Linear-Sized Sparsifiers via
Near-Linear Time Discrepancy Theory

Arun Jambulapati[*]   Victor Reis[†]   Kevin Tian[‡]

Abstract

Discrepancy theory has provided powerful tools for producing higher-quality objects which "beat the union bound" in fundamental settings throughout combinatorics and computer science. However, this quality has often come at the price of more computationally-expensive algorithms. We introduce a new framework for bridging this gap, by allowing for the efficient implementation of discrepancy-theoretic primitives. Our framework repeatedly solves regularized optimization problems to low accuracy to approximate the partial coloring method of [Rot17], and simplifies and generalizes recent work of [JSS23] on fast algorithms for Spencer's theorem. In particular, our framework only requires that the discrepancy body of interest has exponentially large Gaussian measure and is expressible as a sublevel set of a symmetric, convex function. We combine this framework with new tools for proving Gaussian measure lower bounds to give improved algorithms for a variety of sparsification and coloring problems.

As a first application, we use our framework to obtain an $\widetilde{O}(m \cdot \epsilon^{-3.5})$ time algorithm for constructing an ϵ-approximate spectral sparsifier of an m-edge graph, matching the sparsity of [BSS14] up to constant factors and improving upon the $\widetilde{O}(m \cdot \epsilon^{-6.5})$ runtime of [LS17]. We further give a state-of-the-art algorithm for constructing graph ultrasparsifiers and an almost-linear time algorithm for constructing linear-sized degree-preserving sparsifiers via discrepancy theory; in the latter case, such sparsifiers were not known to exist previously. We generalize these results to their analogs in sparsifying isotropic sums of positive semidefinite matrices. Finally, to demonstrate the versatility of our technique, we obtain a nearly-input-sparsity time constructive algorithm for Spencer's theorem (where we recover a recent result of [JSS23]).

---

[*]University of Washington, `jmblpati@uw.edu`

[†]University of Washington, `voreis@cs.washington.edu`

[‡]Microsoft Research, `tiankevin@microsoft.com`

Contents

1 Introduction

Throughout the history of theoretical computer science and combinatorics, the development of discrepancy theory has yielded techniques for producing high-quality objects which minimize deviation from typical behavior. As a well-known example which has received significant recent attention, Spencer's "six standard deviations suffice" theorem [Spe85] says that for any matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, whose rows indicate n sets $S_i \subseteq [n]$, there exists at least one coloring $x \in \{-1,1\}^n$ such that $\|\mathbf{A}x\|_\infty \leq 6\sqrt{n}$; we remark that the problem has a combinatorial interpretation as choosing colors to minimize the largest color discrepancy in any set. Notably, this result improves upon the quality of a random coloring x, which achieves $\|\mathbf{A}x\|_\infty = \Theta(\sqrt{n \log n})$ with high probability for random set systems, by standard concentration and anti-concentration arguments. Discrepancy-theoretic arguments of a similar nature have led to significant progress in a variety of fields spanning computational geometry, complexity theory, approximation algorithms, numerical analysis, graph theory, and experimental design [Mat99, Cha01, BSS14, HSSZ19, ZBG$^+$22].

However, the development of computationally-efficient algorithms for discrepancy minimization is comparatively rather nascent. To use Spencer's theorem as an example, while [Spe85] shows the existence of low-discrepancy colorings, these colorings are highly atypical and it is not immediately clear how to find one algorithmically. Roughly a decade ago, a sequence of works [Ban10, LM15, HSS14, Rot17, ES18] gave *constructive discrepancy* algorithms in settings including Spencer's set coloring problem and variants thereof, settling the polynomial-time computability of high-quality colorings of set systems. The works of [Rot17, ES18] in particular provided very general frameworks for discrepancy minimization. For example, given a symmetric, convex "discrepancy body" $\mathcal{K}$ (e.g. $\mathcal{K} := \{x \in \mathbb{R}^n \mid \|\mathbf{A}x\|_\infty = O(\sqrt{n})\}$ in Spencer's setting) with an $\exp(-O(n))$ Gaussian measure lower bound, [Rot17] showed that the nearest point in $\mathcal{K} \cap [-1,1]^n$ to a random Gaussian vector has a constant fraction of tight hypercube constraints with high probability. A similar statement was shown by [ES18] for the solution to a suitable linear program, and these "partial coloring by Gaussian rounding" subroutines naturally induce full discrepancy minimization algorithms via recursion.

While the constructive frameworks of [Ban10, LM15, HSS14, Rot17, ES18] are elegant and simple to state, their straightforward implementation requires the black-box use of powerful convex programming primitives such as high-accuracy linear or semidefinite programming solvers, which incur substantial runtime overhead. This has led to a recurring theme in algorithmic discrepancy theory: attaining higher object quality may come at the price of worse computational efficiency.

A recent exciting work by [JSS23] was able to overcome this hurdle for the specific setting of Spencer's theorem, by leveraging *approximate* linear programming solvers running in near-linear time. Specifically, [JSS23] designed an algorithm which produces a coloring $x \in \{-1,1\}^n$ satisfying $\|\mathbf{A}x\|_\infty = O(\sqrt{n})$ in time $\widetilde{O}(\mathrm{nnz}(\mathbf{A}) + n)$, by efficiently solving the linear programs required by the [ES18] framework and quantifying their approximation tolerance. However, the [JSS23] analysis relied on properties tailored to the setting of Spencer's theorem, specifically an exponential lower bound on the probability that a random coloring achieves $\|\mathbf{A}x\|_\infty = O(\sqrt{n})$. Such a statement is not known to hold in various other discrepancy-theoretic settings of interest, such as the Komlós problem, even when the corresponding Gaussian measure lower bound required by the [Rot17, ES18] frameworks is known. Our work is motivated by the current lack of a general-purpose, approximation-tolerant framework for discrepancy minimization under the minimal requirements of symmetry, convexity, and a Gaussian measure lower bound, which may pave the way towards near-linear time algorithms in discrepancy-theoretic settings beyond Spencer's theorem.

1.1 Our results

Discrepancy minimization framework. We introduce a new *approximation-tolerant* variant of the Gaussian rounding framework of [Rot17]. Before stating its guarantees, we first briefly recall the main result of [Rot17], deferring a more extended discussion to Section 3. Given a symmetric convex set[1] $\mathcal{K} \subseteq \mathbb{R}^m$ with $\gamma_m(\mathcal{K}) = \exp(-O(m))$ (see Section 2 for notation), Theorem 2 of [Rot17] shows that with high probability over a randomly sampled Gaussian vector $g \sim \mathcal{N}(\mathbb{0}_m, \mathbf{I}_m)$, letting

$$x_\star := \arg \min_{x \in \mathcal{K} \cap [-1,1]^m} \|x - g\|_2,$$

we have $|\{i \mid |[x_\star]_i| = 1\}| = \Omega(m)$. In other words, by rounding g to the intersection of $\mathcal{K}$ and $[-1, 1]^m$, a constant fraction of the hypercube constraints $|[x_\star]_i| \leq 1$ are saturated. When $\mathcal{K}$ is taken to be a discrepancy body, e.g. $\mathcal{K} = \{x \in \mathbb{R}^m \mid \|\mathbf{A}x\|_\infty = O(\sqrt{m})\}$ in Spencer's setting, the result of [Rot17] implies that we can freeze a constant fraction of colors at ± 1 without incurring much discrepancy, and recurse on the uncolored coordinates. High-precision computation of $x_\star$ is further polynomial-time tractable, as it is the solution to a convex optimization problem.

Our approximation-tolerant framework also applies to any symmetric, convex $\mathcal{K} \subseteq \mathbb{R}^m$ with large $\gamma_m(\mathcal{K})$ (as in [Rot17]), but is most easily described when

$$\mathcal{K} = \{x \in \mathbb{R}^m \mid f(x) \leq \rho\},$$

where $\mathcal{K}$ is expressed as a sublevel set of a symmetric, convex function $f : \mathbb{R}^m \to \mathbb{R}$, and $\rho \geq 0$ is a "discrepancy radius." All symmetric convex sets trivially satisfy this description by letting $\rho = \infty$ and f be the indicator function of the set, but in many applications of interest (explored throughout the paper), $\mathcal{K}$ is naturally characterized as a sublevel set with finite ρ. In Section 3, we show that for a parameter $\beta \in (0, 1)$, solving $\widetilde{O}(1)$ regularized problems of the form[2]

$$\min_{x \in [-1,1]^m} f(x) + \lambda \|x - g\|_2^2$$

to additive accuracy $\Theta(\rho\beta^2)$ suffices to produce a point $x \in \mathcal{K} \cap [-1, 1]^m$ with $|\{i \mid |x_i| \geq 1 - \beta\}| = \Omega(m)$. In other words, by approximately solving a sequence of regularized optimization problems to accuracy $\Theta(\rho\beta^2)$, our framework yields a point with low discrepancy and many nearly-saturated hypercube constraints. In applications such as Spencer's setting, this additive accuracy is fairly generous: taking β to be inverse-polylogarithmic means we only need to optimize the regularized objective to error $\approx \sqrt{m}$, which is efficiently implementable in input-sparsity time (after applying reductions from [JSS23]) via stochastic first-order methods. Furthermore, these nearly-saturated coordinates of our approximate solution x can then be randomly rounded to exactly be in $\{\pm 1\}$ without incurring much additional discrepancy. In our graph-theoretic applications of our approximation-tolerant framework, we handle the nearly-saturated coordinates recursively.

As a direct application of our framework in Section 3, we show how to combine it with existing stochastic optimization methods from [CJST20] to recover the main result of [JSS23] through arguably a simpler approach, which does not rely on structural properties of Spencer's problem beyond a Gaussian measure lower bound. This result is provided in Section 8.

---

[1]Throughout the exposition in this section, we let the dimensionality of our coloring variable x be denoted m for consistency with our graph-theoretic applications, and state the implications of our results in Spencer's setting (whose dimension is typically denoted by n) consistently in an abuse of notation.

[2]Here and throughout the paper, we use the notation $\widetilde{O}(\cdot)$ to hide polynomial factors in $\log n$ and $\log(\frac{1}{\varepsilon})$ and $\breve{O}(\cdot)$ to hide polynomial factors in $\log \log n$ and $\log \log(\frac{1}{\varepsilon})$. We defer our notational conventions to Section 2.

Linear-sized sparsifiers. We next illustrate how our framework allows us to leverage powerful discrepancy-theoretic tools to design fast algorithms for the problem of graph sparsification. Given a parameter $\varepsilon \in (0,1)$ and a graph G on n vertices and m edges, with Laplacian $\mathbf{L}_G \in \mathbb{R}^{n \times n}$, the goal is to find a reweighted subgraph H of G with much fewer edges and Laplacian $\mathbf{L}_H$, satisfying

$$(1 - \varepsilon)\mathbf{L}_G \preceq \mathbf{L}_H \preceq (1 + \varepsilon)\mathbf{L}_G.$$

A well-known result of [SS11] shows that randomly sampling edges of G proportionally to their effective resistances yields a reweighted subgraph H satisfying the above guarantee with $O(n \log n \cdot \varepsilon^{-2})$ edges. The proof of correctness of such a sampling strategy is via a direct application of a matrix Chernoff concentration bound. Further, by using sketching techniques and specialized linear system solvers, the [SS11] sparsifiers can be constructed in nearly-linear time $\widetilde{O}(m)$. However, this size bound is not optimal: a breakthrough work by [BSS14] gave a polynomial-time algorithm for constructing a spectral sparsifier H with $O(n \cdot \varepsilon^{-2})$ edges, later shown to be tight [CKST19]. We call such a subgraph H a linear-sized sparsifier (an accurate description for constant ε).

The [BSS14] sparsification result is discrepancy-theoretic in nature, as it improves the quality of a randomly sampled sparsifier (where the quality is the number of edges). Interestingly, however, the techniques used to construct linear-sized sparsifiers are quite different than those employed by the aforementioned works [Ban10, LM15, HSS14, Rot17, ES18]. Indeed, the original announcement of [BSS14] predates all of these works, so at the time Spencer's theorem was not even algorithmic. Instead, the sparsifier of [BSS14] is constructed one edge at a time, guided by the careful use of a potential function which controls the discrepancy. A sequence of follow-up works [ZLO15, LS18] culminated in the current state-of-the-art algorithm of [LS17], which obtains a linear-sized sparsifier in time $\widetilde{O}(m \cdot \varepsilon^{-6.5})$, using structured semidefinite program solvers to efficiently implement updates against a modification of [BSS14]'s potential based on the trace of a matrix exponential.

We give an improved algorithm for linear-sized sparsification via our framework. We prove the following in Section 5 as a special case of a result on sparsifying isotropic matrix sums (Theorem 5). Our result almost quadratically improves upon the ε dependence of [LS17].

