(\mathbf{x})}[(\mathbf{y} - \nabla g(\mathbf{x})) \otimes (\mathbf{y} - \nabla g(\mathbf{x})) \otimes (\mathbf{y} - \nabla g(\mathbf{x}))]
\end{aligned}$$

Now by Hölder's inequality (Fact A.18) and the fact that $\|\mathbf{y}\|_1 = 1$ and $\|\nabla g(\mathbf{x})\|_1 = 1$,

$$|\langle \mathbf{y} - \nabla g(\mathbf{x}), \mathbf{h} \rangle| \leq \|\mathbf{y} - \nabla g(\mathbf{x})\|_1 \|\mathbf{h}\|_\infty \leq 2 \|\mathbf{h}\|_\infty$$

for all $\mathbf{h} \in \mathbb{R}^n$. Thus

$$\begin{aligned}
\left|\nabla^3 g(\mathbf{x})[\mathbf{u}, \mathbf{u}, \mathbf{h}]\right| &= \left|\mathbb{E}_{\mathbf{y} \sim P(\mathbf{x})}[\langle \mathbf{y} - \nabla g(\mathbf{x}), \mathbf{h} \rangle (\mathbf{y} - \nabla g(\mathbf{x})) \otimes (\mathbf{y} - \nabla g(\mathbf{x}))][\mathbf{u}, \mathbf{u}]\right| \\
&\leq 2 \|\mathbf{h}\|_\infty \nabla^2 g(\mathbf{x})[\mathbf{u}, \mathbf{u}]
\end{aligned}$$

for all $\mathbf{u}, \mathbf{h}, \mathbf{x} \in \mathbb{R}^n$, as desired. □

Due to complications arising from Barthe's objective being defined with respect to potentially non-commuting matrices, we follow a substantially different approach in this section. We first prove the following helper lemma, which shows that taking a Schur complement preserves the Loewner order.

Lemma 3.2. *If* $\mathbf{M}, \mathbf{N} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ *and* $\mathbf{M} \approx_\epsilon \mathbf{N}$*, then*

$$\mathrm{SC}(\mathbf{M}, S) \approx_\epsilon \mathrm{SC}(\mathbf{N}, S) \;\; \textit{for all } S \subseteq [d].$$

Proof. It suffices to show that if $\mathbf{A}, \mathbf{B} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$,

$$\mathbf{A} \succeq \mathbf{B} \implies \mathrm{SC}(\mathbf{A}, S) \succeq \mathrm{SC}(\mathbf{B}, S). \tag{3.17}$$

The claim then follows by applying (3.17) with $(\mathbf{A}, \mathbf{B}) \leftarrow (\exp(\epsilon)\mathbf{M}, \mathbf{N})$ and $\leftarrow (\exp(\epsilon)\mathbf{N}, \mathbf{M})$, since $\mathrm{SC}(\alpha\mathbf{M}, S) = \alpha\mathrm{SC}(\mathbf{M}, S)$ for any scaling coefficient $\alpha \in \mathbb{R}$.

We now establish (3.17). It is well-known (see, e.g., Appendix A.5.5 of [BV04b]) that

$$\mathbf{x}^\top \mathrm{SC}(\mathbf{A}, S)\,\mathbf{x} = \min_{\mathbf{y}\in\mathbb{R}^{S^c}} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^\top \mathbf{A} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}$$

for all $S \subseteq [d]$ and $\mathbf{x} \in \mathbb{R}^S$, where $S^c := [d] \setminus S$. Now (3.17) follows from

$$\mathbf{x}^\top \mathrm{SC}(\mathbf{A}, S)\,\mathbf{x} = \min_{\mathbf{y}\in\mathbb{R}^{S^c}} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^\top \mathbf{A} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \geq \min_{\mathbf{y}\in\mathbb{R}^{S^c}} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^\top \mathbf{B} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \mathbf{x}^\top \mathrm{SC}(\mathbf{B}, S)\,\mathbf{x}.$$

□

We are now ready to prove Proposition 3.4.

Proof of Proposition 3.4. Throughout, fix $\mathbf{t}, \mathbf{t}' \in \mathbb{R}^n$ with $\|\mathbf{t} - \mathbf{t}'\|_\infty \leq r$. Our goal is to show

$$\nabla^2 f(\mathbf{t}) \approx_{2r} \nabla^2 f(\mathbf{t}').$$

We follow the notation (3.10), (3.12), and whenever the argument is dropped, it is implied to be at $\mathbf{t}$; we will use a superscript $'$ whenever the argument is at $\mathbf{t}'$. So, for

example, $\mathbf{M}_i \equiv \mathbf{M}_i(\mathbf{t})$ and $\mathbf{M}'_i \equiv \mathbf{M}_i(\mathbf{t}')$ for all $i \in [n]$. Also, to ease notation in this proof we define

$$\mathbf{C}_i := \exp(\mathbf{t}_i)\,\mathbf{a}_i\mathbf{a}_i^\top,\ \mathbf{C}'_i := \exp(\mathbf{t}'_i)\,\mathbf{a}_i\mathbf{a}_i^\top,$$

so that $\mathbf{M}_i = \mathbf{R}\mathbf{C}_i\mathbf{R}$ for all $i \in [n]$. We first claim that for all $i \in [n]$,

$$\begin{pmatrix} \mathbf{C}_i & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_i & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \approx_r \begin{pmatrix} \mathbf{C}'_i & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}'_i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}'_i & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}'_i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}, \tag{3.18}$$

where both matrices in (3.18) have dimensions $(n+1)d \times (n+1)d$, and only the $(1,1)$, $(i+1,1)$, $(1,i+1)$, and $(i+1,i+1)$-indexed $d \times d$ blocks are nonzero. We can verify (3.18) by direct expansion with respect to a $2d$-dimensional test vector with blocks $\mathbf{x}, \mathbf{y}$, which reduces the claim to

$$(\mathbf{x}+\mathbf{y})^\top \mathbf{C}_i(\mathbf{x}+\mathbf{y}) \approx_r (\mathbf{x}+\mathbf{y})^\top \mathbf{C}'_i(\mathbf{x}+\mathbf{y}) \impliedby \mathbf{C}_i \approx_r \mathbf{C}'_i,$$

where the latter fact above follows from $\|\mathbf{t} - \mathbf{t}'\|_\infty \leq r$. Summing (3.18) for all $i \in [n]$ shows

$$\mathbf{L} \approx_r \mathbf{L}',\ \text{where } \mathbf{L} := \begin{pmatrix} \sum_{i\in[n]} \mathbf{C}_i & \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_n \\ \mathbf{C}_1 & \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_2 & \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_n \end{pmatrix},$$

$$\mathbf{L}' := \begin{pmatrix} \sum_{i\in[n]} \mathbf{C}'_i & \mathbf{C}'_1 & \mathbf{C}'_2 & \cdots & \mathbf{C}'_n \\ \mathbf{C}'_1 & \mathbf{C}'_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}'_2 & \mathbf{0} & \mathbf{C}'_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}'_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}'_n \end{pmatrix}.$$

Next, using Lemma 3.2 to take Schur complements of $\mathbf{L}, \mathbf{L}'$ onto the index set $[(n+1)d] \setminus [d]$ shows

$$\mathbf{K} \approx_r \mathbf{K}', \text{ where } \mathbf{K} := \begin{pmatrix} \mathbf{C}_1 - \mathbf{C}_1\mathbf{Z}^{-1}\mathbf{C}_1 & \cdots & -\mathbf{C}_1\mathbf{Z}^{-1}\mathbf{C}_n \\ -\mathbf{C}_2\mathbf{Z}^{-1}\mathbf{C}_1 & \cdots & -\mathbf{C}_2\mathbf{Z}^{-1}\mathbf{C}_n \\ \vdots & \ddots & \vdots \\ -\mathbf{C}_n\mathbf{Z}^{-1}\mathbf{C}_1 & \cdots & \mathbf{C}_n - \mathbf{C}_n\mathbf{Z}^{-1}\mathbf{C}_n \end{pmatrix},$$

$$\mathbf{K}' := \begin{pmatrix} \mathbf{C}'_1 - \mathbf{C}'_1(\mathbf{Z}')^{-1}\mathbf{C}'_1 & \cdots & -\mathbf{C}'_1(\mathbf{Z}')^{-1}\mathbf{C}'_n \\ -\mathbf{C}'_2(\mathbf{Z}')^{-1}\mathbf{C}'_1 & \cdots & -\mathbf{C}'_2(\mathbf{Z}')^{-1}\mathbf{C}'_n \\ \vdots & \ddots & \vdots \\ -\mathbf{C}'_n(\mathbf{Z}')^{-1}\mathbf{C}'_1 & \cdots & \mathbf{C}'_n - \mathbf{C}'_n(\mathbf{Z}')^{-1}\mathbf{C}'_n \end{pmatrix},$$

where we used that $\mathbf{Z} = \sum_{i\in[n]} \mathbf{C}_i$ and $\mathbf{Z}' = \sum_{i\in[n]} \mathbf{C}'_i$. Finally, fix some vector $\mathbf{v} \in \mathbb{R}^n$. Let

$$\mathbf{J} := \mathbf{v}\mathbf{v}^\top \otimes \mathbf{Z}^{-1} = \begin{pmatrix} \mathbf{v}_1^2\mathbf{Z}^{-1} & \mathbf{v}_1\mathbf{v}_2\mathbf{Z}^{-1} & \mathbf{v}_1\mathbf{v}_3\mathbf{Z}^{-1} & \cdots & \mathbf{v}_1\mathbf{v}_n\mathbf{Z}^{-1} \\ \mathbf{v}_1\mathbf{v}_2\mathbf{Z}^{-1} & \mathbf{v}_2^2\mathbf{Z}^{-1} & \mathbf{v}_2\mathbf{v}_3\mathbf{Z}^{-1} & \cdots & \mathbf{v}_2\mathbf{v}_n\mathbf{Z}^{-1} \\ \mathbf{v}_1\mathbf{v}_3\mathbf{Z}^{-1} & \mathbf{v}_2\mathbf{v}_3\mathbf{Z}^{-1} & \mathbf{v}_3^2\mathbf{Z}^{-1} & \cdots & \mathbf{v}_3\mathbf{v}_n\mathbf{Z}^{-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_1\mathbf{v}_n\mathbf{Z}^{-1} & \mathbf{v}_2\mathbf{v}_n\mathbf{Z}^{-1} & \mathbf{v}_3\mathbf{v}_n\mathbf{Z}^{-1} & \cdots & \mathbf{v}_n^2\mathbf{Z}^{-1} \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product. Similarly define $\mathbf{J}' := \mathbf{v}\mathbf{v}^\top \otimes (\mathbf{Z}')^{-1}$. Because we established each $\mathbf{C}_i \approx_r \mathbf{C}'_i$, we also have $\mathbf{Z} \approx_r \mathbf{Z}'$ and thus $\mathbf{Z}^{-1} \approx_r (\mathbf{Z}')^{-1}$. By well-known properties of the Kronecker product (cf. Theorem 2.3, [Sch13]), we conclude that $\mathbf{J} \approx_r \mathbf{J}'$. We have thus shown:

$$\mathbf{K} \approx_r \mathbf{K}', \ \mathbf{J} \approx_r \mathbf{J}'.$$

Now by Fact A.4, $\langle \mathbf{K}, \mathbf{J} \rangle \approx_{2r} \langle \mathbf{K}', \mathbf{J}' \rangle$. The conclusion follows upon realizing

$$\begin{aligned} \langle \mathbf{K}, \mathbf{J} \rangle &= \sum_{i\in[n]} \mathbf{v}_i^2 \mathrm{Tr}\left(\mathbf{Z}^{-1}\mathbf{C}_i\right) - \sum_{(i,j)\in[n]\times[n]} \mathbf{v}_i\mathbf{v}_j \mathrm{Tr}\left(\mathbf{Z}^{-1}\mathbf{C}_i\mathbf{Z}^{-1}\mathbf{C}_j\right) \\ &= \sum_{i\in[n]} \mathbf{v}_i^2 \mathrm{Tr}\left(\mathbf{M}_i\right) - \sum_{(i,j)\in[n]\times[n]} \mathbf{v}_i\mathbf{v}_j \mathrm{Tr}\left(\mathbf{M}_i\mathbf{M}_j\right) = \mathbf{v}^\top \nabla^2 f(\mathbf{t})\mathbf{v}, \end{aligned}$$

and similarly, $\langle \mathbf{K}', \mathbf{J}' \rangle = \mathbf{v}^\top \nabla^2 f(\mathbf{t}')\mathbf{v}$, by comparing to Fact 3.1 and using Fact A.3. This establishes $\mathbf{v}^\top \nabla^2 f(\mathbf{t})\mathbf{v} \approx_{2r} \mathbf{v}^\top \nabla^2 f(\mathbf{t}')\mathbf{v}$ for all $\mathbf{v} \in \mathbb{R}^n$, as desired. $\square$

3.4.2 Termination condition

In this section, we quantify the suboptimality gap (with respect to Barthe's objective) needed for $\mathbf{t} \in \mathbb{R}^n$ to induce a $(\mathbf{c}, \epsilon)$-Forster transform $\mathbf{R}(\mathbf{t})$, as a function of ϵ and problem parameters. Our proof makes use of local adjustments and is inspired by a similar technique in [CMTV17].

Lemma 3.3. *Let $\epsilon \in (0,1)$, and suppose $\mathbf{t} \in \mathbb{R}^n$ satisfies*

$$f(\mathbf{t}) - f(\mathbf{t}^\star) \leq \frac{\epsilon^2 \min_{i\in[n]} \mathbf{c}_i^2}{2}, \tag{3.19}$$

where $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$. Then $\mathbf{R}(\mathbf{t})$ is a $(\mathbf{c}, \epsilon)$-Forster transform.

Proof. We prove the contrapositive. Suppose $\mathbf{R}(\mathbf{t})$ is not a $(\mathbf{c}, \epsilon)$-Forster transform. By Lemma 3.1, there are two cases of leverage score violations to consider. We show that both cases contradict (3.19), by designing local improvements to $\mathbf{t}$ in any coordinate with a violating leverage score.