Theorem 1. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0,1)$, there is a randomized algorithm which in time $\breve{O}(m \log^4(m) \log(\frac{1}{\varepsilon}) \cdot \varepsilon^{-3.5})$ returns $w \in \mathbb{R}_{\geq 0}^E$ satisfying $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$, such that with high probability in n,*

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

We remark that Theorem 1 follows from an instantiation of our framework in Section 3, using an approximate Laplacian linear system solver of [KMP11] and a recent approximate semidefinite program solver from [JT23] to solve the resulting subproblems. Both of these results lie in active areas of research, and improvements therein immediately imply faster algorithms for linear-sized sparsification via our framework. We find it a promising proof-of-concept that we rely on more conventional discrepancy-theoretic techniques (motivated by [Rot17, RR20]) to establish Theorem 1, eschewing conventional wisdom that the use of such techniques comes at a computational price.

To our knowledge, no analog of the structural fact used in [JSS23] (where the set of sparse reweightings which spectrally approximate G has inverse-exponentially large measure) is known in the setting of spectral sparsification. In demonstrating Theorem 1, it is thus important that our framework only relies on measure lower bounds known for the relevant operator norm balls.

Ultrasparsifiers. Ultrasparsifiers were introduced in [ST04] to dramatically reduce the number of edges in a graph at a cost of a larger approximation factor, and can be viewed as "one-sided $\varepsilon > 1$" variants of the sparsifiers in our earlier discussion. We define ultrasparsifiers as follows.

Definition 1 (Graph ultrasparsifiers). *Let G be a graph on n vertices. A graph H is an (κ, ℓ)-ultrasparsifier of G if it has $n - 1 + \frac{n}{\ell}$ edges and satisfies*

$$\mathbf{L}_H \preceq \mathbf{L}_G \preceq \kappa \mathbf{L}_H.$$

In general, taking $\ell \to n$ and the fact that no tree is better than an $O(n)$ approximation of a clique implies the tightest ultrasparsifiers we can hope for in all regimes must have $\kappa = \Omega(\ell)$. The first construction of graph ultrasparsifiers was achieved by [ST04], who gave a nearly-linear time algorithm to construct $(\ell \log^{O(1)}(n), \ell)$-ultrasparsifiers for any $\ell \geq 1$. By applying a matrix Chernoff-based random sampling argument, [KMP11] then gave a simpler construction of $(\breve{O}(\ell \log^2(n)), \ell)$-ultrasparsifiers in nearly-linear time. Further, using techniques inspired by the [BSS14] potential-based argument, higher-quality ultrasparsifiers are known to be constructible in polynomial time: [KMST10] constructed $(\breve{O}(\ell \log(n)), \ell)$-ultrasparsifiers based on state-of-the-art low-stretch spanning trees, and recently [JS21] constructed $(\ell^{1+o(1)} \log^{o(1)}(n), \ell)$-ultrasparsifiers by going beyond the low-stretch spanning tree framework. To the best of our knowledge, it is unknown how to adapt the tools developed for fast linear-sized sparsification in [ZLO15, LS18, LS17] to the ultrasparsifier setting, limiting the algorithmic uses of the higher-quality ultrasparsifiers of [KMST10, JS21].

In Section 6, we prove Theorem 6, a generalization of Theorem 1 which extends our discrepancy minimization techniques to the ultrasparsifier setting. In fact, Theorem 6 follows as a corollary of Theorem 1 in light of a new discrepancy-theoretic result we prove as Theorem 4. We then apply Theorem 6 to obtain the following ultrasparsifier construction, matching the guarantees of [KMST10, JS21] in nearly-linear time and potentially paving the way for their use in fast algorithms.

Corollary 1. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\ell \geq 1$, there is a randomized algorithm which in time $\breve{O}(m \log^4(m))$ returns a $(\min(\breve{O}(\ell \log(n)), \ell^{1+o(1)} \log^{o(1)}(n)), \ell)$-ultrasparsifier of G, with high probability in n.*

Degree-preserving sparsifiers. As a final application of our framework, we turn our attention to degree-preserving sparsification, a primitive introduced by [CGP+18] which asks for a spectral sparsifier that also preserves the weighted degrees of all vertices. Degree-preserving sparsification has arisen as a natural middle ground between undirected graph sparsification and directed graph sparsification. Concretely, Eulerian sparsifiers of Eulerian graphs have become the de facto notion of directed graph sparsification due to e.g. [CKP+16, PS22], which show how to reduce linear system solving in directed graph Laplacians to Eulerian sparsification. A technique known as short-cycle decomposition, introduced in [CGP+18] and refined in [LSY19, PY19], was previously used to obtain efficient constructions of both (undirected) degree-preserving sparsifiers and (directed) Eulerian sparsifiers by repeatedly performing degree-preserving operations. This motivates the study of degree-preserving sparsification as a stepping stone towards more complex notions of sparsifiers.

Perhaps surprisingly, even the existence of linear-sized degree-preserving sparsifiers (with $O(n \cdot \varepsilon^{-2})$ edges) is not known. The state-of-the-art existential result in the literature is degree-preserving sparsifiers with $O(n \log^2 n \cdot \varepsilon^{-2})$ edges, constructible in polynomial time [CGP+18, PY19]; we note that [PY19] also gives an almost-linear time construction of $O(n \log^3 n \cdot \varepsilon^{-2})$-sized degree-preserving sparsifiers.[3] In Section 7, we give the following improved degree-preserving sparsification result.

Theorem 2. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0, 1)$, there is a randomized algorithm which returns $w \in \mathbb{R}_{\geq 0}^E$ satisfying $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$, $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$ in time*

$$O\left((m + \mathcal{T}_{\mathrm{route}}(\alpha)) \cdot \alpha^2 \cdot \mathrm{poly}\left(\frac{\log m}{\varepsilon}\right) \right),$$

---

[3]We additionally remark that a size bound of $O(n \log n \cdot \varepsilon^{-2})$ is known as folklore in the community: it follows from combining the short-cycle decomposition of [CGP+18] with [BSS14].

for any $\alpha \geq 1$ following the notation of Definition 5, such that with high probability in n,

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

The runtime of Theorem 2 is parameterized by the quality and runtime of state-of-the-art *oblivious routings*, defined formally in Definition 5. As we recall in Proposition 10, current oblivous routings [KLOS14] achieve $\mathcal{T}_{\text{route}}(\alpha) = mn^{o(1)}$ for $\alpha = n^{o(1)}$, so Theorem 2 runs in almost-linear time $m^{1+o(1)} \cdot \text{poly}(\varepsilon^{-1})$. However, polylogarithmic-quality *congestion approximators* are known to be constructible in time $m \cdot \text{polylog}(n)$ [Pen16], which are closely-related to oblivous routings; any improvements on oblivious routing quality would then be reflected in Theorem 2 as well.

Finally, we note that expander decompositions (and the fact that electric routings are good oblivious routings for expanders) can improve the runtime of Theorem 2, at the cost of a logarithmic overhead in the sparsity. We give an algorithm achieving this runtime-sparsity tradeoff as Theorem 7.

1.2 Technical overview

Gaussian rounding via regularized optimization. Consider the setting of the [Rot17] framework, where symmetric, convex $\mathcal{K} \subseteq \mathbb{R}^m$ has $\exp(-O(m))$ Gaussian measure. Further, let $g \sim \mathcal{N}(\mathbb{0}_m, \mathbf{I}_m)$, and $x_\star \in \mathcal{K} \cap [-1,1]^m$ minimize the distance to g, (so $x_\star$ has a constant fraction of coordinates at ± 1). Our starting point is the observation that if $x \in \mathcal{K} \cap [-1,1]^m$ satisfies $\|x - x_\star\|_2^2 = O(m\beta^2)$ for an appropriate constant, then it must have many coordinates in $[-1, -1 + \beta] \cup [1 - \beta, 1]$; a simple proof is provided in Lemma 1. Moreover, strong convexity of $\|x - g\|_2^2$ then implies that to achieve this distance bound, it suffices to find $x \in \mathcal{K} \cap [-1,1]^m$ minimizing the squared distance to g up to $O(m\beta^2)$, formalized in Lemma 2. In settings where $\mathcal{K} = \{x \in \mathbb{R}^m \mid f(x) \leq \rho\}$, we wish to find x achieving both small function value (according to f) and distance to g. This motivates the consideration of regularized mixed objectives of the form

$$f_\lambda(x) := f(x) + \lambda \|x - g\|_2^2.$$

The bulk of Section 3 then develops a binary-search procedure which approximately minimizes f_λ over $[-1,1]^m$ for different values of λ, and aggregates these solutions to achieve the desired distance to $x_\star$. We give a formal statement in Proposition 2 quantifying the range of λ under consideration by our search, as well as the accuracy levels required by our subproblem solvers (for optimizing f_λ).

New tools for lower bounding Gaussian measure. To apply the framework of Section 3, we require methods for proving Gaussian measure lower bounds for various discrepancy bodies. For Spencer's setting, since the discrepancy body is a polyhedron with $O(m)$ facets, a Gaussian measure lower bound follows from a routine application of the Sidak-Khatri correlation inequality (see e.g. Lemma 8.9, [Rot16]). For non-polyhedral sets such as the operator norm balls arising in spectral sparsification settings, however, naïve applications of Sidak-Khatri fall short. This bottleneck was alleviated in part by [RR20], who proved that a different sufficient condition for the [Rot17] framework holds for operator norm balls $\mathcal{K}$ arising in spectral sparsification settings, i.e. $\mathcal{K} = \{x \in \mathbb{R}^m \mid \|\sum_{i \in [m]} x_i \mathbf{A}_i\|_{\text{op}} \leq \rho\}$ for matrices $\{\mathbf{A}_i\}_{i \in [m]}$. Roughly speaking, the main result of [RR20] shows that for any α, with probability $\geq \frac{1}{2}$ a random Gaussian vector lies at distance $O(\alpha\sqrt{m})$ from $\frac{1}{\alpha}\mathcal{K}$ (when $\rho = \sqrt{n/m}$). This was proven by conducting a controlled random walk, where the output of the walk is close to a Gaussian and achieves good discrepancy. The same paper asked whether a direct Gaussian measure lower bound holds for the relevant operator norm balls.

In Section 4, we augment [RR20] with two new results of potentially independent interest. The first, Theorem 3, generically reduces Gaussian measure lower bounds for convex sets to statements of the form proven by [RR20], and thus expands the Gaussian measure toolkit for non-polyhedral sets. We use this result in conjunction with an extension of [Rot17] due to [RR22], which tolerates intersections with a subspace, in our degree-preserving sparsification algorithms in Section 7. The second, Theorem 4, generalizes the main result of [RR20] when the matrix set of interest is parameterized by an additional trace bound, and is crucial to our ultrasparsifier algorithms in Section 6.

Spectral sparsification and box-spectraplex games. In the remainder of the paper, we focus on applications of our framework to spectral sparsification. Our algorithms for linear-sized sparsifiers and ultrasparsifiers are essentially identical in light of Theorem 4, so we discuss the former. We begin by observing that the subproblems required from our framework, when $\mathcal{K}$ is an operator-norm ball, are structured semidefinite programs. In particular, they are expressible as *box-spectraplex games*, which are bilinear minimax optimization problems between a min player living in the box $[-1,1]^m$, and a max player living in the spectraplex $\{\mathbf{Y} \in \mathbb{R}^{n \times n} \mid \mathrm{Tr}(\mathbf{Y}) = 1, \mathbf{Y} \succeq \mathbf{0}_n\}$. We adapt a recent nearly-linear time approximation algorithm for box-spectraplex games from [JT23] to solve our subproblems to sufficiently high accuracy for the linear-sized sparsification recursion in [RR20] to work out, up to a small error term which can be discarded. By using random initializations, we only lose polyloglog factors in runtime over the solver of [JT23]. The only additional overhead in Theorem 1 is from solving a Laplacian linear system to bring the problem into isotropic position.

Executing this strategy for degree-preserving sparsification requires more care: we need to make sure all of our approximate solvers and intermediate rounding steps preserve the degree constraints. We first discuss how to solve the subproblems required by our framework. We show that a Frank-Wolfe method analyzed in the ℓ_∞ norm reduces the optimization problem to a small number of approximate linear optimization problems over the intersection of $[-1,1]^m$ and circulation space (the kernel of a reweighted $\mathbf{B}^\top$). We solve these linear optimization problems using oblivous routings (Definition 5), which both parameterize circulation space and precondition the optimization problem. Finally, to handle the rounding error due to the recursion scheme, we develop a key subroutine implemented in nearly-linear time using dynamic data structures [ST83]. Our subroutine lets us round a small flow on a bipartite graph efficiently, zeroing out a constant fraction of its weights while preserving degrees, and without more than doubling any edge weight.