Case 1. Suppose that for some $i \in [n]$, we have $\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) > \exp(\epsilon)\mathbf{c}_i \geq (1+\epsilon)\mathbf{c}_i$ by Fact A.21. Let $\mathbf{t}' := \mathbf{t} - \delta\mathbf{e}_i$ for some choice of $\delta > 0$ that we will optimize later. Then,

$$\begin{aligned}
f(\mathbf{t}) - f(\mathbf{t}') &= \log\left(\frac{\det(\mathbf{Z}(\mathbf{t}))}{\det(\mathbf{Z}(\mathbf{t}'))}\right) - \delta\mathbf{c}_i \\
&= \log\left(\frac{\det(\mathbf{Z}(\mathbf{t}))}{\det\left(\mathbf{Z}(\mathbf{t}) + (\exp(-\delta)-1)\exp(\mathbf{t}_i)\,\mathbf{a}_i\mathbf{a}_i^\top\right)}\right) - \delta\mathbf{c}_i \\
&= \log\left(\frac{\det(\mathbf{Z}(\mathbf{t}))}{\det(\mathbf{Z}(\mathbf{t}))\left(1 + (\exp(-\delta)-1)\exp(\mathbf{t}_i)\,\mathbf{a}_i^\top\mathbf{Z}(\mathbf{t})^{-1}\mathbf{a}_i\right)}\right) - \delta\mathbf{c}_i \\
&= \log\left(\frac{\det(\mathbf{Z}(\mathbf{t}))}{\det(\mathbf{Z}(\mathbf{t}))\left(1 + (\exp(-\delta)-1)\,\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A})\right)}\right) - \delta\mathbf{c}_i \\
&= -\log\left(1 + (\exp(-\delta)-1)\,\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A})\right) - \delta\mathbf{c}_i \\
&\geq (1-\exp(-\delta))\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) - \delta\mathbf{c}_i \\
&\geq \left(\delta - \frac{\delta^2}{2}\right)\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) - \delta\mathbf{c}_i > \delta\epsilon\mathbf{c}_i - \frac{\delta^2}{2} \geq \frac{1}{2}(\epsilon\mathbf{c}_i)^2,
\end{aligned}$$

where the first two lines expanded definitions, the third line uses the matrix determinant lemma (Fact A.7), the fourth line uses the definition of leverage scores (3.6), the sixth line uses Fact A.22, and the seventh line uses Fact A.24, $\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) > (1+\epsilon)\mathbf{c}_i$, and $\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) \leq 1$ (Fact A.5). By choosing the optimal $\delta = \mathbf{c}_i\epsilon$, we have a contradiction to (3.19).

Case 2. Suppose that for some $i \in [n]$, we have $\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) < \exp(-\epsilon)\mathbf{c}_i \leq \frac{\mathbf{c}_i}{1+\epsilon}$ by Fact A.23. Let $\mathbf{t}' := \mathbf{t} + \delta\mathbf{e}_i$ for some choice of $\delta > 0$ that we will optimize later. Then, following analogous derivations as before,

$$\begin{aligned} f(\mathbf{t}) - f(\mathbf{t}') &= -\log\left(1 + (\exp(\delta) - 1)\,\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A})\right) + \delta\mathbf{c}_i \\ &> -\log\left(1 + (\exp(\delta) - 1)\,\frac{\mathbf{c}_i}{1+\epsilon}\right) + \delta\mathbf{c}_i \\ &= \log(1+\epsilon) - (1-\mathbf{c}_i)\log\left(1 + \frac{\epsilon}{1-\mathbf{c}_i}\right) \geq \frac{1}{2}(\epsilon\mathbf{c}_i)^2, \end{aligned}$$

where the second line uses $\boldsymbol{\tau}_i(\mathbf{S}(\mathbf{t})\mathbf{A}) < \frac{\mathbf{c}_i}{1+\epsilon}$, the third line chose $\delta = \log(1 + \frac{\epsilon}{1-\mathbf{c}_i})$ and used Fact A.25. Again this gives a contradiction to (3.19). We remark that the case of $\mathbf{c}_i = 1$ can be handled using a limiting argument. □

3.4.3 Box-constrained Newton's method

Thus far we have established that f is Hessian stable in $\|\cdot\|_\infty$ (Proposition 3.4) and needs to be minimized to error

$$\frac{\epsilon^2 \min_{i\in[n]} \mathbf{c}_i^2}{2}$$

for our desired application (Lemma 3.3). We also are given under Assumption 3.1 that the global minimizer lies inside $\mathbb{B}_\infty(\log(\kappa))$.

It remains to give an algorithm for efficiently optimizing Hessian stable functions. Fortunately, such a toolkit was provided by [CMTV17, CPW21]. The former work designed an approximation-tolerant box-constrained Newton's method, tailored towards objectives whose Hessians display a certain combinatorial structure, and the latter work showed how to optimize box-constrained quadratics in these structured Hessians. We can leverage this toolkit due to the next observation.

Lemma 3.4. *For all* $\mathbf{t} \in \mathbb{R}^n$, $\nabla^2 f(\mathbf{t})$ *is a* graph Laplacian, *i.e.,* $\nabla^2_{ij} f(\mathbf{t}) \geq 0$ *iff* $i = j$, *and*

$$\sum_{j \in [n]} \nabla^2_{ij} f(\mathbf{t}) = 0 \text{ for all } i \in [n].$$

Proof. The first property is immediate by inspection of (3.13), and using that all the $\mathbf{M}_i(\mathbf{t}) \in \mathbb{S}^{d \times d}_{\succeq \mathbf{0}}$. The second property follows because $\sum_{j \in [n]} \mathbf{M}_j(\mathbf{t}) = \mathbf{I}_d$, so for all $i \in [n]$ we have

$$\sum_{j \in [n]} \nabla^2_{ij} f(\mathbf{t}) = \operatorname{Tr}(\mathbf{M}_i(\mathbf{t})) - \left\langle \mathbf{M}_i(\mathbf{t}), \sum_{j \in [n]} \mathbf{M}_j(\mathbf{t}) \right\rangle = \operatorname{Tr}(\mathbf{M}_i(\mathbf{t})) - \langle \mathbf{M}_i(\mathbf{t}), \mathbf{I}_d \rangle = 0.$$

□

We remark that Lemma 3.4 gives another short proof of f's convexity: it is well-known that graph Laplacians are PSD matrices, which follows e.g., by the Gershgorin circle theorem.

One complication that arises in our algorithm is that computing the Hessian $\nabla^2 f$ is more expensive than providing matrix-vector query access to it, due to a convenient factorization.

Lemma 3.5. *Given* $\mathbf{t} \in \mathbb{R}^n$, *we can compute* $\nabla f(\mathbf{t})$ *in* $O(nd^{\omega-1})$ *time and* $\nabla^2 f(\mathbf{t})$ *in* $O(n^2 d^{\omega-2})$ *time. Additionally, given* $\mathbf{t}, \mathbf{v} \in \mathbb{R}^n$, *we can compute* $\nabla^2 f(\mathbf{t})\mathbf{v}$ *in* $O(nd^{\omega-1})$ *time.*

Proof. Recall the formulas for $\nabla f(\mathbf{t})$ and $\nabla^2 f(\mathbf{t})$ in Fact 3.1. For the former claim, following (3.10), (3.12), we first compute $\mathbf{S}(\mathbf{t})\mathbf{A}$ in time $O(nd)$, which lets us compute $\mathbf{Z}(\mathbf{t})$ in time $O(nd^{\omega-1})$ by multiplying $d \times n$ and $n \times d$ matrices. We can then compute $\mathbf{AR}(\mathbf{t})$ to obtain all of the vectors $\tilde{\mathbf{a}}_i(\mathbf{t})$ in $O(nd^{\omega-1})$ time. This lets us obtain all $\operatorname{Tr}(\mathbf{M}_i(\mathbf{t})) = \mathbf{s}_i(\mathbf{t})^2 \|\tilde{\mathbf{a}}_i(\mathbf{t})\|_2^2$ in $O(nd)$ additional time.

It remains to compute the $n \times n$ matrix with $(i,j)^{\text{th}}$ entry $\operatorname{Tr}(\mathbf{M}_i(\mathbf{t})\mathbf{M}_j(\mathbf{t}))$. Observe that

$$\operatorname{Tr}(\mathbf{M}_i(\mathbf{t})\mathbf{M}_j(\mathbf{t})) = \mathbf{s}_i(\mathbf{t})^2 \mathbf{s}_j(\mathbf{t})^2 \left\langle \tilde{\mathbf{a}}_i(\mathbf{t}), \tilde{\mathbf{a}}_j(\mathbf{t}) \right\rangle^2.$$

Thus it is enough to form the matrix with $(i,j)^{\text{th}}$ entry $\langle \tilde{\mathbf{a}}_i(\mathbf{t}), \tilde{\mathbf{a}}_j(\mathbf{t})\rangle$, multiply it entrywise by $\mathbf{s}_i(\mathbf{t})\mathbf{s}_j(\mathbf{t})$, and entrywise square it, in $O(n^2)$ time. The former matrix is $\mathbf{A}\mathbf{Z}(\mathbf{t})^{-1}\mathbf{A}^\top$, which takes time $O(n^2 d^{\omega-2})$ time to compute by multiplying $n \times d$, $d \times d$, and $d \times n$ matrices.

For the latter claim, we can again first compute

$$\mathbf{diag}\left(\left\{\mathrm{Tr}\left(\mathbf{M}_i(\mathbf{t})\right)\right\}_{i\in[n]}\right)\mathbf{v}$$

in time $O(nd^{\omega-1})$ using the steps described above. To implement $\nabla^2 f(\mathbf{t})\mathbf{v}$, it remains to compute

$$\left\langle \mathbf{M}_i(\mathbf{t}), \sum_{j\in[n]} \mathbf{v}_j \mathbf{M}_j(\mathbf{t}) \right\rangle = \exp(\mathbf{t}_i)\left(\mathbf{R}(\mathbf{t})\left(\sum_{j\in[n]} \mathbf{v}_j \mathbf{M}_j(\mathbf{t})\right)\mathbf{R}(\mathbf{t})\right)[\mathbf{a}_i,\mathbf{a}_i]$$

for all $i \in [n]$. Observe that

$$\mathbf{C} := \mathbf{R}(\mathbf{t})\left(\sum_{j\in[n]} \mathbf{v}_j \mathbf{M}_j(\mathbf{t})\right)\mathbf{R}(\mathbf{t}) = \mathbf{Z}(\mathbf{t})^{-1}\left(\sum_{j\in[n]} \mathbf{v}_j \exp(\mathbf{t}_j)\mathbf{a}_j\mathbf{a}_j^\top\right)\mathbf{Z}(\mathbf{t})^{-1},$$

which can be computed in $O(nd^{\omega-1})$ time by first forming the middle matrix on the right-hand side via multiplying $d \times n$ and $n \times d$ matrices. Finally, to compute all $\mathbf{C}[\mathbf{a}_i, \mathbf{a}_i]$, we can take the rows of $\mathbf{AC}$ and obtain their dot products with rows of $\mathbf{A}$ which requires $O(nd^{\omega-1})$ time to compute. □

To capitalize on the faster matrix-vector access given by Lemma 3.4, we will apply Theorem 3.2 to sparsify the Hessian of a regularized variant of Barthe's objective. Next, we require a tool from [CPW21] for optimizing box-constrained quadratics in a graph Laplacian.

Proposition 3.6 (Theorem 1.1, [CPW21]). *Let $\delta \in (0,1)$, let $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ have $\mathbf{l} \le \mathbf{u}$ entrywise, let $\mathbf{b}, \mathbf{t} \in \mathbb{R}^n$, and let $\mathbf{L} \in \mathbb{S}^{n\times n}_{\succeq \mathbf{0}}$ be a graph Laplacian with $\mathrm{nnz}(\mathbf{L}) \le m$. Let*

$$\mathcal{W} := \{\mathbf{w} \in \mathbb{R}^n \mid \mathbf{l}_i \le \mathbf{t}_i + \mathbf{w}_i \le \mathbf{r}_i \text{ for all } i \in [n]\}.$$

There is an algorithm $\mathcal{O}(\mathbf{L}, \mathbf{b}, \mathcal{W})$ that runs in time $O((n+m)^{1+o(1)} \log(\frac{1}{\delta}))$, and with probability $\geq 1-\delta$, it returns $\mathbf{v} \in \mathcal{W}$ satisfying

$$\langle \mathbf{b}, \mathbf{v} \rangle + \frac{1}{2}\mathbf{L}[\mathbf{v}, \mathbf{v}] \leq \frac{1}{2} \min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{b}, \mathbf{w} \rangle + \frac{1}{2}\mathbf{L}[\mathbf{w}, \mathbf{w}] \right\}.$$

In [CPW21] it was only stated for the case $\mathbf{l} = \mathbf{0}_n$ and $\mathbf{u}$ is ∞ in each coordinate, i.e., the box constraint is simply the positive orthant $\mathbb{R}^n_{\geq 0}$. However, the techniques extend straightforwardly to general box constraints [CPW25].

We now show how to use Proposition 3.6 to efficiently optimize an ℓ_∞-Hessian stable function. The following proof is based on Theorem 3.4, [CMTV17], but adapts it to tolerate multiplicative error in the Hessian computation. We note that a similar multiplicatively-robust generalization appeared earlier as Lemma 20, [AJJ+22], but was too restrictive for our purposes.

Lemma 3.6. *Let convex $F : \mathbb{R}^n \to \mathbb{R}$ be $(1,2)$-Hessian stable with respect to $\|\cdot\|_\infty$, and let $\mathbf{t}^\star \in \arg\min_{\mathbf{t} \in \mathbb{R}^n} F(\mathbf{t})$ have $\|\mathbf{t}^\star\|_\infty \leq \log(\kappa)$. For $\mathbf{t} \in \mathbb{R}^n$, $\alpha \geq 1$, let $\widetilde{\mathbf{L}} \in \mathbb{S}^{n \times n}_{\succeq \mathbf{0}}$ be a graph Laplacian with*

$$\nabla^2 F(\mathbf{t}) \preceq \widetilde{\mathbf{L}} \preceq \alpha \nabla^2 F(\mathbf{t}).$$

Then for any $\mathbf{t} \in \mathbb{B}_\infty(\log(\kappa))$, if $\mathbf{t}' \leftarrow \mathbf{t} + \mathcal{O}(8\widetilde{\mathbf{L}}, \nabla F(\mathbf{t}), \mathbb{B}_\infty(\mathbf{t}, 1) \cap \mathbb{B}_\infty(\log(\kappa)))$, where $\mathcal{O}$ is as in Proposition 3.6, we have

$$F(\mathbf{t}') - F(\mathbf{t}^\star) \leq \left(1 - \frac{1}{240\alpha \log(\kappa)}\right)(F(\mathbf{t}) - F(\mathbf{t}^\star)).$$

Proof. For any $\mathbf{u}$ with $\|\mathbf{t} - \mathbf{u}\|_\infty \leq 1$, Hessian stability of F yields the bounds

$$\begin{aligned}
F(\mathbf{u}) - F(\mathbf{t}) - \langle \nabla F(\mathbf{t}), \mathbf{u} - \mathbf{t} \rangle &= \int_0^1 (1-\lambda) \nabla^2 F((1-\lambda)\mathbf{t} + \lambda \mathbf{u})[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}] \, \mathrm{d}\lambda \\
&\leq \int_0^1 (1-\lambda) e^2 \widetilde{\mathbf{L}}[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}] \, \mathrm{d}\lambda \\
&\leq \frac{e^2}{2} \widetilde{\mathbf{L}}[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}] \leq 4\widetilde{\mathbf{L}}[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}], \\
F(\mathbf{u}) - F(\mathbf{t}) - \langle \nabla F(\mathbf{t}), \mathbf{u} - \mathbf{t} \rangle &\geq \frac{1}{2\alpha e^2} \widetilde{\mathbf{L}}[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}] \geq \frac{1}{15\alpha} \widetilde{\mathbf{L}}[\mathbf{u} - \mathbf{t}, \mathbf{u} - \mathbf{t}].
\end{aligned} \tag{3.20}$$

Next define

$$\hat{\boldsymbol{\delta}} := \underset{\substack{\|\boldsymbol{\delta}\|_\infty \le 1 \\ \mathbf{t}+\boldsymbol{\delta} \in \mathbb{B}_\infty(\log(\kappa))}}{\arg\min} \quad \langle \nabla F(\mathbf{t}), \boldsymbol{\delta} \rangle + 4\widetilde{\mathbf{L}}\left[\boldsymbol{\delta}, \boldsymbol{\delta}\right],$$

and observe that for $\boldsymbol{\delta} := \mathbf{t}' - \mathbf{t} = \mathcal{O}(8\widetilde{\mathbf{L}}, \nabla F(\mathbf{t}), \mathbb{B}_\infty(\mathbf{t}, 1) \cap \mathbb{B}_\infty(\log(\kappa)))$, we have

$$\begin{aligned} \langle \nabla F(\mathbf{t}), \boldsymbol{\delta} \rangle + 4\widetilde{\mathbf{L}}\left[\boldsymbol{\delta}, \boldsymbol{\delta}\right] &\le \frac{1}{2}\left(\langle \nabla F(\mathbf{t}), \hat{\boldsymbol{\delta}} \rangle + 4\widetilde{\mathbf{L}}[\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}]\right) \\ &\le \frac{1}{2}\left(\langle \nabla F(\mathbf{t}), \boldsymbol{\delta}^\star \rangle + 4\widetilde{\mathbf{L}}[\boldsymbol{\delta}^\star, \boldsymbol{\delta}^\star]\right), \\ \text{for any } \|\boldsymbol{\delta}^\star\|_\infty &\le 1 \text{ with } \mathbf{t} + \boldsymbol{\delta}^\star \in \mathbb{B}_\infty(\log(\kappa)), \end{aligned} \tag{3.21}$$

from the oracle guarantee and definition of $\hat{\boldsymbol{\delta}}$. Hence, applying the upper bounds in (3.20) and (3.21),

$$\begin{aligned} F\left(\mathbf{t}'\right) &\le F\left(\mathbf{t}\right) + \langle \nabla F\left(\mathbf{t}\right), \boldsymbol{\delta} \rangle + 4\widetilde{\mathbf{L}}\left[\boldsymbol{\delta}, \boldsymbol{\delta}\right] \\ &\le F\left(\mathbf{t}\right) + \frac{1}{2}\left(\langle \nabla F(\mathbf{t}), \boldsymbol{\delta}^\star \rangle + 4\widetilde{\mathbf{L}}[\boldsymbol{\delta}^\star, \boldsymbol{\delta}^\star]\right), \end{aligned} \tag{3.22}$$

for our choice of $\boldsymbol{\delta}^\star$ satisfying the bounds in (3.21). We choose $\boldsymbol{\delta}^\star = \frac{c}{2\log(\kappa)}(\mathbf{t}^\star - \mathbf{t})$ where $c = \frac{1}{60\alpha}$. First observe that this is a valid choice of movement, because

$$\begin{aligned} \mathbf{t} + \boldsymbol{\delta}^\star &= \left(1 - \frac{c}{2\log(\kappa)}\right)\mathbf{t} + \frac{c}{2\log(\kappa)}\mathbf{t}^\star \in \mathbb{B}_\infty\left(\log(\kappa)\right), \\ \|\boldsymbol{\delta}^\star\|_\infty &\le \frac{1}{2\log(\kappa)}\left(\|\mathbf{t}\|_\infty + \|\mathbf{t}^\star\|_\infty\right) \le \frac{2\log(\kappa)}{2\log(\kappa)} = 1, \end{aligned}$$

where both inequalities used that $\mathbf{t}, \mathbf{t}^\star \in \mathbb{B}_\infty(\log(\kappa))$, which is a convex set. Thus,

$$\begin{aligned} \frac{1}{2}\left(\langle \nabla F(\mathbf{t}), \boldsymbol{\delta}^\star \rangle + 4\widetilde{\mathbf{L}}[\boldsymbol{\delta}^\star, \boldsymbol{\delta}^\star]\right) &= \frac{1}{120\alpha}\left(\left\langle \nabla F(\mathbf{t}), \frac{1}{c}\boldsymbol{\delta}^\star \right\rangle + \frac{1}{15\alpha}\widetilde{\mathbf{L}}\left[\frac{1}{c}\boldsymbol{\delta}^\star, \frac{1}{c}\boldsymbol{\delta}^\star\right]\right) \\ &\le \frac{1}{120\alpha}\left(F\left(\mathbf{t} + \frac{1}{c}\boldsymbol{\delta}^\star\right) - F\left(\mathbf{t}\right)\right) \\ &\le -\frac{1}{240\alpha\log(\kappa)}\left(F\left(\mathbf{t}\right) - F\left(\mathbf{t}^\star\right)\right). \end{aligned}$$

The first inequality above used the lower bound in (3.20), and the second inequality used convexity of F. At this point, combining with (3.22) yields the conclusion. □

3.4.4 Proof of Theorem 3.1

In this section we put together the pieces we have built to obtain our final algorithm. To begin, we note that under Assumption 3.1, the following simple initial function error bound holds.

Lemma 3.7. *Under Assumption 3.1, letting* $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t}) \cap \mathbb{B}_\infty(\log(\kappa))$*, we have that*

$$f\left(\mathbf{0}_n\right) - f\left(\mathbf{t}^\star\right) \leq \frac{d\log^2(\kappa)}{2}.$$

Proof. We first note that for all $\mathbf{t}, \mathbf{v} \in \mathbb{R}^n$, we have

$$\nabla^2 f(\mathbf{t})\left[\mathbf{v},\mathbf{v}\right] \leq \mathbf{diag}\left(\left\{\mathrm{Tr}\left(\mathbf{M}_i(\mathbf{t})\right)\right\}_{i\in[n]}\right)\left[\mathbf{v},\mathbf{v}\right] = \sum_{i\in[n]} \boldsymbol{\tau}_i\left(\mathbf{S}(\mathbf{t})\mathbf{A}\right)\mathbf{v}_i^2 \leq d\left\|\mathbf{v}\right\|_\infty^2,$$

i.e., f is d-smooth with respect to $\left\|\cdot\right\|_\infty$ (we used Fact A.5 in the last inequality above). Thus by the second-order Taylor expansion from $\mathbf{t}^\star$ to $\mathbf{0}_n$, and using that $\nabla f(\mathbf{t}^\star) = \mathbf{0}_n$,

$$\begin{aligned} f(\mathbf{0}_n) &= f(\mathbf{t}^\star) + \int_0^1 (1-\lambda)\nabla^2 f\left((1-\lambda)\,\mathbf{t}^\star\right)\left[\mathbf{t}^\star,\mathbf{t}^\star\right]\mathrm{d}\lambda \\ &\leq f(\mathbf{t}^\star) + \frac{d}{2}\left\|\mathbf{t}^\star\right\|_\infty^2 \leq f(\mathbf{t}^\star) + \frac{d\log^2(\kappa)}{2}. \end{aligned}$$

□

We can now prove Theorem 3.1, the main result of this section.

Theorem 3.1. *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$*,* $\mathbf{c} \in (0,1]^n$ *satisfy Assumption 3.1, and let* $\delta, \epsilon \in (0,1)$*. There is an algorithm that computes* $\mathbf{R}$*, a* $(\mathbf{c},\epsilon)$*-Forster transform of* $\mathbf{A}$*, with probability* $\geq 1-\delta$*, in time*

$$O\left(nd^{\omega-1}\log\left(\kappa\right)\left(\frac{n\log(\kappa)}{\delta\epsilon\mathbf{c}_{\min}}\right)^{o(1)}\right), \text{ where } \mathbf{c}_{\min} := \min_{i\in[n]}\mathbf{c}_i.$$

Proof. Throughout this proof, we let f be Barthe's objective, and

$$F(\mathbf{t}) := f(\mathbf{t}) + \frac{\epsilon^2 \mathbf{c}_{\min}^2}{4n \log^2(\kappa)} \mathbf{t}^\top \mathbf{\Pi} \mathbf{t},$$

where $\mathbf{\Pi} := \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top$. Our goal is to optimize F to error $\frac{\epsilon^2 \mathbf{c}_{\min}^2}{4}$ over $\mathbb{B}_\infty(\log(\kappa))$. To see why this suffices, note that for all $\mathbf{t} \in \mathbb{B}_\infty(\log(\kappa))$, we have $\mathbf{t}^\top \mathbf{\Pi} \mathbf{t} \le \|\mathbf{t}\|_2^2 \le n \log^2(\kappa)$, and hence any $\frac{\epsilon^2 \mathbf{c}_{\min}^2}{4}$-minimizer $\mathbf{t}$ of F over $\mathbb{B}_\infty(\log(\kappa))$ satisfies

$$f(\mathbf{t}) - f(\mathbf{t}^\star) \le F(\mathbf{t}) - F(\mathbf{t}^\star) + \frac{\epsilon^2 \mathbf{c}_{\min}^2}{4} \le \frac{\epsilon^2 \mathbf{c}_{\min}^2}{2},$$

where $\mathbf{t}^\star$ is as in Lemma 3.7. Now, applying Lemma 3.3 gives the claim, because computing $\mathbf{R}(\mathbf{t})$ does not dominate the stated runtime (as described in Lemma 3.5). Note that each time we apply Lemma 3.6 with $\alpha \leftarrow (\frac{n \log(\kappa)}{\epsilon \mathbf{c}_{\min}})^{o(1)}$, we improve the function error by a multiplicative $1 - \Omega((\alpha \log(\kappa))^{-1})$. Moreover, the initial function error is bounded as in Lemma 3.7. Thus to obtain the stated runtime, it is enough to show how to implement each call to Lemma 3.6 with $\alpha \leftarrow (\frac{n \log(\kappa)}{\epsilon \mathbf{c}_{\min}})^{o(1)}$, in time

$$O\left(n d^{\omega-1} \left(\frac{n \log(\kappa)}{\delta \epsilon \mathbf{c}_{\min}}\right)^{o(1)}\right) \tag{3.23}$$

Adjusting δ by the number of calls and taking a union bound then gives the claim.

To achieve this runtime we first produce a sparse graph Laplacian matrix $\widetilde{\mathbf{L}}$ satisfying (3.16) for $\Delta := \frac{\epsilon^2 \mathbf{c}_{\min}^2}{4 \log^2(\kappa)}$ and $\mathbf{L} \leftarrow \nabla^2 f(\mathbf{t})$ for some iterate $\mathbf{t}$. Recalling that $\mathrm{Tr}(\mathbf{L}) \le \mathrm{Tr}(\mathbf{I}_d) = d$, Theorem 3.2 and Lemma 3.5 guarantee that we can compute such a $\widetilde{\mathbf{L}}$ with probability $\ge 1 - \delta$ within time (3.23). Given $\widetilde{\mathbf{L}}$, the per-iteration runtime follows from Proposition 3.6, which does not dominate. □

Remark 3.1. For highly-accurate solutions or extremely small failure probabilities (i.e., δ, ϵ smaller than an inverse polynomial in n), the subpolynomial dependences on $\frac{1}{\delta}$, $\frac{1}{\epsilon}$ in Theorem 3.1 could be dominant factors. However, these subpolynomial factors only arise due to the use of Theorem 3.2 to sparsely approximate Hessians of

Barthe's objective. If we instead directly compute the Hessians via Lemma 3.5, then slightly modifying the proof of Theorem 3.1 yields an alternate runtime of

$$O\left(n^2 d^{\omega-2} \log(\kappa) \operatorname{polylog}\left(\frac{n \log(\kappa)}{\delta \epsilon \mathbf{c}_{\min}}\right)\right).$$

This runtime gives a worse dependence on n, but improves the dependences on other parameters (i.e., $\frac{1}{\delta}, \frac{1}{\epsilon}, \frac{1}{\mathbf{c}_{\min}}$) from subpolynomial to polylogarithmic.

3.5 Conditioning of smoothed matrices

In this section, we provide an ℓ_∞ diameter bound for a minimizing vector of Barthe's objective, when computing a Forster transform of a smoothed matrix of the form

$$\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{G}, \text{ where } \mathbf{A} \in \mathbb{R}^{n \times d} \text{ and } \mathbf{G} \in \mathbb{R}^{n \times d} \text{ has entries } \sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma^2).$$

We first define a notion of deepness in Section 3.5.1 and derive a diameter bound for deep vectors, patterned off [AKS20]. In Sections 3.5.2 and 3.5.3, we show tail bounds for the singular values of a smoothed matrix. We combine these results to prove Theorem 3.3 in Section 3.5.4, our main result on the conditioning of Forster transforms of smoothed matrices.

Throughout this section, we follow notation (3.10), (3.11), (3.12) from Section 3.1 and make the following simplifying, and somewhat mild, assumption on the relationship between n and d.

Assumption 3.2. *$n \geq Cd$ for some constant $C > 1$.*

We mention that our result in Theorem 3.3 is stated for the case of $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$, which is the most interesting setting we are aware of in typical applications. However, we discuss the case of general $\mathbf{c}$ in Section 3.5.5, where our techniques readily apply under a strengthening of Assumption 3.2.

3.5.1 Diameter bound for deep vectors

Our strategy for our diameter bound follows an analysis in [AKS20], based on the assumption that $\mathbf{c}$ lies nontrivially inside the interior of the basis polytope (cf. Proposition 3.2).

We start by extending Definition 1.4 and proving a stronger version of Lemma 4.4 from [AKS20].

Definition 3.6 (Deepness)**.** Let $\mathbf{c} \in \mathbb{R}^n_{>0}$ satisfy $\|\mathbf{c}\|_1 = d$, $\eta \in [0,1]$, and $\Delta \geq 0$. We say that $\mathbf{c}$ lies (η, Δ)-*deep* inside the basis polytope of $\{\mathbf{a}_i\}_{i\in[n]} \subset \mathbb{R}^d$ if for all $k \in [d-1]$ and subspaces $E \subseteq \mathbb{R}^d$ with dimension k,

$$\sum_{i\in[n]} \mathbf{c}_i \mathbb{I}_{\|\mathbf{a}_i - \mathbf{\Pi}_E \mathbf{a}_i\|_2 \leq \Delta} \leq (1-\eta)k.$$

When $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$, this is equivalent to the following: for all $k \in [d-1]$ and subspaces E with dimension k, at most $\frac{(1-\eta)kn}{d}$ of the $\mathbf{a}_i$ satisfy $\|\mathbf{a}_i - \mathbf{\Pi}_E \mathbf{a}_i\|_2 \leq \Delta$.