1.3   Related work

Algorithmic discrepancy theory. The theoretical computer science community has developed a number of constructive discrepancy-theoretic proofs over the past decade, in the form of polynomial-time algorithms. It is out of our scope to survey this wide body of work, but beyond those discussed earlier on linear-sized sparsification and Spencer's theorem, we refer the reader to additional examples in [BDG19, Coh16, BDGL19, DNTT18, BJSS20, ALS21, BLV22, PV23]. We further find it interesting to develop an approximate implementation of the framework of [ES18] based on convex programming under minimal assumptions, analogous to our implementation of [Rot17]. Finally, of particular relevance to the themes of our paper are two recent algorithms for hereditary discrepancy minimization in set sytems by [Lar23, DSW22]. The algorithm of [DSW22] runs in input-sparsity time (in the indicator matrix) in some regimes. We find it potentially fruitful to explore improving these runtimes, and broadening the range of problems to which their tools are applicable.

Approximate semidefinite programming. Our sparsification algorithms in Sections 5, 6, and 7 all rely on approximate solvers [JT23] for structured semidefinite programs expressible as box-

spectraplex games. Such problems naturally arise from our framework applied to operator norm balls, as the minimization player lives in $[-1, 1]^m$ (the box), and the dual best response lives in the spectraplex. We hence find investigating these games interesting, as any improvements to [JT23] would reflect in our runtimes, and those of any future applications. For example, could we improve runtimes via low-rank sketches which have found success in simplex-spectraplex games [KV05, BBN13, GHM15, AL17, CDST19]? Improvements beyond rank reduction are also interesting, as the matrix Spencer setting (which was nearly-resolved constructively in [BJM23]) induces an operator norm ball, but the guarantees of current solvers lose $\text{poly}(n)$ factors in the runtime there.

2 Preliminaries

General notation. We let $\widetilde{O}$ suppress polylogarithmic factors in problem parameters, and we let $\breve{O}$ suppress polyloglogarithmic factors in problem parameters. We use $[n]$ to denote $\{i \in \mathbb{N} \mid i \leq n\}$. The ℓ_p norm of a vector argument is $\|\cdot\|_p$. The all-zeroes and all-ones vectors of dimension d are $\mathbb{0}_d$ and $\mathbb{1}_d$. For $S \subseteq [d]$ where d is clear from context, $\mathbb{1}_S \in \{0,1\}^d$ is the vector which is 1 for coordinates in S. We say vector v is s-sparse if at most s of its entries are nonzero. For vectors u, v of equal dimension, $u \circ v$ is their coordinatewise product. We let $\mathbb{B}_2^d := \{v \in \mathbb{R}^d \mid \|v\|_2 \leq 1\}$, and for sets $\mathcal{K}, \mathcal{K}' \subset \mathbb{R}^d$, $\mathcal{K} + \mathcal{K}'$ is their Minkowski sum. For convex sets $A, B \subseteq \mathbb{R}^d$, $N(A, B)$ is their covering number, the fewest number of translates of B needed to cover A. We let $\mathcal{N}(\mu, \mathbf{\Sigma})$ be the Gaussian distribution of specified mean and covariance, and $\mathcal{N}_{\leq \tau}(\mu, \mathbf{\Sigma})$ sets any draw from $\mathcal{N}(\mu, \mathbf{\Sigma})$ with Euclidean norm more than τ to the zero vector. We use "with high probability in n" to mean an event succeeds with probability $1 - n^{-O(1)}$, for an arbitrary constant.

Matrices. Matrices are denoted in boldface. The $n \times n$ identity matrix and all-zeroes matrix are respectively $\mathbf{I}_n$ and $\mathbf{0}_n$. The number of nonzero entries of a matrix or a vector argument is $\text{nnz}(\cdot)$. The set of $n \times n$ real symmetric matrices (and respectively, real positive semidefinite and positive definite matrices) is denoted $\mathbb{S}^n$ (and respectively, $\mathbb{S}_{\succeq \mathbf{0}}^n$ and $\mathbb{S}_{\succ \mathbf{0}}^n$). We equip $\mathbb{S}^n$ with the trace inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}(\mathbf{AB})$ and Loewner partial ordering $\preceq$. The Frobenius, operator, and trace norms are denoted $\|\cdot\|_{\text{F}}$, $\|\cdot\|_{\text{op}}$, and $\|\cdot\|_{\text{tr}}$, and correspond to the 2-norm, ∞-norm, and 1-norm of the singular values of a matrix. For matrices $\{\mathbf{A}_i\}_{i \in [m]}$ of equal dimension clear from context, we associate a pair of operators $\mathcal{A}$ and $\mathcal{A}^*$, taking a vector and matrix argument respectively. We let

$$\mathcal{A}(x) := \sum_{i \in [m]} x_i \mathbf{A}_i \text{ and } \mathcal{A}^*(\mathbf{Y}) := \{\langle \mathbf{A}_i, \mathbf{Y} \rangle\}_{i \in [m]}. \tag{1}$$

We let $\mathcal{N}(\mu, \mathbf{\Sigma})$ denote the multivariate Gaussian with mean μ and covariance $\mathbf{\Sigma}$, and γ_d denotes the standard Gaussian density (of $\mathcal{N}(\mathbb{0}_d, \mathbf{I}_d)$) in dimension d. We let $\mathcal{T}_{\text{mv}}(\mathbf{A})$ denote the time it takes to compute $\mathbf{A}v$ for any vector v of appropriate dimension. Note that when $\mathbf{A}$ is explicit, $\mathcal{T}_{\text{mv}}(\mathbf{A}) = O(\text{nnz}(\mathbf{A}))$, where we let $\text{nnz}(\mathbf{A})$ denote the number of nonzero entries in $\mathbf{A}$. For any $\rho > 0$ and operator $\mathcal{A}$ associated with a matrix set, we denote the associated "discrepancy body" by

$$\mathcal{K}_{\rho, \mathcal{A}} := \{x \in \mathbb{R}^m \mid \|\mathcal{A}(x)\|_{\text{op}} \leq \rho\}. \tag{2}$$

We denote the condition number (ratio of largest and smallest eigenvalues) of $\mathbf{M} \in \mathbb{S}_{\succ \mathbf{0}}^n$ by $\kappa(\mathbf{M})$.

Graphs. We use the notation $G = (V, E, w_G)$ to denote an undirected graph G on vertex set V, where $w_G \in \mathbb{R}_{\geq 0}^E$ is the weight function on the edges E. When G is clear from context we let $\mathbf{W} := \mathbf{diag}(w_G) \in \mathbb{R}^{E \times E}$ and we let $|\mathbf{B}| \in \mathbb{R}^{E \times V}$ be the unsigned edge-vertex incidence matrix (i.e. with

two ones per row corresponding to incident vertices on an edge). We also let $\mathbf{B} \in \mathbb{R}^{E \times V}$ be the signed edge-vertex incidence matrix (where one of the ones per row is arbitrarily but consistently negated) and $\mathbf{L} := \mathbf{B}^\top \mathbf{W} \mathbf{B}$ be the graph Laplacian. We refer to the rows of $\mathbf{B}$ by $\{b_e\}_{e \in E} \in \{-1, 0, 1\}^V$. It is well-known that $x^\top \mathbf{L} x = \sum_{e=(u,v) \in E} [w_G]_e (x_u - x_v)^2$. The pseudoinverse of $\mathbf{L}$ (which has a kernel in the $\mathbb{1}_V$ direction, i.e. it preserves the kernel of $\mathbf{L}$) is denoted $\mathbf{L}^\dagger$. When multiple graphs are in discussion, we will subscript these matrices with G to disambiguate, i.e. $\mathbf{L}_G$ is the Laplacian of G. We denote the identity matrix on the subspace of $\mathbb{R}^V$ orthogonal to the $\mathbb{1}_V$ direction by $\mathbf{I}_{V \setminus \mathbb{1}}$.

3 Approximate partial coloring framework

In this section, we recall the partial coloring method for discrepancy minimization from [Rot17], and develop a binary search framework for approximating it efficiently. This framework will be our primary workhorse in all our applications. First, [Rot17] gave a generic result which transforms Gaussian measure lower bounds on a symmetric set $\mathcal{K} \subset \mathbb{R}^m$ to a high-probability lower bound on the number of "tight constraints" encountered by the closest point in $\mathcal{K} \cap [-1, 1]^m$ to a random Gaussian vector. We state a strengthening by [RR22], which works for any exponential measure lower bound, and also tolerates restriction to a linear subspace of sufficiently large dimension.

Proposition 1 (Theorem 6, [RR22])*. Let $\mathcal{K} \subseteq \mathbb{R}^m$ be symmetric and convex, and suppose there is a constant C such that $\gamma_m(\mathcal{K}) \geq \exp(-Cm)$. There are constants $c_{\mathrm{tight}} \in (0, 1)$ and $C_{\mathrm{set}} > 0$ such that if $S \subseteq \mathbb{R}^m$ is a linear subspace of dimension at least $2c_{\mathrm{tight}} m$, $g \sim \mathcal{N}(\mathbb{0}_m, \mathbf{I}_m)$, and x_g is the closest point to g in $C_{\mathrm{set}} \mathcal{K} \cap [-1, 1]^m \cap S$, then with probability $1 - \exp(-\Omega(m))$, at least $c_{\mathrm{tight}} m$ coordinates of x_g have magnitude 1.*

Proposition 1 is often referred to as a partial coloring result, as it fixes a number of coordinates of x to ± 1 ("colors"), and the remaining coordinates can be recursively handled. When the set $\mathcal{K}$ is a discrepancy body (as it will be throughout this paper), this recursive application then leads to discrepancy minimization algorithms. However, in this section we state our framework for a more general abstract setting, which may be of further utility. Our framework yields an *approximate* partial coloring result, i.e. it will only guarantee that x has many coordinates near ± 1, but has the benefit that in settings of interest it can be implemented efficiently via customized solvers.

Specifically, we consider the setting where $\mathcal{K}$ is a sublevel set of a nonnegative, symmetric convex function $f : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ (i.e. $f(x) = f(-x)$ for all x), and we are given a "discrepancy radius" $\rho \geq 0$ fixed throughout. We then define our set of interest

$$\mathcal{K} := \{x \in \mathbb{R}^m \mid f(x) \leq \rho\}. \tag{3}$$

We also assume that we are given a vector $g \in \mathbb{R}^m$ fixed throughout, which will ultimately be a random Gaussian vector. We use the following assumption which holds except with exponentially small probability $\exp(-\Omega(m))$ by standard Gaussian concentration.

Assumption 1. *We assume that $\|g\|_2 \leq 2\sqrt{m}$.*

We are now ready to describe our binary search framework. In light of Proposition 1, our goal is to approximately compute the value $r_\star$ and point $x_\star$ defined as

$$r_\star := \|x_\star - g\|_2, \text{ where } x_\star := \arg \min_{x \in [-1,1]^m \cap \mathcal{K}} \|x - g\|_2, \tag{4}$$

and $\mathcal{K}$ is defined as in (3). We remark that our framework extends to handle linear constraints as suggested by Proposition 1, e.g. $\mathbf{A}x = b$, by optimizing over $[-1, 1]^m \cap \{x \mid \mathbf{A}x = b\}$. However, for

simplicity of exposition we focus our attention to the case of optimizing over a hypercube in this section, and give a formal application of our framework which handles additional linear constraints in Section 7. Our approximate partial coloring methods are developed as follows.

1. We show x approximating $x_\star$ up to a distance $\approx \sqrt{m}\beta$ for an approximation tolerance $\beta > 0$, set differently in each specific instantiation, suffices for the partial coloring frameworks of e.g. [Rot17, RR20] up to a rounding error component (Lemma 1). We typically handle this error recursively. We further exploit strong convexity of the squared distance function to show that approximately minimizing the objective in (4) suffices for this goal (Lemma 2).

2. We develop an efficient binary-search based reduction from approximating $r_\star$ in (4) to approximating the values of a sequence of regularized convex optimization problems which balance $f(x)$ with a multiple of the squared distance to g (Proposition 2).

3. We approximate the solutions of these regularized optimization problems in each of our applications using customized solvers. For sparsification-related settings, these solvers will usually be modifications of the recent box-spectraplex game solver of [JT23].

Item 3 of our framework will be treated differently in each of our applications, so in this section we focus on providing results encapsulating Items 1 and 2 for ease of use.

Approximating $x_\star$ suffices. We first observe that in applications of Proposition 2, any feasible point $x \in [-1, 1]^m \cap \mathcal{K}$ which is sufficiently close to $x_\star$ must have many nearly-tight coordinates. In our applications we will apply Proposition 2 to both g and $-g$ for a random Gaussian g, and f in (3) will always be symmetric. So, without loss of generality we state our results when $x_\star$ defined in (4) has at least a constant fraction of coordinates at -1, which will be convenient later.