We remark that every vector in the basis polytope is $(0,0)$-deep by Proposition 3.2. Furthermore, increasing Δ potentially increases the number of $\mathbf{c}_i$ considered in the sum, and increasing η tightens the inequality, so (η, Δ)-deepness becomes a stronger condition for larger values of η and Δ.

We now show, following [AKS20], that deepness in the basis polytope implies a diameter bound.

Lemma 3.8. *Let $\mu, M, \eta, \Delta > 0$ and $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rows $\{\mathbf{a}_i\}_{i\in[n]}$ satisfying $\mu \leq \|\mathbf{a}_i\|_2^2 \leq M$ for all $i \in [n]$. If $\mathbf{c} \in \mathbb{R}^n_{>0}$ has minimum entry $\mathbf{c}_{\min}$ and lies (η, Δ)-deep inside the basis polytope of $\{\mathbf{a}_i\}_{i\in[n]}$, then there exists $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$ satisfying*

$$\|\mathbf{t}^\star\|_\infty \leq \frac{1}{2}\log\left(\frac{M}{\mu \mathbf{c}_{\min}}\left(\frac{4M}{\eta\Delta^2}\right)^{d-1}\right).$$

Proof. Let $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$ such that $\min_{i\in[n]} \mathbf{t}^\star_i = 0$, which exists by Proposition 3.2, Proposition 3.3, and (3.15). Let $\boldsymbol{\lambda}_1 \geq \boldsymbol{\lambda}_2 \geq \cdots \geq \boldsymbol{\lambda}_d$ be the eigenvalues of $\mathbf{R} := \mathbf{R}(\mathbf{t}^\star)$.

Fix $k \in [d-1]$. Let E be a k-dimensional subspace spanned by eigenvectors of $\mathbf{R}$ with eigenvalues $\boldsymbol{\lambda}_d, \boldsymbol{\lambda}_{d-1}, \ldots, \boldsymbol{\lambda}_{d-k+1}$. By Proposition 3.3, $\mathbf{R}$ is a $\mathbf{c}$-Forster transform of $\mathbf{A}$, so

$$\sum_{i\in[n]} \mathbf{c}_i \cdot \frac{(\mathbf{R}\mathbf{a}_i)(\mathbf{R}\mathbf{a}_i)^\top}{\|\mathbf{R}\mathbf{a}_i\|_2^2} = \mathbf{I}_d.$$

Projecting onto $E^\perp$ and taking a trace of both sides,

$$\begin{aligned}\sum_{i\in[n]} \mathbf{c}_i \cdot \frac{\|\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i\|_2^2}{\|\mathbf{R}\mathbf{a}_i\|_2^2} &= \operatorname{Tr}\left(\sum_{i\in[n]} \mathbf{c}_i \cdot \frac{(\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i)(\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i)^\top}{\|\mathbf{R}\mathbf{a}_i\|_2^2}\right) \\ &= \operatorname{Tr}(\boldsymbol{\Pi}_{E^\perp}) = d-k.\end{aligned} \tag{3.24}$$

Now, consider the $\mathbf{a}_i$ such that $\|\mathbf{a}_i - \boldsymbol{\Pi}_E\mathbf{a}_i\|_2 > \Delta$, and decompose these $\mathbf{a}_i$ as $\mathbf{a}_i = \mathbf{y}_i + \mathbf{z}_i$, where $\mathbf{y}_i \in E$ and $\mathbf{z}_i \in E^\perp$. Then $\|\mathbf{z}_i\|_2 > \Delta$ and $\|\mathbf{y}_i\|_2 < \sqrt{\|\mathbf{a}_i\|_2^2 - \Delta^2} \le \sqrt{M-\Delta^2}$. Since E and $E^\perp$ are both spanned by eigenvectors of $\mathbf{R}$, $\mathbf{R}\mathbf{y}_i$ and $\mathbf{R}\mathbf{z}_i$ are orthogonal, and $\|\mathbf{R}\mathbf{a}_i\|_2^2 = \|\mathbf{R}\mathbf{y}_i\|_2^2 + \|\mathbf{R}\mathbf{z}_i\|_2^2$. Furthermore, since E is spanned by eigenvectors with eigenvalues at most $\boldsymbol{\lambda}_{d-k+1}$ and $E^\perp$ is spanned by eigenvectors with eigenvalues at least $\boldsymbol{\lambda}_{d-k}$,

$$\begin{aligned}\|\mathbf{R}\mathbf{y}_i\|_2 \le \boldsymbol{\lambda}_{d-k+1}\|\mathbf{y}_i\|_2 \le \boldsymbol{\lambda}_{d-k+1}\sqrt{M-\Delta^2} \\ \text{and } \|\mathbf{R}\mathbf{a}_i\|_2 \ge \|\mathbf{R}\mathbf{z}_i\|_2 \ge \boldsymbol{\lambda}_{d-k}\|\mathbf{z}_i\|_2 \ge \boldsymbol{\lambda}_{d-k}\Delta.\end{aligned}$$

It follows that

$$\begin{aligned}\frac{\|\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i\|_2^2}{\|\mathbf{R}\mathbf{a}_i\|_2^2} &= \left\|\boldsymbol{\Pi}_{E^\perp}\frac{\mathbf{R}\mathbf{a}_i}{\|\mathbf{R}\mathbf{a}_i\|_2}\right\|_2^2 = 1 - \left\|\boldsymbol{\Pi}_E\frac{\mathbf{R}\mathbf{a}_i}{\|\mathbf{R}\mathbf{a}_i\|_2}\right\|_2^2 \\ &= 1 - \frac{\|\mathbf{R}\mathbf{y}_i\|_2^2}{\|\mathbf{R}\mathbf{a}_i\|_2^2} \ge 1 - \frac{\boldsymbol{\lambda}_{d-k+1}^2(M-\Delta^2)}{\boldsymbol{\lambda}_{d-k}^2\Delta^2}.\end{aligned}$$

Combining this with (3.24) and the (η,Δ)-deepness of $\mathbf{c}$,

$$\begin{aligned}d-k = \sum_{i\in[n]} \mathbf{c}_i \cdot \frac{\|\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i\|_2^2}{\|\mathbf{R}\mathbf{a}_i\|_2^2} &\ge \sum_{i\in[n]} \mathbf{c}_i \mathbb{I}_{\|\mathbf{a}_i - \boldsymbol{\Pi}_E\mathbf{a}_i\|_2 > \Delta} \cdot \frac{\|\boldsymbol{\Pi}_{E^\perp}\mathbf{R}\mathbf{a}_i\|_2^2}{\|\mathbf{R}\mathbf{a}_i\|_2^2} \\ &\ge \left(1 - \frac{\boldsymbol{\lambda}_{d-k+1}^2(M-\Delta^2)}{\boldsymbol{\lambda}_{d-k}^2\Delta^2}\right)\sum_{i\in[n]} \mathbf{c}_i \mathbb{I}_{\|\mathbf{a}_i - \boldsymbol{\Pi}_E\mathbf{a}_i\|_2 > \Delta} \\ &\ge \left(1 - \frac{\boldsymbol{\lambda}_{d-k+1}^2(M-\Delta^2)}{\boldsymbol{\lambda}_{d-k}^2\Delta^2}\right)(d-(1-\eta)k),\end{aligned}$$

which rearranges to

$$\frac{\boldsymbol{\lambda}_{d-k}}{\boldsymbol{\lambda}_{d-k+1}} \leq \left(\frac{d-k+\eta k}{\eta k} \cdot \frac{M-\Delta^2}{\Delta^2}\right)^{\frac{1}{2}}. \tag{3.25}$$

Since (3.25) holds for all $k \in [d-1]$,

$$\begin{aligned}
\frac{\boldsymbol{\lambda}_1}{\boldsymbol{\lambda}_d} &= \prod_{k\in[d-1]} \frac{\boldsymbol{\lambda}_{d-k}}{\boldsymbol{\lambda}_{d-k+1}} \leq \prod_{k\in[d-1]} \left(\frac{d-k+\eta k}{\eta k} \cdot \frac{M-\Delta^2}{\Delta^2}\right)^{\frac{1}{2}} \\
&= \left(\frac{M-\Delta^2}{\Delta^2}\right)^{\frac{d-1}{2}} \prod_{k\in[d-1]} \left(\frac{d-k+\eta k}{\eta k}\right)^{\frac{1}{2}} \\
&= \left(\frac{M-\Delta^2}{\Delta^2}\right)^{\frac{d-1}{2}} \prod_{k=1}^{\lfloor \frac{d}{1+\eta} \rfloor} \left(\frac{d-k+\eta k}{\eta k}\right)^{\frac{1}{2}} \prod_{k=\lfloor \frac{d}{1+\eta} \rfloor+1}^{d-1} \left(\frac{d-k+\eta k}{\eta k}\right)^{\frac{1}{2}} \\
&\leq \left(\frac{M-\Delta^2}{\Delta^2}\right)^{\frac{d-1}{2}} \prod_{k=1}^{\lfloor \frac{d}{1+\eta} \rfloor} \left(\frac{2(d-k)}{\eta k}\right)^{\frac{1}{2}} \prod_{k=\lfloor \frac{d}{1+\eta} \rfloor+1}^{d-1} \left(\frac{2}{\eta}\right)^{\frac{1}{2}} \\
&= \left(\frac{2(M-\Delta^2)}{\eta\Delta^2}\right)^{\frac{d-1}{2}} \prod_{k=1}^{\lfloor \frac{d}{1+\eta} \rfloor} \left(\frac{d-k}{k}\right)^{\frac{1}{2}} = \left(\frac{2(M-\Delta)^2}{\eta\Delta^2}\right)^{\frac{d-1}{2}} \left(\frac{d-1}{\lfloor \frac{d}{1+\eta} \rfloor}\right)^{\frac{1}{2}} \\
&\leq \left(\frac{4(M-\Delta^2)}{\eta\Delta^2}\right)^{\frac{d-1}{2}} \leq \left(\frac{4M}{\eta\Delta^2}\right)^{\frac{d-1}{2}},
\end{aligned} \tag{3.26}$$

where the fourth line uses $\frac{d-k+\eta k}{\eta k} \leq \frac{2(d-k)}{\eta k}$ for $k \leq \frac{d}{1+\eta}$ and $\frac{d-k+\eta k}{\eta k} \leq 2 \leq \frac{2}{\eta}$ for $k \geq \frac{d}{1+\eta}$ and the sixth line uses $\binom{a}{b} \leq 2^a$.

Let $j \in [n]$ such that $\mathbf{t}^\star_j = \|\mathbf{t}^\star\|_\infty$, and note that the largest eigenvalue of $\mathbf{Z}(\mathbf{t}^\star)$ is $\frac{1}{\boldsymbol{\lambda}_d^2}$. Thus

$$\begin{aligned}
\frac{1}{\boldsymbol{\lambda}_d^2} &= \max_{\|\mathbf{x}\|_2=1} \sum_{i\in[n]} \exp(\mathbf{t}^\star_i) \langle \mathbf{a}_i, \mathbf{x}\rangle^2 \geq \max_{\|\mathbf{x}\|_2=1} \exp(\mathbf{t}^\star_j) \langle \mathbf{a}_j, \mathbf{x}\rangle^2 \\
&= \|\mathbf{a}_j\|_2^2 \exp(\mathbf{t}^\star_j) \geq \mu \exp(\|\mathbf{t}^\star\|_\infty).
\end{aligned} \tag{3.27}$$

Let $\ell \in [n]$ such that $\mathbf{t}^\star_\ell = 0$. Since $\mathbf{t}^\star$ minimizes f, we have

$$\mathbf{c}_\ell = \mathrm{Tr}(\mathbf{M}_\ell(\mathbf{t}^\star)) = \|\mathbf{R}\mathbf{a}_\ell\|_2^2 \leq \boldsymbol{\lambda}_1^2 \|\mathbf{a}_\ell\|_2^2 \leq M\boldsymbol{\lambda}_1^2 \tag{3.28}$$

by Fact 3.1. Combining (3.27) and (3.28) gives

$$\frac{\boldsymbol{\lambda}_1^2}{\boldsymbol{\lambda}_d^2} \geq \frac{\mu \mathbf{c}_{\min}}{M} \exp(\|\mathbf{t}^\star\|_\infty),$$

and combining with (3.26) and rearranging gives

$$\|\mathbf{t}^\star\|_\infty \leq \log\left(\frac{M}{\mu \mathbf{c}_{\min}}\left(\frac{4M}{\eta\Delta^2}\right)^{d-1}\right).$$

Since f is invariant to translations by $\mathbf{1}_n$ and $\min_{i\in[n]} \mathbf{t}_i^\star = 0$ by assumption, we can shift $\mathbf{t}^\star$ to obtain a minimizer that has extreme coordinates averaging to 0, which gives the result. □

With Lemma 3.8 in hand, we must show $\frac{M}{\mu}$ and $\frac{1}{\Delta}$ are polynomially bounded for an appropriate choice of η, in the smoothed setting. The bulk of our remaining analysis establishes this result.

Remark 3.2. Our strategy in this section is to restrict η to be a constant, e.g., $\eta = 0.1$, in which case our goal is to show that at most $\frac{0.9k}{d}n$ of the vectors in $\{\mathbf{a}_i\}_{i\in[n]}$ lie close to any k-dimensional subspace. If, for instance, $\frac{n}{d} < 1.1$, this is clearly impossible, because choosing $k = 1$ implies that not even a single vector lies close to any 1-dimensional subspace, which is false just by taking $\mathrm{Span}(\mathbf{a}_i)$ for any $i \in [n]$. Thus, the restriction $\frac{n}{d} \geq 1.1$ (or more generally, a constant bounded away from 1) is somewhat inherent in the regime $\eta = \Omega(1)$. It is possible that our strategy can be modified to extend to even smaller n, but for simplicity, we focus on the setting of Assumption 3.2.

To frame the rest of the section, we provide a helper lemma relating the condition in Definition 3.6 to appropriate submatrices having small singular values.

Lemma 3.9. *Let $\mathbf{A} = \{\mathbf{a}_i\}_{i\in[m]} \in \mathbb{R}^{d\times m}$, $k < m$, and $\Delta > 0$. Suppose there exists a k-dimensional subspace E of $\mathbb{R}^d$ such that $\|\mathbf{a}_i - \boldsymbol{\Pi}_E \mathbf{a}_i\|_2 \leq \Delta$ for all $i \in [m]$. Then $\boldsymbol{\sigma}_{k+1}(\mathbf{A}) \leq \sqrt{m}\Delta$.*

Proof. Let $\mathbf{V} \in \mathbb{R}^{d\times(d-k)}$ have columns that form an orthonormal basis for $E^\perp$. By assumption,

$$\left\|\mathbf{V}\mathbf{V}^\top \mathbf{a}_i\right\|_2^2 = \left\|\mathbf{V}^\top \mathbf{a}_i\right\|_2^2 \le \Delta^2 \text{ for all } i \in [m].$$

By the min-max theorem,

$$\begin{aligned}\boldsymbol{\sigma}_{k+1}(\mathbf{A}) &= \min_{\substack{E\subseteq\mathbb{R}^d \\ \dim(E)=d-k}} \max_{\substack{\mathbf{x}\in E \\ \|\mathbf{x}\|_2=1}} \left\|\mathbf{A}^\top \mathbf{x}\right\|_2 \\ &\le \max_{\substack{\mathbf{x}\in \mathrm{Span}(\mathbf{V}) \\ \|\mathbf{x}\|_2=1}} \left\|\mathbf{A}^\top \mathbf{x}\right\|_2 = \max_{\substack{\mathbf{v}\in\mathbb{R}^{d-k} \\ \|\mathbf{v}\|_2=1}} \left\|\mathbf{A}^\top \mathbf{V}\mathbf{v}\right\|_2 .\end{aligned}$$

Observe that $\mathbf{A}^\top\mathbf{V}$ has rows $\{\mathbf{V}^\top\mathbf{a}_i\}_{i\in[m]}$, so by the Cauchy–Schwarz inequality,

$$\left\|\mathbf{A}^\top \mathbf{V}\mathbf{v}\right\|_2^2 = \sum_{i\in[m]} \left\langle \mathbf{V}^\top\mathbf{a}_i, \mathbf{v}\right\rangle^2 \le \sum_{i\in[m]} \Delta^2 = m\Delta^2$$

for all $\mathbf{v} \in \mathbb{R}^{d-k}$ with $\|\mathbf{v}\|_2 = 1$. It follows that $\boldsymbol{\sigma}_{k+1}(\mathbf{A}) \le \sqrt{m}\Delta$. □

In Sections 3.5.2 and 3.5.3, we show that with high probability, the conclusion of Lemma 3.9 is violated for all appropriately-sized smoothed submatrices. This shows that the premise of Lemma 3.9 is also violated, which we establish in Section 3.5.4, so that $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$ is indeed (η, Δ)-deep.

3.5.2 Conditioning of wide and near-square smoothed matrices

Throughout this section, we let

$$\eta := 1 - \frac{1}{\sqrt{C}},\ K := \sqrt[3]{C},\ k \in [d-1],\ \text{and } m := \left\lceil \frac{(1-\eta)kn}{d} \right\rceil .$$

Remark 3.3. We note that these definitions imply $0 < \eta < 1 - \frac{1}{C}$ and $1 < K < (1-\eta)C$. Indeed, nothing in our analysis relies on our choices of η and K other than the fact that they are constants that satisfy these inequalities, which limit our analysis in the following places.

- In Lemma 3.10, we require $(1-\eta)C > 1$ (equivalently, $\eta < 1 - \frac{1}{C}$) so that $m - k = \Omega(m)$.

- In Lemma 3.11, we require $K < (1-\eta)C$ so that $d - k = \Omega(d)$.
- In Lemma 3.11 and Lemma 3.16, we require $K > 1$, and in Theorem 3.3, we require $\eta > 0$.

We first use the following results from the literature to establish tail bounds for the singular values of smoothed matrices in the cases $m \le d$ and $d < m \le Kd$, i.e., m that are at most a constant factor larger than d. In Section 3.5.3, we use a different argument to handle $m > Kd$.

Fact 3.2 (Theorem 1.2, [Sza91])**.** *Let $\mathbf{G} \in \mathbb{R}^{d\times d}$ have entries $\sim_{\text{i.i.d.}} \mathcal{N}(0,1)$. Then for all $j \in [d]$,*

$$\Pr\left[\boldsymbol{\sigma}_{d-j+1}(\mathbf{G}) < \frac{\alpha j}{\sqrt{d}}\right] \le \left(\sqrt{2e}\alpha\right)^{j^2}.$$

Fact 3.3 (Theorem 2.4, [BKMS21])**.** *Let $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d\times d}$ such that $\boldsymbol{\sigma}_i(\mathbf{M}) \ge \boldsymbol{\sigma}_i(\mathbf{N})$ for all $i \in [d]$. Then for every $t \ge 0$, there exists a joint distribution on pairs of matrices $(\mathbf{G}, \mathbf{H}) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d}$ such that the marginals $\mathbf{G}$ and $\mathbf{H}$ have entries $\sim_{\text{i.i.d.}} \mathcal{N}(0,1)$ and*

$$\Pr[\boldsymbol{\sigma}_i(\mathbf{M} + t\mathbf{G}) \ge \boldsymbol{\sigma}_i(\mathbf{N} + t\mathbf{H})] = 1 \text{ for all } i \in [d].$$

We note that these results imply tail bounds for the singular values of square smoothed matrices: given $\sigma > 0$, $\mathbf{A} \in \mathbb{R}^{d\times d}$, and $\mathbf{G} \in \mathbb{R}^{d\times d}$ with entries $\sim_{\text{i.i.d.}} \mathcal{N}(0,1)$, we let $\mathbf{M} \leftarrow \mathbf{A}$, $\mathbf{N} \leftarrow \mathbf{0}$, $t \leftarrow \sigma$, and $(\mathbf{G}, \mathbf{H})$ have the distribution in Fact 3.3. Then by Fact 3.3 and Fact 3.2,

$$\begin{aligned}\Pr\left[\boldsymbol{\sigma}_{d-j+1}(\mathbf{A} + \sigma\mathbf{G}) < \frac{\alpha\sigma j}{\sqrt{d}}\right] &\le \Pr\left[\boldsymbol{\sigma}_{d-j+1}(\sigma\mathbf{H}) < \frac{\alpha\sigma j}{\sqrt{d}}\right] \\ &= \Pr\left[\boldsymbol{\sigma}_{d-j+1}(\mathbf{H}) < \frac{\alpha j}{\sqrt{d}}\right] \le \left(\sqrt{2e}\alpha\right)^{j^2}.\end{aligned} \tag{3.29}$$

Lemma 3.10. *Under Assumption 3.2, let $\delta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$, $\eta := 1 - \frac{1}{\sqrt{C}}$, $k \in [d-1]$, $m := \lceil \frac{(1-\eta)kn}{d} \rceil$, $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{G} \in \mathbb{R}^{n\times d}$ have entries $\sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma^2)$, $\widetilde{\mathbf{A}} := \mathbf{A} + \mathbf{G}$, and suppose $m \le d$. Then for any $S \subseteq [n]$ with $|S| = m$,*

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \le \sqrt{m}\Delta\right] \le \frac{\delta}{(d-1)\binom{n}{m}}, \text{ where } \Delta = \left(\frac{\delta\sigma}{n}\right)^{O(1)}.$$

Proof. Remove any $d-m$ columns of $\widetilde{\mathbf{A}}_{S:}$ to obtain $\mathbf{B} \in \mathbb{R}^{m\times m}$. We have $\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \geq \boldsymbol{\sigma}_{k+1}(\mathbf{B})$ by the min-max theorem:

$$\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) = \max_{\substack{E\subset\mathbb{R}^d \\ \dim(E)=k+1}} \min_{\substack{\mathbf{x}\in E \\ \|\mathbf{x}\|_2=1}} \left\|\widetilde{\mathbf{A}}_{S:}\mathbf{x}\right\|_2 \geq \max_{\substack{E\subset\mathbb{R}^m \\ \dim(E)=k+1}} \min_{\substack{\mathbf{x}\in E \\ \|\mathbf{x}\|_2=1}} \|\mathbf{B}\mathbf{x}\|_2 = \boldsymbol{\sigma}_{k+1}(\mathbf{B}).$$

Then, by (3.29) with $d \leftarrow m$ and $j \leftarrow m-k$,

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \leq \frac{\alpha\sigma(m-k)}{\sqrt{m}}\right] \leq \Pr\left[\boldsymbol{\sigma}_{k+1}(\mathbf{B}) \leq \frac{\alpha\sigma(m-k)}{\sqrt{m}}\right] \leq \left(\sqrt{2e}\alpha\right)^{(m-k)^2}. \tag{3.30}$$

Let $C' := \frac{1}{2}\left(1 - \frac{1}{(1-\eta)C}\right) > 0$, and note that

$$m-k \geq \left(\frac{(1-\eta)n}{d} - 1\right)k = \left(1 - \frac{d}{(1-\eta)n}\right)\left(\frac{(1-\eta)kn}{d}\right) \geq C'm.$$

Setting $\alpha = \frac{m\Delta}{\sigma(m-k)} \leq \frac{\Delta}{C'\sigma}$ in (3.30),

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \leq \sqrt{m}\Delta\right] \leq \left(\sqrt{2e}\alpha\right)^{(m-k)^2} \leq \left(\frac{\sqrt{2e}\Delta}{C'\sigma}\right)^{(m-k)^2}.$$

Now, we can set $\Delta = \frac{C'}{\sqrt{2e}}\left(\frac{\delta\sigma}{n}\right)^{\frac{2}{C'}} \leq \frac{C'\sigma}{\sqrt{2e}}\left(\frac{\delta}{n}\right)^{\frac{2}{C'}}$ to give

$$\left(\frac{\sqrt{2e}\Delta}{C'\sigma}\right)^{(m-k)^2} \leq \left(\frac{\delta}{n}\right)^{\frac{2(m-k)^2}{C'}} \leq \left(\frac{\delta}{n}\right)^{2m(m-k)} \leq \left(\frac{\delta}{n}\right)^{m+1} \leq \frac{\delta^{m+1}}{n\binom{n}{m}} \leq \frac{\delta}{(d-1)\binom{n}{m}},$$

which establishes the claim. □

Lemma 3.11. *In the setting of Lemma 3.10, the result holds if we suppose instead that $d < m \leq Kd$, where $K := \sqrt[3]{C}$.*

Proof. Similarly to the proof of Lemma 3.10, remove any $m-d$ rows of $\widetilde{\mathbf{A}}_{S:}$ to obtain $\mathbf{B} \in \mathbb{R}^{d\times d}$. Then

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \leq \frac{\alpha\sigma(d-k)}{\sqrt{d}}\right] \leq \Pr\left[\boldsymbol{\sigma}_{k+1}(\mathbf{B}) \leq \frac{\alpha\sigma(d-k)}{\sqrt{d}}\right] \leq \left(\sqrt{2e}\alpha\right)^{(d-k)^2}. \tag{3.31}$$

Since $m \le Kd$, we have $\frac{(1-\eta)kn}{d} \le Kd$, which implies $k \le \frac{Kd^2}{(1-\eta)n} \le \frac{Kd}{(1-\eta)C} < d$. Thus $d - k \ge K'd$, where $K' := 1 - \frac{K}{(1-\eta)C} > 0$. Setting $\alpha = \frac{\sqrt{md}\Delta}{\sigma(d-k)} \le \frac{\sqrt{K}\Delta}{K'\sigma}$ in (3.31),

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \le \sqrt{m}\Delta\right] \le \left(\sqrt{2e}\alpha\right)^{(d-k)^2} \le \left(\frac{\sqrt{2eK}\Delta}{K'\sigma}\right)^{(d-k)^2}.$$

Now, we can set $\Delta = \frac{K'}{\sqrt{2eK}}\left(\frac{\delta\sigma}{n}\right)^{\frac{2K}{K'}} \le \frac{K'\sigma}{\sqrt{2eK}}\left(\frac{\delta}{n}\right)^{\frac{2K}{K'}}$ to give

$$\left(\frac{\sqrt{2eK}\Delta}{K'\sigma}\right)^{(d-k)^2} \le \left(\frac{\delta}{n}\right)^{\frac{2K(d-k)^2}{K'}} \le \left(\frac{\delta}{n}\right)^{2Kd} \le \left(\frac{\delta}{n}\right)^{m+1} \le \frac{\delta^{m+1}}{n\binom{n}{m}} \le \frac{\delta}{(d-1)\binom{n}{m}},$$

which establishes the claim. □

3.5.3 Conditioning of tall smoothed matrices

In this section we provide tools for lower bounding the smallest singular value of a random $\Omega(d) \times d$ smoothed matrix $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{G}$, where $\mathbf{G}$ has entries $\sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma^2)$. Specifically we provide estimates, for sufficiently small α, on the quantity

$$\Pr\left[\boldsymbol{\sigma}_d\left(\widetilde{\mathbf{A}}\right) < \alpha\right]. \tag{3.32}$$

We first use the following standard result bounding $\|\widetilde{\mathbf{A}}\|_{\text{op}}$.

Lemma 3.12. *Let $\sigma \in (0,1)$, $m \ge d$, $\mathbf{A} \in \mathbb{R}^{m\times d}$ have rows $\{\mathbf{a}_i\}_{i\in[m]}$ such that $\|\mathbf{a}_i\|_2 = 1$ for all $i \in [m]$, $\mathbf{G} \in \mathbb{R}^{m\times d}$ have entries $\sim_{\text{i.i.d.}} \mathcal{N}(0,\sigma^2)$, and $\widetilde{\mathbf{A}} := \mathbf{A} + \mathbf{G}$. Then there exists a constant $C_{\text{op}} > 0$ such that for all $\delta \in (0,1)$,*

$$\Pr\left[\|\widetilde{\mathbf{A}}\|_{\text{op}} > C_{\text{op}}\sqrt{m + \log\left(\frac{1}{\delta}\right)}\right] \le \delta.$$

Proof. By Theorem 4.4.5, [Ver24], we have that for some constant $C_{\text{op}} > 2$,

$$\Pr\left[\|\mathbf{G}\|_{\text{op}} > \frac{C_{\text{op}}}{2}\sqrt{m + \log\left(\frac{1}{\delta}\right)}\right] \le \delta,$$

where we used $\sigma \le 1$. The conclusion follows as $\|\mathbf{A}\|_{\text{op}} \le \|\mathbf{A}\|_{\text{F}} \le \sqrt{m}$. □

We also require the definition of an ϵ-net and a standard bound on its size.

Definition 3.7. Let $S \subseteq \mathbb{R}^d$, $\mathcal{N} \subset S$ be finite, and $\epsilon \in (0,1)$. We say that $\mathcal{N}$ is an ϵ-net of S if

$$\sup_{\mathbf{u} \in S} \min_{\mathbf{v} \in \mathcal{N}} \|\mathbf{v} - \mathbf{u}\|_2 \leq \epsilon.$$

Fact 3.4 (Corollary 4.2.13, [Ver24])**.** *Let $\partial\mathbb{B}_2(1)$ denote the boundary of the unit norm ball in $\mathbb{R}^d$. For all $\epsilon \in (0,1)$, there exists an ϵ-net of $\partial\mathbb{B}_2(1)$ with $|\mathcal{N}| \leq (\frac{3}{\epsilon})^d$.*

We next observe that it suffices to provide estimates on a net, given an operator norm bound.