Lemma 1. *Suppose $x_\star \in [-1, 1]^m$ is -1 in at least $\frac{c_{\text{tight}} m}{2}$ coordinates, and $x \in [-1, 1]^m$ satisfies*

$$\|x - x_\star\|_2^2 \leq \frac{c_{\text{tight}} m \beta^2}{4}.$$

Then, at least $\frac{c_{\text{tight}} m}{4}$ coordinates x_i satisfy $x_i \leq -1 + \beta$.

Proof. Assume otherwise, and let $I \subseteq [m]$ be the set of coordinates where $[x_\star]_i = -1$. More than half of the coordinates $i \in I$ have $x_i \geq -1 + \beta$ (since we assumed less than $\frac{c_{\text{tight}} m}{4}$ coordinates of x in total are below this threshold), which gives a contradiction since

$$\|x - x_\star\|_2^2 \geq \sum_{\substack{i \in I \\ x_i \geq -1+\beta}} (x_i - [x_\star]_i)^2 > \frac{c_{\text{tight}} m \beta^2}{4}.$$

$\square$

We next show that by strong convexity, we can certify a bound on $\|\tilde{x} - x_\star\|_2$ by finding any feasible point $\tilde{x} \in \mathcal{K}_\rho \cap [-1, 1]^m$ with near-minimal distance to g.

Lemma 2. *Suppose $\tilde{x} \in \mathcal{K} \cap [-1, 1]^m$ and $\|\tilde{x} - g\|_2^2 \leq r_\star^2 + \frac{c_{\text{tight}} m \beta^2}{4}$. Then $\|\tilde{x} - x_\star\|_2^2 \leq \frac{c_{\text{tight}} m \beta^2}{4}$.*

Proof. Recall that by definition, $x_\star = \arg\min_{x \in \mathcal{K} \cap [-1,1]^m} \|x - g\|_2^2$. Since $\|x - g\|_2^2$ is 2-strongly convex in the ℓ_2 norm and $\tilde{x} \in \mathcal{K} \cap [-1,1]^m$, optimality of $x_\star$ implies the claim by rearranging

$$\|\tilde{x} - x_\star\|_2^2 + r_\star^2 = \|\tilde{x} - x_\star\|_2^2 + \|x_\star - g\|_2^2 \leq \|\tilde{x} - g\|_2^2 \leq r_\star^2 + \frac{c_{\text{tight}} m \beta^2}{4}.$$

$\square$

In light of Lemma 2, we find a point in $[-1,1]^m \cap \mathcal{K}$ with near-minimal distance to g by repeatedly solving optimization problems which balance the objectives $f(x)$ and $\|x - g\|_2^2$, described next.

Testing via regularized optimization. Our algorithm is based on a binary search subroutine, each iteration of which is based on approximately solving an optimization problem f_λ parameterized by a regularization parameter λ, belonging to the following family:

$$\min_{x \in [-1,1]^m} f_\lambda(x) := f(x) + \lambda \|x - g\|_2^2. \tag{5}$$

We will discuss how to efficiently approximate the optimizer of f_λ in (5) using tools from [JT23] and other optimization subroutines in the other sections of this paper. Here, we show how approximate solutions to (5) can be used to perform a binary search on λ and approximate $x_\star$. Throughout, we will denote our desired additive error threshold (as specified in Lemma 2) by

$$\tau := \frac{c_{\text{tight}} m \beta^2}{4}.$$

We will also denote for convenience,

$$x_g := \arg\min_{x \in [-1,1]^m} \|x - g\|_2, \quad x_\lambda := \arg\min_{x \in [-1,1]^m} f_\lambda(x) \text{ for all } \lambda \geq 0. \tag{6}$$

We begin by establishing crude bounds on our λ binary search, assuming the bounds

$$\Theta \geq f(x) \geq 0 \text{ for all } x \in [-1,1]^m, \text{ and } f(\mathbb{0}_m) = 0. \tag{7}$$

We remark that because f is symmetric and convex, it is minimized at $\mathbb{0}_m$, and in all our applications we have $f(\mathbb{0}_m) = 0$; otherwise, one may first perform an additive shift to f to use our framework.

Lemma 3. *Suppose f satisfies (7). For $\lambda \geq \frac{4\Theta}{\tau}$, x_g minimizes f_λ to additive error $\frac{\lambda\tau}{4}$ over $[-1,1]^m$.*

Proof. Defining x_λ, x_g as in (6), and using optimality of x_g and (7),

$$f_\lambda(x_g) \leq \Theta + \lambda \|x_g - g\|_2^2 \leq \frac{\lambda\tau}{4} + f(x_\lambda) + \lambda \|x_\lambda - g\|_2^2.$$

$\square$

Lemma 4. *Suppose f satisfies (7). For $\lambda \leq \frac{\rho}{8m}$, and under Assumption 1, any x minimizing f_λ to additive error $\frac{\rho}{4}$ over $[-1,1]^m$ has $f(x) \leq \frac{3\rho}{4}$.*

Proof. Since $\mathbb{0}_m \in [-1,1]^m$ and $f(\mathbb{0}_m) = 0$, we have by the assumption on $\|g\|_2 \leq 2\sqrt{m}$ that

$$f(x) \leq \lambda \left(\|g\|_2^2 - \|x - g\|_2^2 \right) + \frac{\rho}{4} \leq 4\lambda m + \frac{\rho}{4} \leq \frac{3\rho}{4}.$$

$\square$

10

We also require a helper lemma that rounds a near-feasible point with bounded distance increase.

Lemma 5. *Suppose f satisfies (7). If $x \in [-1, 1]^m$ satisfies $f(x) \leq \rho(1+c)$, then defining $\tilde{x} \leftarrow \frac{1}{1+c}x$, under Assumption 1 we have $\tilde{x} \in \mathcal{K} \cap [-1, 1]^m$ and $\|\tilde{x} - g\|_2^2 \leq \|x - g\|_2^2 + 4cm$.*

Proof. The first conclusion follows by definition of $\tilde{x}$ and convexity of f, where we use $f(\mathbb{0}_m) = 0$. The second uses $\|g\|_2 \|x\|_2 \leq 2m$, and so we may bound

$$\|\tilde{x} - g\|_2^2 - \|x - g\|_2^2 = \frac{2c}{1+c} \langle g, x \rangle - \left(1 - \left(\frac{1}{1+\alpha}\right)^2\right) \|x\|_2^2 \leq 4cm.$$

$\square$

Finally, we show that for nearby values of λ, we can aggregate approximate minimizers of f_λ to obtain a point in $\mathcal{K} \cap [-1, 1]^m$ with nearly-optimal distance to g. In the statement of the following result, we assume only multiplicative error guarantees on querying the value of f. This is because in our applications where $f(x) = \|\mathcal{A}(x)\|_{\mathrm{op}}$, a full eigendecomposition is prohibitively expensive so we will approximate operator norms via the power method (see Proposition 4).

Lemma 6. *Let $\lambda' \in (\lambda, (1 + \frac{\tau}{10m})\lambda)$, and suppose x, x' respectively minimize f_λ and $f_{\lambda'}$ to additive error $\frac{\lambda'\tau}{4}$ over $[-1, 1]^m$. Further assume for $c := \min(\frac{\tau}{12m}, \frac{\tau\lambda}{3\rho})$ that we know scalar values $A \leq \rho$ and $A' \geq (1 + c)\rho$ such that $f(x) \in [(1 - c)A, A]$ and $f(x') \in [(1 - c)A', A']$. Define $\alpha := 1 - \frac{\rho - A}{A' - A}$ and $\tilde{x} := \alpha x + (1 - \alpha)x'$. Then under Assumption 1,*

$$\|\tilde{x} - g\|_2^2 \leq r_\star^2 + \tau \text{ and } \tilde{x} \in [-1, 1]^m \cap \mathcal{K}.$$

Proof. By convexity of the set $[-1, 1]^m$ and the function f, and because $\alpha A + (1 - \alpha)A' = \rho$ by construction, we clearly have $y \in [-1, 1]^m \cap \mathcal{K}$, so it suffices to show the distance bound. Recall from (4) that there exists $x_\star \in [-1, 1]^m \cap \mathcal{K}$ with $\|x_\star - g\|_2^2 = r_\star^2$ by definition. Since

$$(1 - c)A + \lambda \|x - g\|_2^2 \leq f(x) + \lambda \|x - g\|_2^2 \leq f(x_\star) + \lambda r_\star^2 + \frac{\lambda'\tau}{4} \leq \rho + \lambda r_\star^2 + \frac{\lambda'\tau}{4},$$

$$(1 - c)A' + \lambda \|x' - g\|_2^2 \leq f(x') + \lambda' \|x' - g\|_2^2 \leq f(x_\star) + \lambda' r_\star^2 + \frac{\lambda'\tau}{4} \leq \rho + \lambda' r_\star^2 + \frac{\lambda'\tau}{4},$$

we have by a convex combination of the above display that, for $\bar{\lambda} := \alpha\lambda + (1 - \alpha)\lambda'$,

$$(1 - c)\rho + \alpha\lambda \|x - g\|_2^2 + (1 - \alpha)\lambda' \|x - g\|_2^2 \leq \rho + \bar{\lambda}r_\star^2 + \frac{\lambda'\tau}{4}.$$

Rearranging and dividing by $\bar{\lambda} \in (\lambda, \lambda')$, we have

$$\|\tilde{x} - g\|_2^2 \leq r_\star^2 + \frac{c\rho}{\bar{\lambda}} + \frac{\lambda'\tau}{4\bar{\lambda}} + \frac{1}{\bar{\lambda}} \left(\bar{\lambda} \|\tilde{x} - g\|_2^2 - \alpha\lambda \|x - g\|_2^2 - (1 - \alpha)\lambda' \|x' - g\|_2^2\right)$$

$$\leq r_\star^2 + \frac{\tau}{3} + \frac{\tau}{3} + \frac{1}{\bar{\lambda}} \left((\bar{\lambda} - \lambda)\alpha \|x - g\|_2^2\right) \leq r_\star^2 + \frac{2\tau}{3} + \frac{10m(\lambda' - \lambda)}{\lambda} \leq r_\star^2 + \tau.$$

The second inequality used $\|\tilde{x} - g\|_2^2 \leq \alpha \|x - g\|_2^2 + (1 - \alpha) \|x' - g\|_2^2$ by convexity as well as our bounds on c and $\frac{\lambda'}{\lambda}$, and the third used $\|x - g\|_2^2 \leq 2\|x\|_2^2 + 2\|g\|_2^2 \leq 10m$ by Assumption 1. $\square$

We combine these developments to give our final procedure for approximate partial coloring.

Proposition 2. *Let $\beta \in (0,1)$, $\rho > 0$, and let $\tau = \Theta(m\beta^2)$, $T = O(\log \frac{1}{\beta} + \log\log \frac{\Theta}{\rho})$ for appropriate constants. Let symmetric and convex $f : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ satisfy (7), and define $\mathcal{K}$ as in (3). Under Assumption 1 and following notation (4), there is an algorithm returning $\tilde{x} \in \mathcal{K} \cap [-1,1]^m$ with*

$$\|\tilde{x} - g\|_2^2 \leq r_\star^2 + \tau.$$

The runtime is the cost of solving f_λ to additive error $\frac{\lambda\tau}{4}$ for T values of $\lambda \geq \frac{\rho}{8m}$, and computing $f(x)$ to within $\frac{\tau}{64m}$ multiplicative error for T values of $x \in [-1,1]^m$.

Proof. Let $c := \frac{\tau}{64m} = \Theta(\beta^2)$, and divide the range $[\frac{\rho}{8m}, \frac{4\Theta}{\tau}]$ into multiplicative intervals of the form $[\lambda, (1+c)\lambda]$; there are clearly $O(\frac{1}{c}\log\frac{m\Theta}{\tau\rho}) = O(\frac{1}{\beta^2}\log\frac{\Theta}{\beta\rho})$ such intervals. We will initialize $\lambda \leftarrow \frac{\rho}{8m}$ and $\lambda' \leftarrow \frac{4\Theta}{\tau}$, and throughout our binary search we maintain the following invariants.

1. λ, λ' are lower and upper endpoints of different multiplicative intervals.

2. x and x' respectively minimize f_λ and $f_{\lambda'}$ to additive error $\frac{\lambda\tau}{4}$ and $\frac{\lambda'\tau}{4}$ over $[-1,1]^m$.

3. A and A' respectively satisfy $f(x) \in [(1-c)A, A]$ and $f(x') \in [(1-c)A', A']$.

4. $A \leq \rho$ and $A' \geq (1+c)\rho$.