Lemma 3.13. *Let $m \geq d$, $\epsilon \in (0,1)$, and $\mathcal{N}$ be an ϵ-net of $\partial\mathbb{B}_2(1) \subset \mathbb{R}^d$, and suppose that $\widetilde{\mathbf{A}} \in \mathbb{R}^{m \times d}$ satisfies $\|\widetilde{\mathbf{A}}\|_{\mathrm{op}} \leq \rho$. Then $\boldsymbol{\sigma}_d(\widetilde{\mathbf{A}}) \geq \min_{\mathbf{v} \in \mathcal{N}} \|\widetilde{\mathbf{A}}\mathbf{v}\|_2 - \epsilon\rho$.*

Proof. Let $\mathbf{u}$ realize $\boldsymbol{\sigma}_d(\widetilde{\mathbf{A}})$ in the definition $\boldsymbol{\sigma}_d(\widetilde{\mathbf{A}}) = \min_{\mathbf{u} \in \partial\mathbb{B}_2(1)} \|\widetilde{\mathbf{A}}\mathbf{u}\|_2$. Then if we define $\mathbf{v} := \arg\min_{\mathbf{v} \in \mathcal{N}} \|\mathbf{u} - \mathbf{v}\|_2$, the result follows from the triangle inequality:

$$\|\widetilde{\mathbf{A}}\mathbf{u}\|_2 \geq \|\widetilde{\mathbf{A}}\mathbf{v}\|_2 - \|\widetilde{\mathbf{A}}\|_{\mathrm{op}} \|\mathbf{u} - \mathbf{v}\|_2 \geq \min_{\mathbf{v} \in \mathcal{N}} \|\widetilde{\mathbf{A}}\mathbf{v}\|_2 - \epsilon\rho.$$

□

Finally, we provide tail bounds on the contraction given by $\widetilde{\mathbf{A}}$ on a single fixed vector.

Lemma 3.14. *In the setting of Lemma 3.12, let $\mathbf{v} \in \mathbb{R}^d$ have $\|\mathbf{v}\|_2 = 1$. Then,*

$$\Pr\left[\|\widetilde{\mathbf{A}}\mathbf{v}\|_2 < \alpha\right] \leq \left(\frac{\alpha}{\sigma}\right)^m \text{ for all } \alpha \in (0,1).$$

Proof. Observe that if $\|\widetilde{\mathbf{A}}\mathbf{v}\|_2 \leq \alpha$, then every coordinate of $\widetilde{\mathbf{A}}\mathbf{v}$ is bounded by α. Each coordinate of $\widetilde{\mathbf{A}}\mathbf{v}$ is distributed independently as $\mathcal{N}(\langle \mathbf{a}_i, \mathbf{v} \rangle, \sigma^2)$. Now the claim follows because for all $i \in [m]$,

$$\begin{aligned}
\Pr_{\xi \sim \mathcal{N}(\langle \mathbf{a}_i, \mathbf{v} \rangle, \sigma^2)} [|\xi| < \alpha] &= \Pr_{\xi \sim \mathcal{N}(\frac{1}{\sigma}\langle \mathbf{a}_i, \mathbf{v} \rangle, 1)} \left[|\xi| < \frac{\alpha}{\sigma}\right] \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\frac{\alpha}{\sigma}}^{\frac{\alpha}{\sigma}} \exp\left(-\frac{(\xi - \frac{1}{\sigma}\langle \mathbf{a}_i, \mathbf{v} \rangle)^2}{2}\right) \mathrm{d}\xi \leq \frac{\alpha}{\sigma}.
\end{aligned}$$

□

We can now prove our main tail bound on $\boldsymbol{\sigma}_d(\widetilde{\mathbf{A}})$.

Lemma 3.15. *In the setting of Lemma 3.12, suppose that $m \geq Kd$ for a constant $K > 1$. Then there exists a constant $\beta > 0$ such that*

$$\Pr\left[\boldsymbol{\sigma}_d\left(\widetilde{\mathbf{A}}\right) < \alpha\right] \leq 2\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}} \quad \text{for all } \alpha \in \left(0, \frac{\sigma^{\frac{K+3}{K-1}}}{\beta m^{\frac{2}{K-1}}}\right).$$

Proof. Let $\delta := 2\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}}$. By Lemma 3.12, there exists $C_{\mathrm{op}} > 0$ such that

$$\Pr\left[\|\widetilde{\mathbf{A}}\|_{\mathrm{op}} > \rho\right] \leq \frac{\delta}{2} \text{ for } \rho := C_{\mathrm{op}}\sqrt{m + \log\left(\frac{2}{\delta}\right)}.$$

Let $L > 0$ be a constant such that $\sqrt{\log(\frac{c}{2})} \leq Lc^{\frac{K-1}{4}}$ for all $c \geq 2$, and let

$$\beta := \max\left(2e, \left(2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}L\right)^{\frac{4}{K-1}}\right) \geq 2e.$$

Then for the stated range of α,

$$\alpha \leq \frac{\sigma}{2e} \implies \rho = C_{\mathrm{op}}\sqrt{m + \frac{(K-1)m}{2K}\log\left(\frac{\sigma}{2\alpha}\right)} \leq 2C_{\mathrm{op}}\sqrt{m\log\left(\frac{\sigma}{2\alpha}\right)}.$$

Let $\epsilon := \frac{\alpha}{\rho}$, and let $\mathcal{N}$ be an ϵ-net of $\partial\mathbb{B}_2(1) \in \mathbb{R}^d$ with size $|\mathcal{N}| \leq (\frac{3}{\epsilon})^d$, as guaranteed by Fact 3.4. Then by taking a union bound over Lemma 3.14 applied to each $\mathbf{v} \in \mathcal{N}$, $\Pr\left[\min_{\mathbf{v}\in\mathcal{N}}\|\widetilde{\mathbf{A}}\mathbf{v}\|_2 < 2\alpha\right]$ is at most

$$\begin{aligned}
|\mathcal{N}|\left(\frac{2\alpha}{\sigma}\right)^m &\leq \left(\frac{6C_{\mathrm{op}}\sqrt{m\log(\frac{\sigma}{2\alpha})}}{\alpha}\right)^{\frac{m}{K}}\left(\frac{2\alpha}{\sigma}\right)^m \\
&= \left(\frac{6C_{\mathrm{op}}\sqrt{m\log(\frac{\sigma}{2\alpha})}}{\alpha}\right)^{\frac{m}{K}}\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K+1)m}{2K}}\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}} \\
&\leq \left(\frac{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}\sqrt{m\log(\frac{\sigma}{2\alpha})}}{\sigma^{\frac{K+1}{2}}} \cdot \frac{\sigma^{\frac{K+3}{4}}\alpha^{\frac{K-1}{4}}}{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}L\sqrt{m}}\right)^{\frac{m}{K}}\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}} \\
&\leq \left(\frac{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}\sqrt{m\log(\frac{\sigma}{2\alpha})}}{\sigma^{\frac{K+1}{2}}} \cdot \frac{\sigma^{\frac{K+1}{2}}}{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}\sqrt{m\log(\frac{\sigma}{2\alpha})}}\right)^{\frac{m}{K}}\left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}} \\
&= \left(\frac{2\alpha}{\sigma}\right)^{\frac{(K-1)m}{2K}} = \frac{\delta}{2},
\end{aligned}$$

where the third line uses

$$\alpha^{\frac{K-1}{4}} \le \frac{\sigma^{\frac{K+3}{4}}}{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}L\sqrt{m}} \implies \alpha^{\frac{K-1}{2}} \le \frac{\sigma^{\frac{K+3}{4}}\alpha^{\frac{K-1}{4}}}{2^{\frac{K+1}{2}} \cdot 6C_{\mathrm{op}}L\sqrt{m}}$$

for the stated range of α and the fourth line uses $\sqrt{\log(\frac{c}{2})} \le Lc^{\frac{K-1}{4}}$ for $c \ge 2$. The claim follows from a union bound on the above two events and Lemma 3.13. □

Applying Lemma 3.15 then gives our extension to tall matrices.

Lemma 3.16. *In the setting of Lemma 3.10, the result holds if we suppose instead that $m > Kd$, where $K := \sqrt[3]{C}$, and in addition that $\mathbf{A}$ has rows $\{\mathbf{a}_i\}_{i\in[n]}$ satisfying $\|\mathbf{a}_i\|_2 = 1$ for all $i \in [n]$.*

Proof. Let β be the constant in Lemma 3.15, and let $\alpha = \sqrt{m}\Delta$, where

$$\Delta = \frac{1}{2}\left(\frac{\delta\sigma}{\beta n}\right)^{\frac{4K}{K-1}+1} \le \frac{\sigma}{2\sqrt{m}}\left(\frac{\delta\sigma}{\beta n}\right)^{\frac{4K}{K-1}} \in \left(0, \frac{\sigma^{\frac{K+3}{K-1}}}{\beta m^{\frac{2}{K-1}+\frac{1}{2}}}\right).$$

By Lemma 3.15,

$$\begin{aligned}\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \le \sqrt{m}\Delta\right] \le \Pr\left[\boldsymbol{\sigma}_{d}(\widetilde{\mathbf{A}}_{S:}) \le \sqrt{m}\Delta\right] &\le 2\left(\left(\frac{\delta\sigma}{\beta n}\right)^{\frac{4K}{K-1}}\right)^{\frac{(K-1)m}{2K}} \\ &\le \left(\frac{\delta}{n}\right)^{2m} \le \left(\frac{\delta}{n}\right)^{m+1} \le \frac{\delta^{m+1}}{n\binom{n}{m}} \le \frac{\delta}{(d-1)\binom{n}{m}},\end{aligned}$$

which establishes the claim. □

3.5.4 Assumption 3.1 for smoothed matrices

We can now put together the previous results to give a diameter bound for smoothed matrices. To begin, we show simple norm bounds on the rows of a smoothed matrix.

Lemma 3.17. *Under Assumption 3.2, let $\delta \in (0,1)$, $\frac{1}{\sigma} \ge 10(d + \log(\frac{n}{\delta}))$, $\mathbf{A} \in \mathbb{R}^{n\times d}$ have rows $\{\mathbf{a}_i\}_{i\in[n]}$ such that $\|\mathbf{a}_i\|_2 = 1$ for all $i \in [n]$, $\mathbf{G} \in \mathbb{R}^{n\times d}$ have entries $\sim_{\text{i.i.d.}} \mathcal{N}(0,\sigma^2)$, and $\widetilde{\mathbf{A}} := \mathbf{A} + \mathbf{G}$ have rows $\{\tilde{\mathbf{a}}_i\}_{i\in[n]}$. Then with probability $\ge 1-\delta$, $\frac{1}{6} \le \|\tilde{\mathbf{a}}_i\|_2^2 \le 2$ for all $i \in [n]$.*

Proof. By using the inequalities

$$\frac{1}{2}\left\|\mathbf{a}_i\right\|_2^2 - \left\|\mathbf{g}_i\right\|_2^2 \leq \left\|\mathbf{a}_i + \mathbf{g}_i\right\|_2^2 \leq \frac{3}{2}\left\|\mathbf{a}_i\right\|_2^2 + 3\left\|\mathbf{g}_i\right\|_2^2,$$

it is enough to show that for all $i \in [n]$, the probability that $\left\|\mathbf{g}_i\right\|_2^2 \geq \frac{1}{6}$ is bounded by $\frac{\delta}{n}$. This follows from a standard χ^2 tail bound, e.g., Lemma 1, [LM00], for our choice of σ. □

It remains to show that $\frac{d}{n}\mathbf{1}_n$ is deep inside the basis polytope of the rows of $\widetilde{\mathbf{A}}$, so that we can use Lemma 3.8 to obtain a diameter bound for minimizing Barthe's objective.

Lemma 3.18. *In the setting of Lemma 3.17, $\frac{d}{n}\mathbf{1}_n$ is (η, Δ)-deep inside the basis polytope of the rows of $\widetilde{\mathbf{A}}$ with probability $\geq 1-\delta$, where $\eta := 1-\frac{1}{\sqrt{C}}$ and $\Delta = \left(\frac{\delta\sigma}{n}\right)^{O(1)}$.*

Proof. Let $k \in [d-1]$. By Lemma 3.9, if some $m := \lceil \frac{(1-\eta)kn}{d} \rceil$ rows of $\widetilde{\mathbf{A}}$ indexed by S violate the condition for (η, Δ)-deepness, then $\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \leq \sqrt{m}\Delta$. By Lemma 3.10, Lemma 3.11, and Lemma 3.16,

$$\Pr\left[\boldsymbol{\sigma}_{k+1}(\widetilde{\mathbf{A}}_{S:}) \leq \sqrt{m}\Delta\right] \leq \frac{\delta}{(d-1)\binom{n}{m}}$$

for any $S \subseteq [n]$ with $|S| = m$, in every range of $k \in [d-1]$. By a union bound over all S, the failure probability is at most $\frac{\delta}{d-1}$. The result follows by a union bound over all $k \in [d-1]$. □

At this point, we have all the tools necessary to prove Theorem 3.3.

Theorem 3.3. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ have rows $\{\mathbf{a}_i\}_{i\in[n]}$ such that $\left\|\mathbf{a}_i\right\|_2 = 1$ for all $i \in [n]$, let $\mathbf{c} := \frac{d}{n}\mathbf{1}_n$, let $\delta \in (0,1)$, and let $\sigma \in (0, \frac{\delta}{10nd})$. Let $\widetilde{\mathbf{A}} := \mathbf{A} + \mathbf{G}$, where $\mathbf{G} \in \mathbb{R}^{n\times d}$ has entries $\sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma^2)$. Then with probability $\geq 1-\delta$, if $n \geq Cd$ where C is any constant larger than 1, Assumption 3.1 holds for Barthe's objective f defined with respect to $(\widetilde{\mathbf{A}}, \mathbf{c})$, where*

$$\log(\kappa) = O\left(d\log\left(\frac{1}{\sigma}\right)\right).$$

Proof. By Lemma 3.17, in the relevant range of σ, the conclusion

$$\frac{1}{6} \leq \|\tilde{\mathbf{a}}_i\|_2^2 \leq 2$$

holds for all $i \in [n]$ with probability $\geq 1 - \frac{\delta}{2}$. Moreover, let $\Delta = (\frac{\delta\sigma}{n})^{O(1)}$ so that $\frac{d}{n}\mathbf{1}_n$ is (η, Δ)-deep inside the basis polytope of the rows of $\widetilde{\mathbf{A}}$ with probability $\geq 1 - \frac{\delta}{2}$ by Lemma 3.18. By a union bound on these events, with probability $\geq 1 - \delta$, we can apply Lemma 3.8 and conclude that there exists $\mathbf{t}^\star \in \arg\min_{\mathbf{t}\in\mathbb{R}^n} f(\mathbf{t})$ satisfying

$$\|\mathbf{t}^\star\|_\infty \leq \frac{1}{2}\log\left(\frac{12n}{d}\left(\frac{8}{\eta\Delta^2}\right)^{d-1}\right) = O\left(d\log\left(\frac{1}{\sigma}\right)\right).$$

□

3.5.5 Extension to non-uniform c

Our smoothed diameter bound in Theorem 3.3 is stated with respect to uniform marginals $\mathbf{c} = \frac{d}{n}\mathbf{1}_n$. However, the analysis in this section can be straightforwardly extended to hold for $\mathbf{c}$ with nonuniform entries by a reduction, as long as $\mathbf{c}$ is sufficiently bounded away from $\mathbf{1}_n$ entrywise.