At initialization, since $\frac{\tau\rho}{32m} \leq \frac{\rho}{4}$, we can set x to any $\frac{\tau\rho}{32m}$-approximate minimizer of f_λ and A to be any value satisfying $f(x) \in [(1-c)A, A]$, since A will necessarily be less than ρ by Lemma 4. Further, Lemma 3 shows that it suffices to initialize $x' \leftarrow x_g$, and let A' be any value satisfying $\|\mathcal{A}(x')\|_{\mathrm{op}} \in [(1-c)A', A']$. If $A' \leq (1+c)\rho$, we can apply Lemma 5 to x_g yielding $\tilde{x}$ with

$$\|\tilde{x} - g\|_2^2 \leq \|x_g - g\|_2^2 + 4cm \leq r_\star^2 + \frac{\tau}{16}, \ \tilde{x} \in \mathcal{K}_\rho \cap [-1,1]^m. \tag{8}$$

Hence it suffices to return $\tilde{x}$. Otherwise, $A' \geq (1+c)\rho$ so all invariants are met at initialization. Next, let λ_{test} be the lower endpoint of the middle interval between λ and λ', let x_{test} be a $\frac{\lambda_{\mathrm{test}}\tau}{4}$-approximate minimizer to $f_{\lambda_{\mathrm{test}}}$, and let $f(x_{\mathrm{test}}) \in [(1-c)A_{\mathrm{test}}, A_{\mathrm{test}}]$. If $A_{\mathrm{test}} \in [\rho, (1+c)\rho]$,

$$f(x_{\mathrm{test}}) + \lambda_{\mathrm{test}}\|x_{\mathrm{test}} - g\|_2^2 \leq f(x_\star) + \lambda_{\mathrm{test}}r_\star^2 + \frac{\lambda_{\mathrm{test}}\tau}{4}$$
$$\leq A_{\mathrm{test}} + \lambda_{\mathrm{test}}r_\star^2 + \frac{\lambda_{\mathrm{test}}\tau}{4}$$
$$\leq \frac{1}{1-c}f(x_{\mathrm{test}}) + \lambda_{\mathrm{test}}r_\star^2 + \frac{\lambda_{\mathrm{test}}\tau}{4},$$

so rearranging and dividing by $\lambda_{\mathrm{test}} \geq \frac{\rho}{8m}$ proves that

$$\|x_{\mathrm{test}} - g\|_2^2 \leq \frac{c}{1-c} \cdot \frac{(1+c)\rho}{\lambda_{\mathrm{test}}} + r_\star^2 + \frac{\tau}{4} \leq 2c\rho \cdot \frac{8m}{\rho} + r_\star^2 + \frac{\tau}{4} \leq r_\star^2 + \frac{\tau}{2},$$

where we used our choice of c. Therefore, letting $\tilde{x}$ be the result of applying Lemma 5 to x_{test} with parameter c, the same calculation as (8) shows

$$\|\tilde{x} - g\|_2^2 \leq r_\star^2 + \tau, \ \tilde{x} \in \mathcal{K} \cap [-1,1]^m.$$

Otherwise, we either have $A_{\mathrm{test}} \leq \rho$ or $A_{\mathrm{test}} \geq (1+c)\rho$, so we may update λ or λ' to λ_{test} and continue. At termination we must have λ and λ' within a multiplicative factor of $1+c$, and therefore applying Lemma 6 yields the desired point $\tilde{x}$, where we check that indeed $c \leq \min(\frac{\tau}{12m}, \frac{\tau}{3\rho} \cdot \frac{\rho}{8m})$.

Finally, we test $T = O(\log(\frac{1}{\beta^2}\log\frac{\Theta}{\beta\rho}))$ values of λ, and the runtime is from running approximate minimization routines to maintain x, x' and approximate function calls to maintain A, A'. $\quad\square$

12

In a typical application of the framework of this section, we will first specify a discrepancy radius ρ such that the body $\mathcal{K}$ from (3) has an exponential Gaussian measure lower bound. We then can combine Proposition 1, Lemma 1, Lemma 2, and Proposition 2 to efficiently obtain an approximate partial coloring, by approximating $x_\star$ for both g and $-g$ (in the notation of (4)). Both of our approximations via Proposition 2 will lie in $[-1,1]^m \cap \mathcal{K}$, and then the other three results imply that one of the approximations has a constant fraction of coordinates in $[-1, -1 + \beta]$.

4 Gaussian measure lower bounds

In this section, we give two new tools for obtaining Gaussian measure lower bounds, a key ingredient of the [Rot17] framework. In Section 4.1, we first state a general reduction from bounding Gaussian measure for a set $\mathcal{K}$ to proving a closely related notion introduced by [RR20], which roughly says that a random Gaussian vector is not too far from a multiple of $\mathcal{K}$ with constant probability. This can be seen as a generalization of Conjecture 1 in [RR20]. We complement this reduction with a strengthening of the main result of [RR20], which is parameterized by a trace bound, and is needed to prove our ultrasparsifier results in Section 6. Finally, in Section 4.2 we prove as corollaries of these tools the main Gaussian lower bounds we will use in the paper.

4.1 Reduction to Gaussian distance bound

In this section, we prove the following reduction holds.

Theorem 3. *Suppose $\mathcal{K} \subset \mathbb{R}^m$ is symmetric and convex, and that for a constant C_0,*

$$\gamma_m \left(\frac{C_0}{\alpha} \mathcal{K} + \alpha \sqrt{m} \mathbb{B}_2^m \right) \geq \frac{1}{2} \text{ for all } \alpha \in (0,1).$$

Then there is a constant C such that $\gamma_m(\mathcal{K}) \geq \exp(-Cm)$.

The main technical tool we use is the existence of 1-*regular M-ellipsoids*. We state a variant of [Pis89], which gives a result for p-regular M-ellipsoids for all $p \in (0,2)$; we only use $p = 1$.

Proposition 3 (Corollary 7.16, [Pis89])**.** *There is a constant C_1 such that for any symmetric convex $\mathcal{K} \subseteq \mathbb{R}^m$, there is an ellipsoid $\mathcal{E}$ with*

$$\max\{N(\mathcal{K}, t\mathcal{E}), N(\mathcal{E}, t\mathcal{K})\} \leq \exp(C_1 m t^{-1}).$$

Before giving our main technical result Lemma 9, we need a few facts about covering numbers. For the following standard facts, see Theorem 4.1.13 and Facts 4.1.7, 4.1.8 and 4.1.9 in [AAGM15].

Fact 1. *For any convex sets $A, B, C \subseteq \mathbb{R}^m$, we have $N(A, B) \leq N(A, C)N(C, B)$.*

Fact 2. *For any convex sets $A, B, C \subseteq \mathbb{R}^m$, we have $N(A + C, B + C) \leq N(A, B)$.*

Fact 3. *For any convex sets $A, B \subseteq \mathbb{R}^m$, we have $N(A, B) \geq \frac{\text{Vol}(A)}{\text{Vol}(B)}$. Similarly, $N(A, B) \geq \frac{\gamma_m(A)}{\gamma_m(B)}$.*

Fact 4. *For any convex sets $A, B \subseteq \mathbb{R}^m$ with B symmetric, $N(A, B) \leq \frac{\text{Vol}(2A+B)}{\text{Vol}(B)}$.*

Fact 5. *For any convex sets $A, B \subseteq \mathbb{R}^m$ with A symmetric, $N(A, 2(A \cap B)) \leq N(A, B)$.*

For the following fact, see Lemma 3.3 in [DJR22].

Fact 6. *For any symmetric convex set $\mathcal{K} \subseteq \mathbb{R}^m$ with $\gamma_m(\mathcal{K}) \geq \frac{1}{2}$, we have $N(\sqrt{m}\mathbb{B}_2^m, \mathcal{K}) \leq \exp(C_2 m)$ for a universal constant $C_2 > 0$.*

As a corollary of Fact 4 we also have the following.

Lemma 7. *For symmetric convex sets $A, B \subseteq \mathbb{R}^m$, $N(A, B) \leq 3^m \cdot \frac{\text{Vol}(A)}{\text{Vol}(A \cap B)}$.*

Proof. Since $A \cap B$ is also symmetric, by Fact 4 we have

$$N(A, B) \leq N(A, A \cap B) \leq \frac{\text{Vol}(2A + (A \cap B)))}{\text{Vol}(A \cap B)} \leq \frac{\text{Vol}(3A)}{\text{Vol}(A \cap B)} = 3^m \cdot \frac{\text{Vol}(A)}{\text{Vol}(A \cap B)}. \qquad \square$$

We will also need a more specific result about covering numbers of slices by subspaces.

Lemma 8. *For any symmetric convex $A, B \subseteq \mathbb{R}^m$ and any d-dimensional subspace $U \subseteq \mathbb{R}^m$, $N(A \cap U, B \cap U) \leq 6^d \cdot N(A, B)$.*

Proof. By Lemma 7, we have

$$N(A \cap U, B \cap U) \leq N(A \cap U, A \cap B \cap U) \leq 3^d \cdot \frac{\text{Vol}(A \cap U)}{\text{Vol}(A \cap B \cap U)}.$$

It remains to note that by Fact 5, we can cover A with $N(A, B)$ copies of $2(A \cap B)$, so that Fact 3 yields $\text{Vol}(A \cap U) \leq N(A, B) \cdot \text{Vol}(2(A \cap B) \cap U) = 2^d \cdot N(A, B) \cdot \text{Vol}(A \cap B \cap U)$. $\qquad \square$

We are now ready to state and prove our main technical lemma.

Lemma 9. *Let $\mathcal{K} \subseteq \mathbb{R}^m$ be a symmetric convex set such that $\gamma_m(\frac{C_0}{\alpha}\mathcal{K} + \alpha\sqrt{m}\mathbb{B}_2^m) \geq \frac{1}{2}$ for all $\alpha > 0$ and let $\mathcal{E}$ be a 1-regular M-ellipsoid for $\mathcal{K}$. Then there is a universal constant C' such that for every $r > 0$, the number of axes of $\mathcal{E}$ of length at most r is at most $C'\sqrt{r} \cdot m^{\frac{3}{4}}$.*

Proof. Let U denote the span of directions corresponding to the axes of $\mathcal{E}$ of length at most r with $d := \dim(U)$. To simplify notation, let $\mathcal{K}(\alpha) := \frac{C_0}{\alpha}\mathcal{K} + \alpha\sqrt{m}\mathbb{B}_2^m$, so that we still have $\gamma_U(\mathcal{K}(\alpha) \cap U) \geq \gamma_m(\mathcal{K}(\alpha)) \geq \frac{1}{2}$, where γ_U is the d-dimensional Gaussian measure in U. In particular, letting $\mathbb{B}_2^U$ denote the unit Euclidean ball restricted to U, $N(\sqrt{d}\mathbb{B}_2^U, \mathcal{K}(\alpha) \cap U) \leq \exp(C_2 d)$ by Fact 6.