Corollary 3.1. *In the setting of Theorem 3.3, if we suppose instead that* $\mathbf{c} \in (0, 1]^n$ *satisfies* $\|\mathbf{c}\|_1 = d$ *and* $\mathbf{c} \leq c \cdot \frac{d}{n}\mathbf{1}_n$ *entrywise, where* $1 < c < C \leq \frac{n}{d}$ *for constants* c, C*, then Assumption 3.1 holds for Barthe's objective* f *defined with respect to* $(\widetilde{\mathbf{A}}, \mathbf{c})$*, where*

$$\log(\kappa) = O\left(d\log\left(\frac{1}{\sigma}\right) + \log\left(\frac{1}{\mathbf{c}_{\min}}\right)\right) \text{ and } \mathbf{c}_{\min} := \min_{i\in[n]} \mathbf{c}_i.$$

Proof. Our proof of Lemma 3.18 shows that with probability $\geq 1 - \delta$ in the setting of Theorem 3.3, $\frac{d}{n}\mathbf{1}_n$ is (η, Δ)-deep for a constant η arbitrarily close to $1 - \frac{1}{C}$, where $C \leq \frac{n}{d}$ (see Remark 3.3). However, this also implies that for all $k \in [d-1]$ and k-dimensional subspaces E, recalling Definition 3.6,

$$\sum_{i\in[n]} \mathbf{c}_i \mathbb{I}_{\|\mathbf{a}_i - \mathbf{\Pi}_E \mathbf{a}_i\|_2 \leq \Delta} \leq c \sum_{i\in[n]} \frac{d}{n} \mathbb{I}_{\|\mathbf{a}_i - \mathbf{\Pi}_E \mathbf{a}_i\|_2 \leq \Delta} \leq c(1-\eta)k = (1 - (1 - c(1-\eta)))k.$$

Thus, we have shown that $\mathbf{c}$ is also $(1 - c(1-\eta), \Delta)$-deep. Since η can be arbitrarily close to $1 - \frac{1}{C}$ and $c < C$, we can verify that the new parameter $1 - c(1-\eta)$ satisfies $0 < 1 - c(1-\eta) < 1 - \frac{1}{C}$, so the rest of our proof applies (propagating constant changes appropriately) by Remark 3.3. $\square$

Appendix A

Mathematical Facts

In this appendix, we provide, without proof, several mathematical results that are used throughout the thesis. While many of these results are well-known within their respective domains, they are included here for completeness and convenience. Detailed proofs and broader discussions of these results can be found in the standard literature; we refer to e.g. [Bha07, HJ12] for matrix theory and [Roc70, BV04b] for convex analysis.

A.1 Matrix theory

Fact A.1 (Spectral theorem)**.** *Let* $\mathbf{A} \in \mathbb{S}^{d\times d}$*. Then there exist an orthogonal matrix* $\mathbf{U} \in \mathbb{R}^{d\times d}$ *and a diagonal matrix* $\mathbf{\Lambda} \in \mathbb{R}^{d\times d}$ *such that* $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ *and the diagonal entries of* $\mathbf{\Lambda}$ *are the eigenvalues of* $\mathbf{A}$*.*

Fact A.2. *Let* $\mathbf{A} \in \mathbb{S}^{d\times d}$*. The following are equivalent.*

1. $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ *(resp.* $\mathbf{A} \in \mathbb{S}^{d\times d}_{\succ \mathbf{0}}$*).*
2. $\mathbf{v}^\top \mathbf{A}\mathbf{v} \geq 0$ *for all* $\mathbf{v} \in \mathbb{R}^d$ *(resp.* $\mathbf{v}^\top \mathbf{A}\mathbf{v} > 0$ *for all* $\mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$*).*
3. *The eigenvalues of* $\mathbf{A}$ *are nonnegative (resp. positive).*

4. *There exists* $n \in \mathbb{N}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times d}$ *(resp. invertible* $\mathbf{B} \in \mathbb{R}^{d\times d}$*) such that* $\mathbf{A} = \mathbf{B}^\top\mathbf{B}$.

Fact A.3. *For matrices* $\mathbf{A}, \mathbf{B}$ *with compatible dimensions,* $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$.

Fact A.4. *If* $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ *are PSD matrices of the same dimension and* $\mathbf{A} \preceq \mathbf{B}$ *and* $\mathbf{C} \preceq \mathbf{D}$*, then* $\langle \mathbf{A}, \mathbf{C}\rangle \leq \langle \mathbf{B}, \mathbf{D}\rangle$.

Fact A.5. *For all* $\mathbf{A} \in \mathbb{R}^{n\times d}$*, we have* $\boldsymbol{\tau}(\mathbf{A}) \in [0,1]^n$*, and* $\sum_{i\in[n]} \boldsymbol{\tau}_i(\mathbf{A}) = \mathrm{rank}(\mathbf{A})$.

Fact A.6 (Cauchy–Binet formula)**.** *Let* $\mathbf{A} \in \mathbb{R}^{d\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times d}$*. Then*

$$\det(\mathbf{AB}) = \sum_{\substack{S\subseteq[n]\\|S|=d}} \det(\mathbf{A}_{:S}) \det(\mathbf{B}_{S:}).$$

Fact A.7 (Matrix determinant lemma)**.** *Let* $\mathbf{A} \in \mathbb{R}^{d\times d}$ *be invertible and* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$*. Then* $\det(\mathbf{A} + \mathbf{uv}^\top) = (1 + \mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}) \det(\mathbf{A})$.

Fact A.8 (Min-max theorem)**.** *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$*. For all* $k \in [\min(n, d)]$*,*

$$\boldsymbol{\sigma}_k(\mathbf{A}) = \min_{\substack{S\subseteq\mathbb{R}^d\\ \dim(S)=d-k+1}} \max_{\substack{\mathbf{x}\in S\\ \|\mathbf{x}\|_2=1}} \|\mathbf{Ax}\|_2 = \max_{\substack{S\subseteq\mathbb{R}^d\\ \dim(S)=k}} \min_{\substack{\mathbf{x}\in S\\ \|\mathbf{x}\|_2=1}} \|\mathbf{Ax}\|_2 .$$

A.2 Convex analysis

Definition A.1 (Convex set)**.** A set $S \subseteq \mathbb{R}^d$ is *convex* if $(1-\lambda)\mathbf{x} + \lambda\mathbf{y} \in S$ for all $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0,1]$.

Definition A.2 (Convex function)**.** A function $f : S \to \mathbb{R} \cup \{\pm\infty\}$ is *convex* if S is convex and $f((1-\lambda)\mathbf{x} + \lambda\mathbf{y}) \leq (1-\lambda)f(\mathbf{x}) + \lambda f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0,1]$. If the inequality is strict, then f is *strictly convex*.

Fact A.9. *S is a convex set iff its* characteristic function

$$\chi_S(\mathbf{x}) := \begin{cases} 0 & \mathbf{x} \in S \\ \infty & \mathbf{x} \notin S \end{cases}$$

is a convex function. $f : S \to \mathbb{R} \cup \{\pm\infty\}$ *is a convex function iff its* epigraph

$$\operatorname{epi}(f) := \{(\mathbf{x}, r) \in S \times \mathbb{R} \mid r \geq f(\mathbf{x})\}$$

is a convex set.

Throughout the rest of this section, $S \subseteq \mathbb{R}^d$ is assumed to be a convex set.

Definition A.3 (Effective domain)**.** The *effective domain* of $f : S \to \mathbb{R} \cup \{\pm\infty\}$ is

$$\operatorname{dom}(f) := \{\mathbf{x} \in S \mid f(\mathbf{x}) < +\infty\}.$$

Definition A.4 (Closed function)**.** A function $f : S \to \mathbb{R} \cup \{\pm\infty\}$ is *closed* if $\operatorname{epi}(f)$ is closed.

Definition A.5 (Proper function)**.** A function $f : S \to \mathbb{R} \cup \{\pm\infty\}$ is *proper* if it is finite on a nonempty set.

Definition A.6 (Subgradient)**.** A vector $\mathbf{g}$ is a *subgradient* of $f : S \to \mathbb{R}$ at $\mathbf{x} \in S$ if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle \text{ for all } \mathbf{y} \in S.$$

$\partial f(\mathbf{x})$ denotes the set of subgradients of f at $\mathbf{x}$.

Fact A.10 (First-order condition)**.** *A differentiable function* $f : S \to \mathbb{R}$ *is convex iff*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

Fact A.11 (Second-order condition)**.** *A twice-differentiable function* $f : S \to \mathbb{R}$ *is convex iff* $\nabla^2 f(\mathbf{x}) \in \mathbb{S}^{d \times d}_{\succeq \mathbf{0}}$ *for all* $\mathbf{x} \in \operatorname{dom}(f)$.

Definition A.7 (Dual norm)**.** Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$. The *dual norm* of $\|\cdot\|$ is $\|\cdot\|_* := \max_{\|\mathbf{x}\| \leq 1} \langle \mathbf{x}, \cdot \rangle$.

Fact A.12. *For any norm* $\|\cdot\|$ *on* $\mathbb{R}^d$, $\|\cdot\|_{**} = \|\cdot\|$.

Definition A.8 (Strong convexity)**.** A function $f : S \to \mathbb{R}$ is *μ-strongly convex* on S with respect to $\|\cdot\|$ if

$$f((1-\lambda)\mathbf{x} + \lambda\mathbf{y}) \leq (1-\lambda)f(\mathbf{x}) + \lambda f(\mathbf{y}) - \frac{\mu\lambda(1-\lambda)}{2}\|\mathbf{x}-\mathbf{y}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

Fact A.13. *If $f : S \to \mathbb{R}$ is differentiable, then f is μ-strongly convex on S with respect to $\|\cdot\|$ iff*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

If f is twice-differentiable, then f is μ-strongly convex on S with respect to $\|\cdot\|$ iff

$$\nabla^2 f(\mathbf{x})[\mathbf{y}, \mathbf{y}] \geq \mu\|\mathbf{y}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

Definition A.9 (Smoothness)**.** A differentiable function $f : S \to \mathbb{R}$ is *L-smooth* on S with respect to $\|\cdot\|$ if

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_* \leq L\|\mathbf{x}-\mathbf{y}\| \text{ for all } \mathbf{x}, \mathbf{y} \in S,$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

Fact A.14. *If $f : S \to \mathbb{R}$ is differentiable and convex, then f is L-smooth on S with respect to $\|\cdot\|$ iff*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

If f is twice-differentiable and possibly nonconvex, then f is L-smooth on S with respect to $\|\cdot\|$ iff

$$\left|\nabla^2 f(\mathbf{x})[\mathbf{y}, \mathbf{y}]\right| \leq L\|\mathbf{y}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in S.$$

Definition A.10 (Convex conjugate)**.** Let X be a real topological vector space with dual space X^*, and let $f : X \to \mathbb{R} \cup \{\pm\infty\}$. The *convex conjugate* (or *Fenchel dual*, or *Legendre transform*) of f is the function $f^* : X^* \to \mathbb{R} \cup \{\pm\infty\}$ given by

$$f^*(\mathbf{x}^*) := \sup_{\mathbf{x} \in X} \langle \mathbf{x}^*, \mathbf{x}\rangle - f(\mathbf{x}) \text{ for all } \mathbf{x}^* \in X^*.$$

Fact A.15. *Let f^* be the convex conjugate of a convex, closed, and proper function f. Then $f^{**} = f$, and for all $\mathbf{x}^* \in X^*$, $\mathbf{x} \in \partial f^*(\mathbf{x}^*)$ iff $\mathbf{x} \in \arg\max_{\mathbf{x} \in X} \langle \mathbf{x}^*, \mathbf{x}\rangle - f(\mathbf{x})$.*

A.3 Useful inequalities

Fact A.16 (Titu's lemma)**.** *For all* $\mathbf{u} \in \mathbb{R}^d$ *and* $\mathbf{v} \in \mathbb{R}^d_{>0}$*, we have*

$$\left(\sum_{i\in[d]} \mathbf{u}_i\right)^2 \leq \left(\sum_{i\in[d]} \frac{\mathbf{u}_i^2}{\mathbf{v}_i}\right)\left(\sum_{i\in[d]} \mathbf{v}_i\right).$$

Fact A.17 (Weighted AM–GM inequality)**.** *For all* $\mathbf{x} \in \mathbb{R}^d_{\geq 0}$ *and* $\mathbf{w} \in \Delta^d$*, we have*

$$\prod_{i\in[d]} \mathbf{x}_i^{\mathbf{w}_i} \leq \sum_{i\in[d]} \mathbf{w}_i \mathbf{x}_i.$$

Fact A.18 (Hölder's inequality)**.** *For all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ *and* $p, q \in [1, \infty]$ *satisfying* $\frac{1}{p} + \frac{1}{q} = 1$*, we have*

$$|\langle \mathbf{x}, \mathbf{y}\rangle| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q.$$

Fact A.19. *For all* $c \in \mathbb{R}$*, we have* $c(\max(0, 1-c) + \max(0, 1-c)^2) \leq \max(0, 1-c)$.

Fact A.20. *For all* $b, c \in [0, 1]$*, we have* $\log(\frac{1}{1+bc}) \leq -b(1-b)c$.

Fact A.21. *For all* $c \in \mathbb{R}$*, we have* $1 + c \leq \exp(c)$.

Fact A.22. *For all* $c > -1$*, we have* $\log(1 + c) \leq c$.

Fact A.23. *For all* $c > -1$*, we have* $\exp(-c) \leq \frac{1}{1+c}$.

Fact A.24. *For all* $c > 0$*, we have* $c - \frac{c^2}{2} \leq 1 - \exp(-c)$.

Fact A.25. *For all* $b, c \in (0, 1)$*, we have* $\frac{1}{2}((1-b)c)^2 \leq \log(1 + c) - b\log(1 + \frac{c}{b})$.

Bibliography

[ACSS20] Josh Alman, Timothy Chu, Aaron Schild, and Zhao Song. Algorithms and hardness for linear algebra on geometric graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 541–552. IEEE, 2020. 3.2.3

[ADK+16] Ittai Abraham, David Durfee, Ioannis Koutis, Sebastian Krinninger, and Richard Peng. On fully dynamic graph sparsifiers. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 335–344. IEEE Computer Society, 2016. 3.2.3

[ADV+25] 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 2025*, pages 2005–2039. SIAM, 2025. 1.3

[AGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 172–181. ACM, 2018. 3.2, 3.2.2

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012. 2.1

[AJJ+22] Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 627–669. SIAM, 2022. 3.4.3

[AKS20] Shiri Artstein-Avidan, Haim Kaplan, and Micha Sharir. On radial isotropic position: Theory and algorithms. *CoRR*, abs/2005.04918, 2020. 3.1.1, 3.2, 3.3.3, 3.3.4, 3.5, 3.5.1, 3.5.1

[ALdOW17] Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Much faster algorithms for matrix scaling. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 890–901. IEEE Computer Society, 2017. 3.2, 3.3, 3.3.1

[ALO16] Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 1824–1831. SIAM, 2016. 2.1, 3.3.2

[Ans00] Kurt M. Anstreicher. The volumetric barrier for semidefinite programming. *Math. Oper. Res.*, 25(3):365–380, 2000. 1.1

[AO19] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly linear-time packing and covering LP solvers - achieving width-independence and -convergence. *Math. Program.*, 175(1-2):307–353, 2019. 2.1

[ARV09] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009. 1.1

[AW08] Arash A. Amini and Martin J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *2008 IEEE International Symposium on Information Theory, ISIT 2008*, pages 2454–2458. IEEE, 2008. 1.1

[Bar98] Franck Barthe. On a reverse form of the brascamp-lieb inequality. *Inventiones Mathematicae*, 134(2):335–361, 1998. 3.1, 3.1.1, 3.1.1, 3.1.1, 3.1.1, 3.3

[BFG+18] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 883–897. IEEE Computer Society, 2018. 3.2.2

[BFG+19] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 845–861. IEEE Computer Society, 2019. 3.2.2

[Bha07] Rajendra Bhatia. *Positive Definite Matrices*. Princeton University Press, 2007. 3.1.1, A

[BKMS21] Jess Banks, Archit Kulkarni, Satyaki Mukherjee, and Nikhil Srivastava. Gaussian regularization of the pseudospectrum and davies' conjecture. *Communications on Pure and Applied Mathematics*, 74(10):2114–2131, 2021. 3.3

[BLNW20] Peter Bürgisser, Yinan Li, Harold Nieuwboer, and Michael Walter. Interior-point methods for unconstrained geometric programming and scaling problems. *CoRR*, abs/2008.12110, 2020. 3.2.1, 3.3

[BV04a] Dimitris Bertsimas and Santosh S. Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004. 1.1

[BV04b] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 3.4.1, A

[Che24] Yeshwanth Cherapanamjeri. Computing approximate centerpoints in polynomial time. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024*, pages 1654–1668. IEEE, 2024. 3.1, 3.2, 3.3.1, 3.3.3

[CJJ+20] 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: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. 3.3.1, 3.4, 3.4.1, 3.4.1

[CKV20] L. Elisa Celis, Vijay Keswani, and Nisheeth K. Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 1349–1359. PMLR, 2020. 3.2.1, 1

[CKYV19] L. Elisa Celis, Vijay Keswani, Ozan Yildiz, and Nisheeth K. Vishnoi. Fair distributions from biased samples: A maximum entropy optimization framework. *CoRR*, abs/1906.02164, 2019. 3.2.1, 1

[CLL04] Eric Carlen, Elliott Lieb, and Michael Loss. A sharp analog of young's inequality on s^n and related entropy inequalities. *The Journal of Geometric Analysis*, 14:487–520, 2004. 3.1.1, 3.1.1, 3.2

[CLMW11] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011. 4

[CMTV17] Michael B. Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained newton's method and interior point methods. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 902–913. IEEE Computer Society, 2017. 3.2, 3.3, 3.3.1, 3.4, 3.4, 3.4.2, 3.4.3, 3.4.3

[CMY20] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 141–148. IEEE, 2020. 2.1, 2.1

[CPW21] Li Chen, Richard Peng, and Di Wang. 2-norm flow diffusion in near-linear time. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 540–549. IEEE, 2021. 3.3.1, 3.4.3, 3.4.3, 3.6, 3.4.3

[CPW25] Li Chen, Richard Peng, and Di Wang. Personal communication, 2025. 3.4.3

[CT10] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inf. Theory*, 56(5):2053–2080, 2010. 2.1.1

[dGJL07] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Rev.*, 49(3):434–448, 2007. 1.1

[DJ07] Florian Diedrich and Klaus Jansen. Faster and simpler approximation algorithms for mixed packing and covering problems. *Theor. Comput. Sci.*, 377(1-3):181–204, 2007. 2.1

[DKK+21] Ilias Diakonikolas, Daniel Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. List-decodable mean estimation in nearly-pca time. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 10195–10208, 2021. 2, 2.1, 2.2

[DKT21] Ilias Diakonikolas, Daniel Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 7732–7744, 2021. 3.1, 3.1.1, 3.2

[DR24] Daniel Dadush and Akshay Ramachandran. Strongly polynomial frame scaling to high precision. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024*, pages 962–981. SIAM, 2024. 3.1.1, 3.2, 3.2.2, 3.3.3

[DSW14] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Breaking the quadratic barrier for 3-lcc's over the reals. In *Symposium on Theory of Computing, STOC 2014*, pages 784–793. ACM, 2014. 3.1, 3.1.1, 3.1.1

[DTK23] Ilias Diakonikolas, Christos Tzamos, and Daniel M. Kane. A strongly polynomial algorithm for approximate forster transforms and its application to halfspace learning. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 1741–1754. ACM, 2023. 3.1, 3.1.1, 3.2, 3.2.2, 3.3.3

[DV22] Xuan Vinh Doan and Stephen A. Vavasis. Low-rank matrix recovery with ky fan 2-k-norm. *J. Glob. Optim.*, 82(4):727–751, 2022. 2.1.1

[For02] Jurgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002. 3.1, 3.1.1

[Fra18] Cole Franks. Operator scaling with specified marginals. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 190–203. ACM, 2018. 3.2.2

[GGdOW16] Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 109–117. IEEE Computer Society, 2016. 3.2.2

[GGdOW17] Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of brascamp-lieb inequalities, via operator scaling. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 397–409. ACM, 2017. 3.2.2

[GGMS87] I. M. Gelfand, R. M. Goresky, R. D. MacPherson, and V. V. Serganova. Combinatorial geometries, convex polyhedra, and schubert cells. *Advances in Mathematics*, 63(3):301–316, 1987. 3.1, 3.1.1

[GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Comb.*, 1(2):169–197, 1981. 1.1

[GSW22] Elisabeth Gaar, Melanie Siebenhofer, and Angelika Wiegele. An sdp-based approach for computing the stability number of a graph. *Math. Methods Oper. Res.*, 95(1):141–161, 2022. 1.1

[GV02] Jean-Louis Goffin and Jean-Philippe Vial. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optim. Methods Softw.*, 17(5):805–867, 2002. 1.1

[GW94] Michel X. Goemans and David P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2sat. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 422–431. ACM, 1994. 1.1

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 2012. A

[HJS+22] Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. Solving SDP faster: A robust IPM framework and efficient implementation. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, pages 233–244. IEEE, 2022. 1.1

[HKLM20] Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Point location and active learning: Learning halfspaces almost optimally. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 1034–1044. IEEE, 2020. 3.1, 3.1.1, 3.1, 3.1.1, 3.1.1

[HM13] Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *COLT 2013 - The 26th Annual Conference on Learning Theory*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 354–375. JMLR.org, 2013. 3.1, 3.1.1, 3.2, 3.2.1, 2, 3.3.1, 3.3.3, 3.3.4

[HM19] Linus Hamilton and Ankur Moitra. The paulsen problem made simple. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019*, volume 124 of *LIPIcs*, pages 41:1–41:6. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 3.1, 3.2, 3.3.3

[IQS17] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017*, volume 67 of *LIPIcs*, pages 55:1–55:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 3.2.2

[JC16] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A.*, 2016. 2.3

[JKL+20] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 910–918. IEEE, 2020. 1.1

[JKL+24] Arun Jambulapati, Syamantak Kumar, Jerry Li, Shourya Pandey, Ankit Pensia, and Kevin Tian. Black-box k-to-1-pca reductions: Theory and applications. In *The Thirty Seventh Annual Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 2564–2607. PMLR, 2024. 2.3

[JLL+20] Arun Jambulapati, Yin Tat Lee, Jerry Li, Swati Padmanabhan, and Kevin Tian. Positive semidefinite programming: mixed, parallel, and width-independent. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 789–802. ACM, 2020. 2.1

[JLM+23] Arun Jambulapati, Jerry Li, Christopher Musco, Kirankumar Shiragur, Aaron Sidford, and Kevin Tian. Structured semidefinite programming for recovering structured preconditioners. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023. 3.2.3, 3.3.2

[JLSW20] 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*, pages 944–953. ACM, 2020. 1.1, 3.2

[JLT20] Arun Jambulapati, Jerry Li, and Kevin Tian. Robust sub-gaussian principal component analysis and width-independent schatten packing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020. 1.1, 2, 2.1, 2.1, 9, 1, 4, 3.3.2

[JLT25] Arun Jambulapati, Jonathan Li, and Kevin Tian. Radial isotropic position via an implicit newton's method. *CoRR*, abs/2504.05687, 2025. 3, 3.3.2

[JY11] Rahul Jain and Penghui Yao. A parallel approximation algorithm for positive semidefinite programming. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 463–471. IEEE Computer Society, 2011. 2.1

[Kad52] Richard V. Kadison. A generalized schwarz inequality and algebraic invariants for operator algebras. *Annals of Mathematics*, 56(3):494–503, 1952. 3.1.1, 3.3.1

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Comb.*, 4(4):373–396, 1984. 1.1

[KDRZ23] Matthäus Kleindessner, Michele Donini, Chris Russell, and Muhammad Bilal Zafar. Efficient fair PCA for fair representation learning. In *International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5250–5270. PMLR, 2023. 4

[Kha80] Leonid G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980. 1.1

[KHFM22] Mohammad Mahdi Kamani, Farzin Haddadpour, Rana Forsati, and Mehrdad Mahdavi. Efficient fair principal component analysis. *Mach. Learn.*, 111(10):3671–3702, 2022. 4

[KL20] Ferath Kherif and Adeliya Latypova. Principal component analysis. In *Machine Learning*, pages 209–225. Academic Press, 2020. 2.3

[KLM+17] Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. *SIAM J. Comput.*, 46(1):456–477, 2017. 3.2.3

[KMM+20] Michael Kapralov, Aida Mousavifar, Cameron Musco, Christopher Musco, Navid Nouri, Aaron Sidford, and Jakab Tardos. Fast and space efficient spectral sparsification in dynamic streams. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1814–1833. SIAM, 2020. 3.2.3

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. 1.1

[KSJ18] Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of newton's method without strong-convexity or lipschitz gradients. *CoRR*, abs/1806.00413, 2018. 3.4

[LCB+04] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004. 1.1

[LKJO22] Xiyang Liu, Weihao Kong, Prateek Jain, and Sewoong Oh. DP-PCA: statistically optimal and differentially private PCA. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022. 4

[LM00] Béatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000. 3.5.4

[LN93] Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 448–457. ACM, 1993. 2.1

[LS17] Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 678–687. ACM, 2017. 1.1

[LSW00] Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Comb.*, 20(4):545–568, 2000. 3.2

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 1049–1065. IEEE Computer Society, 2015. 1.1

[Mac08] Lester W. Mackey. Deflation methods for sparse PCA. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 1017–1024. Curran Associates, Inc., 2008. 4

[Mav22] Masiala Mavungu. Computation of financial risk using principal component analysis. *Algorithmic Finance*, 10(1-2):1–20, 2022. 2.3

[MRWZ16] Michael W. Mahoney, Satish Rao, Di Wang, and Peng Zhang. Approximating the solution to mixed packing and covering lps in parallel o∼(epsilonˆ{-3}) time. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, volume 55 of *LIPIcs*, pages 52:1–52:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. 2, 2.1, 2.2.1

[MS16] Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 814–827. ACM, 2016. 1.1

[Nes08] Yurii E. Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optim. Methods Softw.*, 23(1):109–128, 2008. 2.1

[NN94] Yurii E. Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *Siam studies in applied mathematics*. SIAM, 1994. 1.1

[PST95] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20(2):257–301, 1995. 2.1

[PTZ16] Richard Peng, Kanat Tangwongsan, and Peng Zhang. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. *CoRR*, abs/1201.5135v3, 2016. 2.1, 3.3.2

[Qua21] Kent Quanrud. Spectral sparsification of metrics and kernels. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 1445–1464. SIAM, 2021. 3.2.3

[Roc70] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970. A

[Rou20] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020. 3.3.3

[RSL18] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 10900–10910, 2018. 1.1

[Sch13] Kathrin Schacke. On the kronecker product, 2013. 3.4.1

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. 3.3.2

[SST06] Arvind Sankar, Daniel A. Spielman, and Shang-Hua Teng. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM J. Matrix Anal. Appl.*, 28(2):446–476, 2006. 3.3.3

[ST04] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004. 3.2.3, 3.3.3

[ST14] Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, 2014. 3.3.2

[STM+18] Samira Samadi, Uthaipon Tao Tantipongpipat, Jamie Morgenstern, Mohit Singh, and Santosh S. Vempala. The price of fair PCA: one extra dimension. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 10999–11010, 2018. 4

[SV19] Damian Straszak and Nisheeth K. Vishnoi. Maximum entropy distributions: Bit complexity and stability. In *Conference on Learning Theory, COLT 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 2861–2891. PMLR, 2019. 3.2, 3.2.1

[Sza91] Stanislaw J. Szarek. Condition numbers of random matrices. *Journal of Complexity*, 7(2):131–149, 1991. 3.2

[Vai96] Pravin M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Math. Program.*, 73:291–341, 1996. 1.1

[Ver24] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2024. 3.5.3, 3.4

[Wat93] G. A. Watson. On matrix approximation problems with ky fan$k$ norms. *Numer. Algorithms*, 5(5):263–272, 1993. 2.1.1

[WGR+09] John Wright, Arvind Ganesh, Shankar R. Rao, YiGang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 2080–2088. Curran Associates, Inc., 2009. 2.1.1

[You01] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *42nd Annual Symposium on Foundations of Computer*

Science, FOCS 2001, pages 538–546. IEEE Computer Society, 2001. 2.1

[Zha20] Richard Y. Zhang. On the tightness of semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. 1.1

[ZX18] Hui Zou and Lingzhou Xue. A selective overview of sparse principal component analysis. *Proc. IEEE*, 106(8):1311–1320, 2018. 4