Further, note that for any $\alpha > 0$ we have

$$N\left(\mathcal{K}(\alpha) \cap U, \left(\frac{C_0}{\alpha} \cdot \frac{mr}{d} + \alpha\sqrt{m}\right)\mathbb{B}_2^U\right) \leq N\left(\mathcal{K}(\alpha) \cap U, \left(\frac{C_0}{\alpha} \cdot \frac{m}{d}\mathcal{E} + \alpha\sqrt{m}\mathbb{B}_2^m\right) \cap U\right)$$

$$\leq 6^d \cdot N\left(\mathcal{K}(\alpha), \frac{C_0}{\alpha} \cdot \frac{m}{d}\mathcal{E} + \alpha\sqrt{m}\mathbb{B}_2^m\right)$$

$$\leq 6^d \cdot N\left(\mathcal{K}, \frac{m}{d}\mathcal{E}\right) \leq 6^d \cdot \exp(C_1 d),$$

where in the second inequality we use Lemma 8, in the third we use Fact 2 and in the fourth we use Proposition 3. Setting $\alpha := \sqrt{C_0 \frac{mr}{d}} \cdot m^{-\frac{1}{4}}$, we have $\frac{C_0}{\alpha} \cdot \frac{mr}{d} + \alpha\sqrt{m} = \sqrt{C_0 \frac{r}{d}} \cdot m^{\frac{3}{4}}$, so by Fact 1,

$$N\left(\sqrt{d}\mathbb{B}_2^U, 2\sqrt{\frac{C_0 r}{d}}m^{\frac{3}{4}}\mathbb{B}_2^U\right) \leq N\left(\sqrt{d}\mathbb{B}_2^U, \mathcal{K}(\alpha) \cap U\right) \cdot N\left(\mathcal{K}(\alpha) \cap U, 2\sqrt{\frac{C_0 r}{d}}m^{\frac{3}{4}}\mathbb{B}_2^U\right)$$

$$\leq \exp(C_2 d) \cdot 6^d \cdot \exp(C_1 d).$$

On the other hand, by Fact 3,

$$N\left(\sqrt{d}\mathbb{B}_2^U, 2\sqrt{\frac{C_0 r}{d}}m^{\frac{3}{4}}\mathbb{B}_2^U\right) \geq \frac{\text{Vol}_d(\sqrt{d}\mathbb{B}_2^U)}{\text{Vol}_d(2\sqrt{\frac{C_0 r}{d}}m^{\frac{3}{4}}\mathbb{B}_2^U)} = \left(\frac{d}{2\sqrt{C_0 r}m^{\frac{3}{4}}}\right)^d.$$

Combining the above two displays yields the claim. $\qquad \square$

Proof of Theorem 3. First we show a Gaussian measure lower bound for a 1-regular M-ellipsoid $\mathcal{E}$ of $\mathcal{K}$ with axes of lengths $\{\lambda_i\}_{i\in[m]}$ sorted in increasing order. Let k denote the maximum index with $\lambda_k \leq \sqrt{m}$ and let $\mathcal{E}'$ denote the ellipsoid with the same eigenvectors as $\mathcal{E}$ and compressed axes of length $\min\{\lambda_i, \sqrt{m}\}$, so that in particular $\mathcal{E}' \subseteq \mathcal{E}$. Note that

$$\gamma_m(\mathcal{E}) \geq \gamma_m(\mathcal{E}') \geq \int_{\mathcal{E}'} \frac{1}{(2\pi)^{m/2}} \exp(-\tfrac{1}{2}\underbrace{\|x\|_2^2}_{\leq m})\mathrm{d}x \geq \exp(-C'm)\cdot \mathrm{Vol}(\mathcal{E}') \geq \exp(-C''m)\prod_{i\in[k]}\frac{\lambda_i}{\sqrt{m}},$$

for some constants $C', C'' > 0$. Denote $I_r := \{i \in [k] : \lambda_i \in [\frac{r}{2}, r]\}$. We apply Lemma 9 as follows:

$$\prod_{i\in[k]}\frac{\lambda_i}{\sqrt{m}} \geq \prod_{r\in\sqrt{m}\cdot 2^{\mathbb{Z}\leq 0}}\left(\frac{r}{2\sqrt{m}}\right)^{|I_r|} \geq \prod_{r\in\sqrt{m}\cdot 2^{\mathbb{Z}\leq 0}}\left(\frac{r}{2\sqrt{m}}\right)^{C'\sqrt{r}m^{\frac{3}{4}}}.$$

Denoting $\frac{r}{2\sqrt{m}} = 2^{-a}$, the product becomes, for a constant C_3,

$$\prod_{i\in[k]}\frac{\lambda_i}{\sqrt{m}} \geq \left(\prod_{a=0}^{\infty}(2^{-a})^{2^{-\frac{a}{2}}}\right)^{C'm} = \exp(-C_3 m).$$

Finally, we may use Fact 3 and Proposition 3 to lower bound the Gaussian measure of $\mathcal{K}$:

$$\gamma_m(\mathcal{K}) \geq \frac{\gamma_m(\mathcal{E})}{N(\mathcal{E}, \mathcal{K})} \geq \exp(-(C'' + C_1 + C_3)m). \qquad \square$$

4.2 Operator norm discrepancy bodies

In this section, we combine Theorem 3 with the following generalization of the main result in [RR20] to obtain new Gaussian measure lower bounds. The argument we use to prove Theorem 4 is almost identical to [RR20], where we simulate a random walk while blocking "dangerous" directions to control a potential function, which results in a measure lower bound. The main difference is in the choice of the potential function we use to capture our trace condition. Due to the significant overlap with [RR20], we defer a proof of the following result to Appendix A for completeness.

Theorem 4. *Suppose* $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ *satisfy* $\mathcal{A}(\mathbb{1}_m) \preceq \mathbf{I}_n$ *and* $\mathrm{Tr}(\mathcal{A}(\mathbb{1}_m)) \leq \tau$ *for sufficiently large* m *and* $n \leq 2^{\frac{m}{5}}$. *Following notation* (2), *there is a constant* C_0 *such that*

$$\gamma_m\left(\frac{C_0}{\alpha}\mathcal{K}_{\sqrt{\frac{\tau}{m}},\mathcal{A}} + \alpha\sqrt{m}\mathbb{B}_2^m\right) \geq \frac{1}{2}, \text{ for all } \alpha \in (0,1).$$

As a first application of our techniques, we prove the following new Gaussian measure lower bound on the operator norm ball, which shows Conjecture 1 of [RR20] is true.

Corollary 2. *Suppose* $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ *satisfy* $\mathcal{A}(\mathbb{1}_m) \preceq \mathbf{I}_n$, *and* $\frac{m}{n}$ *is at least a sufficiently large constant. Following notation* (2), *there is a constant* $C > 0$ *such that*

$$\gamma_m\left(\mathcal{K}_{\sqrt{\frac{n}{m}},\mathcal{A}}\right) \geq \exp\left(-Cm\right).$$

Proof. It suffices to directly combine Theorem 3 and Theorem 1 from [RR20]. $\qquad \square$

We also prove the following variant of Corollary 2, which will be used to design ultrasparsifiers.

Corollary 3. *Suppose $\{\mathbf{A}_i\}_{i \in [m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ satisfy $\mathcal{A}(\mathbb{1}_m) \preceq \mathbf{I}_n$ and $\mathrm{Tr}(\mathcal{A}(\mathbb{1}_m)) \leq \tau$ for sufficiently large m and $n \leq 2^{\frac{m}{5}}$. Following notation (2), there is a constant $C > 0$ such that*

$$\gamma_m \left(\mathcal{K}_{\sqrt{\frac{\tau}{m}}, \mathcal{A}} \right) \geq \exp\left(-Cm \right).$$

Proof. It suffices to directly combine Theorem 3 and Theorem 4. $\qquad\square$

5 Linear-sized sparsifiers

In this section we will be concerned with the following problem: we are given a set of matrices $\{\mathbf{M}_i\}_{i \in [m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ satisfying $\mathcal{M}(\mathbb{1}_m) = \mathbf{I}_n$ where $\mathcal{M}$ is the operator associated with the set in the sense of (1), and a parameter $\varepsilon \in (0, 1)$ (both fixed throughout). Our goal is to construct a vector $w \in \mathbb{R}_{\geq 0}^m$ such that $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$, and $\|\mathcal{M}(w) - \mathbf{I}_n\|_{\mathrm{op}} \leq \varepsilon$. We will achieve this goal by an instantiation of the framework in Section 3, combined with recursive rounding. We state the main regularized optimization and approximate function query subroutines we use (as required by Proposition 2) in Section 5.1, and put the pieces together to prove our main result in Section 5.2. Finally, we discuss implications and runtime considerations for sparsifying graphs in Section 5.3.

5.1 Approximate Gaussian rounding in nearly-linear time

In the applications of the framework in Section 3 to the sparsification setting, we will define a discrepancy body with respect to $f(x) := \|\mathcal{A}(x)\|_{\mathrm{op}}$, where $\mathcal{A}$ is the operator associated to a family of matrices $\{\mathbf{A}_i\}_{i \in [m]}$. The matrix family will always be a reweighting of a subset of our original matrices $\{\mathbf{M}_i\}_{i \in [m]}$. To leverage Proposition 2 to approximate Gaussian roundings, we require two helper tools. We first recall a guarantee on approximate top singular value computation.

Proposition 4 (Theorem 1, [MM15]). *For any $c \in (0, 1)$ and $x \in \mathbb{R}^m$, we can return a scalar value A such that $\|\mathcal{A}(x)\|_{\mathrm{op}} \in [(1 - c)A, A]$ with probability $\geq 1 - \delta$ in time*

$$O\left(\left(\sum_{i \in [m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i) \right) \cdot \frac{\log \frac{m}{\delta}}{\sqrt{c}} \right).$$

Proposition 2 also requires approximate solvers for regularized optimization problems of the form f_λ defined in (5). When $f(x) = \|\mathcal{A}(x)\|_{\mathrm{op}}$, we have the characterization

$$f_\lambda(x) = \|\mathcal{A}(x)\|_{\mathrm{op}} + \lambda \|x - g\|_2^2 = \max_{\mathbf{Y} \in \Delta^{2n \times 2n}} \left\langle \mathbf{Y}, \sum_{i \in [m]} x_i \widetilde{\mathbf{A}}_i \right\rangle + \lambda \|x - g\|_2^2, \tag{9}$$

$$\text{where } \widetilde{\mathbf{A}}_i := \begin{pmatrix} \mathbf{A}_i & \mathbf{0}_n \\ \mathbf{0}_n & -\mathbf{A}_i \end{pmatrix}, \text{ and } \Delta^{2n \times 2n} := \{\mathbf{Y} \in \mathbb{S}_{\succeq \mathbf{0}}^{2n} \mid \mathrm{Tr}\mathbf{Y} = 1\}.$$

The recent work [JT23] developed a generic framework for solving box-spectraplex games, which (9) is almost a case of; the main difference is that (9) is regularized. We prove the following modification of Theorem 3 of [JT23] in Appendix B, which handles the extra part of the objective.

Proposition 5. *Let $\delta, \Delta \in (0, 1)$, $\lambda \geq 0$, and $g \in \mathbb{R}^m$. There is an algorithm which returns $\tilde{x} \in [-1, 1]^m$ minimizing f_λ (defined in (9)) to additive error Δ, with probability $\geq 1 - \delta$, in time*

$$O\left(\left(\sum_{i \in [m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i) \right) \cdot \frac{\|\mathcal{A}(\mathbb{1}_m)\|_{\mathrm{op}}^{3.5} \log^2 \left(\frac{mn}{\delta \Delta} \right) \log(n)}{\Delta^{3.5}} \right).$$

16

5.2 Linear-sized sparsifiers in nearly-linear time

Finally, we put the pieces together to prove our main result. To warm start our algorithm, we use the following standard random sampler which loses a logarithmic factor from the desired sparsity.

Lemma 10. *Let $\{\mathbf{M}_i\}_{i \in [m]} \subset \mathbb{S}^n_{\succeq \mathbf{0}}$ satisfy $\mathcal{M}(\mathbb{1}_m) \preceq \mathbf{I}_n$ and $\mathrm{Tr}(\mathcal{M}(\mathbb{1}_m)) \leq \tau$, and let $K = \Theta(\frac{\tau}{\varepsilon^2} \log \frac{n}{\delta})$ for an appropriate constant. Define independently distributed random matrices $\{\mathbf{X}_k\}_{k \in [K]}$ such that*

$$\mathbf{X}_k = \frac{1}{Kp_i}\mathbf{M}_i \text{ with probability } p_i := \frac{\mathrm{Tr}(\mathbf{M}_i)}{\sum_{i \in [m]} \mathrm{Tr}(\mathbf{M}_i)} \text{ for all } k \in [K].$$

Then with probability $\geq 1 - \delta$, $\|\sum_{k \in [K]} \mathbf{X}_k - \mathcal{M}(\mathbb{1}_m)\|_{\mathrm{op}} \leq \varepsilon$.

Proof. First, it is clear that $\{p_i\}_{i \in [m]}$ is a valid sampling distribution. Next we define random matrices $\mathbf{Y}_k := \frac{1}{Kp_i}\mathbf{M}_i - \frac{1}{K}\mathcal{M}(\mathbb{1}_m)$, which are mean-zero and satisfy $\|\mathbf{Y}_k\|_{\mathrm{op}} \leq \frac{\tau}{K}$. Further, recall

$$(\mathbf{A} - \mathbf{B})^2 \preceq 2\mathbf{A}^2 + 2\mathbf{B}^2, \quad \frac{1}{\mathrm{Tr}\mathbf{A}}\mathbf{A}^2 \preceq \mathbf{A},$$

for all $\mathbf{A}, \mathbf{B} \in \mathbb{S}^n_{\succeq \mathbf{0}}$. The first inequality follows from $(\mathbf{A} + \mathbf{B})^2 \in \mathbb{S}^n_{\succeq \mathbf{0}}$ and the second follows because the two sides commute and we compare eigenvalues. Using these claims and $\mathcal{M}(\mathbb{1}_m) \preceq \mathbf{I}_n$, we have

$$\sum_{k \in [K]} \mathbb{E}\mathbf{Y}_k^2 = \frac{1}{K}\sum_{i \in [m]} p_i \left(\frac{1}{p_i}\mathbf{M}_i - \mathcal{M}(\mathbb{1}_m)\right)^2$$

$$\preceq \frac{2}{K}\sum_{i \in [m]} \frac{1}{p_i}\mathbf{M}_i^2 + \frac{2}{K}\mathcal{M}(\mathbb{1}_m)^2$$

$$\preceq \frac{2\tau}{K}\sum_{i \in [m]} \frac{1}{\mathrm{Tr}(\mathbf{M}_i)}\mathbf{M}_i^2 + \frac{2}{K}\mathbf{I}_n \preceq \frac{4\tau}{K}\mathbf{I}_n.$$

The desired conclusion then follows by the matrix Bernstein bound (Theorem 1.6.2, [Tro15]). $\qquad\square$

Combining Propositions 2, 4, and 5 with Lemma 10 gives our main subroutine, which reweights a sum of matrices into a "sparse" component and a "small" component which we can recurse on.

Proposition 6. *Let $\varepsilon \in (0, 1)$, $\delta \in (\exp(-\Omega(n)), 1)$ for an appropriate constant, and let $\{\mathbf{M}_i\}_{i \in [m]} \subset \mathbb{S}^n_{\succeq \mathbf{0}}$ satisfy $\mathcal{M}(\mathbb{1}_m) \preceq \mathbf{I}_n$. There is an algorithm returning $u = v + w$ for $v, w \in \mathbb{R}^m_{\geq 0}$ satisfying*

$$\|\mathcal{M}(u - \mathbb{1}_m)\|_{\mathrm{op}} \leq \varepsilon, \ \mathrm{nnz}(w) \leq \frac{C_{\mathrm{sparse}}n}{\varepsilon^2}, \ \|\mathcal{M}(v)\|_{\mathrm{op}} \leq \frac{1}{10}, \tag{10}$$

with probability $\geq 1 - \delta$, for a universal constant C_{sparse}, in time

$$O\left(\left(\sum_{i \in [m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{M}_i)\right) \cdot \frac{\log(m)\log^2(\frac{m}{\delta})\log\log^{11+o(1)}(\frac{m}{\delta})}{\varepsilon^{3.5}}\right).$$

Proof. Without loss of generality, assume ε is sufficiently small, and let $v_0 := \mathbb{0}_m$. Lemma 10 with $\tau \leftarrow n$ produces an initial reweighting w_0 of the matrices such that $\mathrm{nnz}(w_0) = O(\frac{n}{\varepsilon^2} \log \frac{n}{\delta})$, and $\|\mathcal{M}(w_0 - \mathbb{1}_m)\|_{\mathrm{op}} \leq \varepsilon$. Our algorithm proceeds in $K = O(\log\log\frac{n}{\delta})$ phases indexed by $k \in [K]$. We define $\beta := \frac{1}{20K}$, and for all $k \in [K]$, we let $m_{k-1} := \mathrm{nnz}(w_{k-1})$, and set the discrepancy radius

$$\rho_k := \max\left(2C_{\mathrm{set}}\sqrt{\frac{n}{m_{k-1}}}, \frac{\varepsilon}{(K - k + 1)\log^2(K - k + 1)}\right).$$

We terminate if some phase has $m_k \le \frac{C_{\text{sparse}} n}{\varepsilon^2}$. Otherwise, each phase k computes x_k such that for at least $\frac{c_{\text{tight}}}{4} m_{k-1}$ coordinates i, $[x_k]_i \le -1 + \beta$ and

$$\|\mathcal{M}(w_{k-1} \circ x_k)\|_{\text{op}} = O(\rho_k). \tag{11}$$

We then let S_k be the set of coordinates where $[x_k]_i \le -1 + \beta$, and update

$$[w_k]_i \leftarrow 0, \ [v_k]_i \leftarrow [v_{k-1}]_i + [w_{k-1}]_i (1 + [x_k]_i) \text{ for all } i \in S_k,$$
$$[w_k]_i \leftarrow (1 + [x_k]_i)[w_{k-1}]_i \text{ for all } i \notin S_k \text{ with } [w_{k-1}]_i \ne 0.$$

Evidently the update rules imply that in every iteration, letting $u_k := v_k + w_k$,

$$\|\mathcal{M}(u_k - u_{k-1})\|_{\text{op}} = \|\mathcal{M}(w_{k-1} \circ x_k)\|_{\text{op}} = O(\rho_k),$$
$$\text{nnz}(w_k) \le \left(1 - \frac{c_{\text{tight}}}{4}\right) \text{nnz}(w_{k-1}).$$

Inducting on the second line above, m_k is geometrically decreasing and at least $\Omega(\frac{n}{\varepsilon^2})$ by the termination condition, so $\sum_{k \in [K]} \rho_k = O(\varepsilon)$. Hence, telescoping the first line above using the triangle inequality shows that for each $k \in [K]$, we have $\|\mathcal{M}(u_k)\|_{\text{op}} \le \|\mathcal{M}(u_0)\|_{\text{op}} + O(\varepsilon) \le 2$ for sufficiently small ε. This in turn implies by the update rule on v_k that

$$\|\mathcal{M}(v_K)\|_{\text{op}} \le \sum_{k \in [K]} \beta \|\mathcal{M}(w_{k-1})\|_{\text{op}} \le \sum_{k \in [K]} \beta \|\mathcal{M}(u_{k-1})\|_{\text{op}} \le \frac{1}{10}.$$

Hence, at termination we have all three guarantees in (10), after adjusting ε by a constant.

To produce x_k in each phase satisfying (11), we first draw $g \sim \mathcal{N}(\mathbb{0}_{m_{k-1}}, \mathbf{I}_{m_{k-1}})$, and define the matrices $\mathbf{A}_i \leftarrow [w_{k-1}]_i \mathbf{M}_i$ for all i where $[w_{k-1}]_i \ne 0$; note that under this definition, $\mathcal{A}(\mathbb{1}_{m_{k-1}}) \preceq 2\mathbf{I}_n$ as proven earlier. The precondition of Corollary 2 is satisfied, so the conclusion of Proposition 1 holds except with exponentially small probability. We then apply Proposition 2 (with Propositions 4 and 5 as subroutines) to solve both of the following problems (following notation (2))

$$\min_{x \in \mathcal{K}_{\rho_k, \mathcal{A}} \cap [-1,1]^{m_{k-1}}} \|x - g\|_2^2, \quad \min_{x \in \mathcal{K}_{\rho_k, \mathcal{A}} \cap [-1,1]^{m_{k-1}}} \|x + g\|_2^2,$$

to additive error $\tau = \frac{c_{\text{tight}} m_{k-1} \beta^2}{4}$. In the context of Proposition 2, $\mathcal{K} = \mathcal{K}_{\rho_k, \mathcal{A}}$ is defined with respect to the discrepancy radius ρ_k, $f(x) = \|\mathcal{A}(x)\|_{\text{op}}$, and $\mathcal{A}$ is the operator associated with the $\{\mathbf{A}\}_{i \in [m_{k-1}]}$. Note that $f(x)$ satisfies (7) with $\Theta = 2$ by the assumption $\mathcal{A}(\mathbb{1}_{m_{k-1}}) \preceq 2\mathbf{I}_n$. Hence, both Assumption 1 and the conclusion of Proposition 1 both hold over all phases with probability at least $1 - \frac{\delta}{2}$ by a union bound, so Lemmas 1 and 2 show at least one of the approximate solutions (to g or $-g$) has at least $\frac{c_{\text{tight}} m_{k-1}}{4}$ coordinates in $[-1, -1 + \beta]$. The runtime follows from applying Propositions 4, 5 with failure probability $\frac{\delta}{4K}$, since each accuracy parameter in Proposition 2 satisifes

$$\frac{\lambda \tau}{4} = \Omega(\rho_k \beta^2) = \Omega\left(\frac{\varepsilon}{\log \log^{3+o(1)} \frac{m}{\delta}}\right),$$

by the second term in the definition of ρ_k and our upper bound on K. We lose two additional $\log \log \frac{m}{\delta}$ factors: one from running K phases, and one from the T overhead in Proposition 2. $\quad \square$

By recursively applying Proposition 6, we finally obtain the main result of this section.

Theorem 5. *Let $\varepsilon \in (0,1)$, $\delta \in (\exp(-\Omega(n)), 1)$ for an appropriate constant, and let $\{\mathbf{M}_i\}_{i \in [m]} \subset \mathbb{S}^n_{\succeq \mathbf{0}}$ satisfy $\mathcal{M}(\mathbb{1}_m) = \mathbf{I}_n$. There is an algorithm which returns $w \in \mathbb{R}^m_{\geq 0}$ satisfying $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$ and $\|\mathcal{M}(w) - \mathbf{I}_n\|_{\mathrm{op}} \leq \varepsilon$, with probability $\geq 1 - \delta$, in time*

$$O\left(\left(\sum_{i \in [m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{M}_i)\right) \cdot \frac{\log(m) \log^2(\frac{m}{\delta}) \log\log^{11+o(1)}(\frac{m}{\delta})}{\varepsilon^{3.5}}\right).$$

Proof. We proceed in $K = O(\log \frac{1}{\varepsilon})$ phases indexed by $k \in [K]$, and in each iteration we set

$$\varepsilon_k = \varepsilon \cdot (1 + c)^k,$$

where K satisfies $10^K \geq \frac{1}{\varepsilon}$ and c is a small enough constant such that $\varepsilon_K < 1$. We initialize $\bar{v}_0 \leftarrow \mathbb{1}_m$ and $\bar{w}_0 \leftarrow \mathbb{0}_m$. In each phase $k \in [K]$, we then run the algorithm in Proposition 6 on $\{[\bar{v}_{k-1}]_i \mathbf{M}_i\}_{i \in S_{k-1}}$ where S_{k-1} is the set of i with $[\bar{v}_{k-1}]_i \neq 0$, with parameter ε_k. Proposition 6 then produces $u_k = w_k + v_k$ satisfying the conclusions of Proposition 6, i.e.

$$\|\mathcal{M}(\bar{v}_{k-1} - u_k)\|_{\mathrm{op}} \leq \varepsilon_k, \; \mathrm{nnz}(w_k) \leq \frac{C_{\mathrm{sparse}} n}{\varepsilon_k^2}, \; \|\mathcal{M}(v_k)\|_{\mathrm{op}} \leq \frac{1}{10}.$$

Finally, we update $\bar{v}_k \leftarrow 10 v_k$ and $\bar{w}_k \leftarrow \bar{w}_{k-1} + \frac{1}{10^{k-1}} w_k$. We observe that for all $k \in [K]$, the above guarantees imply that $\|\mathcal{M}(\bar{v}_k)\|_{\mathrm{op}} \leq 1$ and hence it was valid to run Proposition 6. Moreover,

$$\bar{w}_K = \sum_{k \in [K]} (\bar{w}_k - \bar{w}_{k-1}) = \sum_{k \in [K]} \frac{1}{10^{k-1}} (u_k - v_k)$$

$$= \sum_{k \in [K]} \frac{1}{10^{k-1}} (u_k - \bar{v}_{k-1}) + \sum_{k \in [K]} \frac{1}{10^{k-1}} (\bar{v}_{k-1} - v_k)$$

$$= \sum_{k \in [K]} \frac{1}{10^{k-1}} (u_k - \bar{v}_{k-1}) + \bar{v}_0 - \frac{1}{10^{K-1}} v_K.$$

Therefore, since $\bar{v}_0 = \mathbb{1}_m$, we can rearrange the above and apply the triangle inequality to obtain

$$\|\mathcal{M}(\bar{w}_K) - \mathbf{I}_n\|_{\mathrm{op}} \leq \sum_{k \in [K]} \frac{1}{10^{k-1}} \|\mathcal{M}(\bar{v}_{k-1} - u_k)\|_{\mathrm{op}} + \frac{1}{10^{K-1}} \|\mathcal{M}(v_K)\|_{\mathrm{op}}$$

$$\leq \sum_{k \in [K]} \frac{\varepsilon_k}{10^{k-1}} + \frac{1}{10^K} = O(\varepsilon).$$

Finally, the sparsity of $\bar{w}_K$ is clearly $O(\frac{n}{\varepsilon^2})$ since the sparsities of each w_k are geometrically decreasing, and the conclusion follows by shifting ε by a constant and δ by a factor of K. We note that we do not pay a K-factor overhead in the runtime of Proposition 6 as ε_k is geometrically increasing. $\square$

We remark that Theorem 5 does not strongly rely on the assumption that $\mathcal{M}(\mathbb{1}_m) = \mathbf{I}_n$: indeed, any initially $O(1)$-bounded operator norm matrix sum enjoys the same guarantees up to constant factors. However, to obtain a ε-multiplicative spectral sparsification guarantee via Theorem 5, we require a constant spectral lower bound as well. Finally, the restriction $\delta \geq \exp(-\Omega(n))$ can be removed at a logarithmic overhead by running independent copies with constant failure probability.

5.3 Graph sparsification

We now specialize the machinery in this section to the setting of graphs. Throughout this section let $G = (V, E, w_G)$ be a weighted graph, and let $m := |E|$ and $n := |G|$; we define the matrices $\mathbf{B}$, $\mathbf{W}$, and $\mathbf{L}$ as in Section 2. We recall the following result on constructing an approximation to $\mathbf{L}^\dagger$.

Proposition 7 ([KMP11]). *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0, 1)$, there is a randomized algorithm which returns a linear operator $\widetilde{\mathbf{L}} : \mathbb{R}^V \to \mathbb{R}^V$ such that we can run the algorithm and apply $\widetilde{\mathbf{L}}$ in time $O(m \log^{1+o(1)}(n) \log \frac{1}{\varepsilon})$, and with high probability in n,*

$$(1 - \varepsilon)\mathbf{L}^\dagger \preceq \widetilde{\mathbf{L}} \preceq (1 + \varepsilon)\mathbf{L}^\dagger.$$

We now consider applying Theorem 5 to the matrices

$$\mathbf{M}_e := \mathbf{W}^{\frac{1}{2}}\mathbf{B}\widetilde{\mathbf{L}}(w_e b_e b_e^\top)\widetilde{\mathbf{L}}\mathbf{B}^\top\mathbf{W}^{\frac{1}{2}}, \text{ for all } e \in E,$$

where $\widetilde{\mathbf{L}}$ is the matrix from Proposition 7 with parameter ε, and $w := w_G$. Since

$$(1 - \varepsilon)^2\mathbf{L}^\dagger \preceq \widetilde{\mathbf{L}}\mathbf{L}\widetilde{\mathbf{L}} \preceq (1 + \varepsilon)^2\mathbf{L}^\dagger,$$

and $\mathcal{M}(\mathbb{1}_E) = \mathbf{W}^{\frac{1}{2}}\mathbf{B}\widetilde{\mathbf{L}}\mathbf{L}\widetilde{\mathbf{L}}\mathbf{B}^\top\mathbf{W}^{\frac{1}{2}}$, defining $\mathbf{\Pi_L} := \mathbf{W}^{\frac{1}{2}}\mathbf{B}\mathbf{L}^\dagger\mathbf{B}^\top\mathbf{W}^{\frac{1}{2}}$, we have

$$(1 - \varepsilon)^2\mathbf{\Pi_L} \preceq \mathcal{M}(\mathbb{1}_E) \preceq (1 + \varepsilon)^2\mathbf{\Pi_L}.$$

Notice that $\mathbf{\Pi_L}$ is the projection matrix onto the column span of $\mathbf{W}^{\frac{1}{2}}\mathbf{B}$. It is straightforward to check that the argument in Section 5.2 then returns a reweighting x such that $\mathcal{M}(x - \mathbb{1}_E) \preceq \varepsilon\mathbf{\Pi_L}$, with $\mathrm{nnz}(x) = O(\frac{n}{\varepsilon^2})$; this is made explicit by Theorem 4, which only cares about the trace bound on $\mathcal{M}(\mathbb{1}_E)$ (which is n), rather than the ambient dimension m. Finally, assuming ε is sufficiently small, we may rewrite this condition as

$$(1 - 4\varepsilon)\mathbf{\Pi_L} \preceq \mathcal{M}(w) \preceq (1 + 4\varepsilon)\mathbf{\Pi_L}$$

$$\implies (1 - 4\varepsilon)\mathbf{L} \preceq \mathbf{L}\widetilde{\mathbf{L}}\left(\sum_{e \in E} x_e w_e b_e b_e^\top\right)\widetilde{\mathbf{L}}\mathbf{L} \preceq (1 + 4\varepsilon)\mathbf{L}$$

$$\implies (1 - 7\varepsilon)\mathbf{L} \preceq \sum_{e \in E} x_e w_e b_e b_e^\top \preceq (1 + 7\varepsilon)\mathbf{L}.$$

In the second line, we left-multiplied all matrices by $\mathbf{B}^\top\mathbf{W}^{\frac{1}{2}}$ and right-multiplied by $\mathbf{W}^{\frac{1}{2}}\mathbf{B}$, and in the third line, we first multiplied on both sides by $\mathbf{L}^\dagger$, and then multiplied on both sides by $\widetilde{\mathbf{L}}^\dagger$ (using that $\widetilde{\mathbf{L}}$ and $\mathbf{L}$ share a kernel) and used Proposition 7. Finally, the matrix-vector multiplication time to all $\{\mathbf{M}_e\}_{e \in E}$ is dominated by applying $\widetilde{\mathbf{L}}$. Adjusting ε by a constant, we obtain our claimed result on linear-sized sparsification, reproduced here for convenience.

Theorem 1. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0, 1)$, there is a randomized algorithm which in time $\breve{O}(m \log^4(m) \log(\frac{1}{\varepsilon}) \cdot \varepsilon^{-3.5})$ returns $w \in \mathbb{R}_{\geq 0}^E$ satisfying $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$, such that with high probability in n,*

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

6 Ultrasparsifiers

In this section, we give a short application of the techniques in Section 5 to the setting of ultrasparsifiers. We will be concerned with the following problem: we are given a set of matrices $\{\mathbf{M}_i\}_{i\in[m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ and an additional matrix $\mathbf{N} \in \mathbb{S}_{\succeq \mathbf{0}}^n$, satisfying $\mathcal{M}(\mathbb{1}_m) + \mathbf{N} = \mathbf{I}_n$ where $\mathcal{M}$ is the operator associated with the set in the sense of (1). We also are promised

$$\mathrm{Tr}(\mathcal{M}(\mathbb{1}_m)) \leq \tau,$$

for a parameter τ such that $\frac{n}{\tau}$ is sufficiently large. We also require the relatively mild assumption that n is subexponential in m; indeed, in our applications $m \geq n$. Our goal is to construct $w \in \mathbb{R}_{\geq 0}^m$ such that $\mathrm{nnz}(w) = O(\tau)$, and $\|\mathcal{M}(w) + \mathbf{N} - \mathbf{I}_n\|_{\mathrm{op}} \leq \frac{1}{2}$. We explain the connections of this problem to primitives in spectral graph theory at the end of the section, after giving our sparsification algorithm in Theorem 6. We begin by stating the following variant of Proposition 6.

Corollary 4. *Let $\varepsilon \in (0,1)$, $\delta \in (\exp(-\Omega(\tau)), 1)$ for an appropriate constant, and let $\{\mathbf{M}_i\}_{i\in[m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ satisfy $\mathcal{M}(\mathbb{1}_m) \preceq \mathbf{I}_n$ and $\mathrm{Tr}(\mathcal{M}(\mathbb{1}_m)) \leq \tau$, where $\frac{n}{\tau}$ is sufficiently large and $n = \exp(O(m))$ for an appropriate constant. There is an algorithm returning $u = v + w$ for $v, w \in \mathbb{R}_{\geq 0}^m$ satisfying*

$$\|\mathcal{M}(u - \mathbb{1}_m)\|_{\mathrm{op}} \leq \varepsilon, \ \mathrm{nnz}(w) \leq \frac{C_{\mathrm{sparse}}\tau}{\varepsilon^2}, \ \|\mathcal{M}(v)\|_{\mathrm{op}} \leq \frac{1}{10},$$

with probability $\geq 1 - \delta$, for a universal constant C_{sparse}, in time

$$O\left(\left(\sum_{i\in[m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{M}_i)\right) \cdot \frac{\log(n)\log^2(\frac{mn}{\delta})\log\log^{11+o(1)}(\frac{n}{\delta})}{\varepsilon^{3.5}}\right).$$

Proof. The proof is essentially identical to the proof of Proposition 6, so we summarize the differences here, following the notation of our earlier proof. First, we initialize with the reweighting from Lemma 10 with a sparsity parameterized by τ; there are still $K = \log\log \frac{n}{\delta}$ phases, each of which shrinks the sparsity of a maintained w_k by a constant factor, until it reaches $\frac{C_{\mathrm{sparse}}\tau}{\varepsilon^2}$. We use

$$\rho_k = \max\left(2C_{\mathrm{set}}\sqrt{\frac{\tau}{m_{k-1}}}, \frac{\varepsilon}{(K-k+1)\log^2(K-k+1)}\right),$$

and so for sufficiently large C_{sparse} we may continue using the measure lower bound from Corollary 3 at each phase with the framework from Section 3 to preserve the necessary invariants. The remaining parameters and runtime analysis of this algorithm are all the same as in Proposition 6. $\square$

Theorem 6. *Let $\delta \in (\exp(-\Omega(\tau)), 1)$ for an appropriate constant, and let $\{\mathbf{M}_i\}_{i\in[m]} \subset \mathbb{S}_{\succeq \mathbf{0}}^n$ and $\mathbf{N} \in \mathbb{S}_{\succeq \mathbf{0}}^n$ satisfy $\mathcal{M}(\mathbb{1}_m) + \mathbf{N} = \mathbf{I}_n$ and $\mathrm{Tr}(\mathcal{M}(\mathbb{1}_m)) \leq \tau$, where $\frac{n}{\tau}$ is sufficiently large and $n = \exp(O(m))$ for an appropriate constant. There is an algorithm which returns $w \in \mathbb{R}_{\geq 0}^m$ satisfying $\mathrm{nnz}(w) = O(\tau)$ and $\|\mathcal{M}(w) + \mathbf{N} - \mathbf{I}_n\|_{\mathrm{op}} \leq \frac{1}{2}$, with probability $\geq 1 - \delta$, in time*

$$O\left(\left(\sum_{i\in[m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{M}_i)\right) \cdot \log(n)\log^2\left(\frac{n}{\delta}\right)\log\log^{11+o(1)}\left(\frac{n}{\delta}\right)\right).$$

Proof. We recursively apply Corollary 4 in the same way Proposition 6 was used in Theorem 5. This yields a $O(\tau)$-sparse reweighting w satisfying $\|\mathcal{M}(w - \mathbb{1}_m)\|_{\mathrm{op}} = \|\mathcal{M}(w) + \mathbf{N} - \mathbf{I}_n\|_{\mathrm{op}} \leq \frac{1}{2}$, where it suffices to take $\varepsilon \leq \frac{1}{2}$ to be a sufficiently small constant in the proof of Theorem 5. $\square$

We now describe consequences of Theorem 6. As discussed in Section 1, the state-of-the-art ultrasparsifiers constructible in near-linear time have parameters $(\breve{O}(\ell \log^2(n)), \ell)$, but there are higher-quality $(\min(\breve{O}(\ell \log(n)), \ell^{1+o(1)} \log^{o(1)}(n)), \ell)$-ultrasparsifiers constructible in polynomial time [KMST10, JS21]. We show how to match [KMST10, JS21] in near-linear time. Our starting point (as is the starting point of prior constructions [ST04, KMST10, KMP11, JS21]) is the following notion of a low-distortion subgraph (generalizing the combinatorial notion of a low-stretch spanning tree, as is demonstrated formally by Theorem 2.1 of [SW09]).

Definition 2. *We say subgraph H is a σ-distortion subgraph of $G = (V, E, w_G)$ if $\mathrm{Tr}(\mathbf{L}_H^\dagger \mathbf{L}_G) \leq \sigma$.*

We further recall the following known constructions of low-distortion subgraphs. The first result was used in [KMST10] and the latter was both developed and used in [JS21].

Proposition 8 ([AN19]). *Let $G = (V, E, w_G)$ have $m = |E|$ and $n = |V|$. There is an algorithm running in time $O(m \log n \log \log n)$ which produces a $O(m \log n \log \log n)$-distortion subgraph of G. Further, this subgraph is a spanning tree of G.*

Proposition 9 ([JS21]). *Let $G = (V, E, w_G)$ have $m = |E|$ and $n = |V|$. There is an algorithm which takes a parameter $\gamma > 1$ and runs in time $O(m)$, which produces a $O(m\gamma^{o(1)}(\log n)^{o(1)})$-distortion subgraph of G with $n - 1 + \frac{m}{\gamma}$ edges.*

Because the statement of Proposition 9 is not explicit in [JS21], we discuss how to obtain it in Appendix C. We now combine Propositions 8 and 9 with Theorem 6. First, by applying Theorem 1 with constant ε, we may assume that $m = O(n)$ at the cost of constant spectral distortion and a near-linear runtime overhead. Overloading G to mean the graph after applying this preprocessing, we let H be a σ-distortion subgraph of G, and for a parameter $\kappa \geq 1$, we let

$$\mathbf{N} := (\kappa \mathbf{L}_H + \mathbf{L}_G)^{\frac{\dagger}{2}} (\kappa \mathbf{L}_H) (\kappa \mathbf{L}_H + \mathbf{L}_G)^{\frac{\dagger}{2}},$$

$$\mathbf{M}_e := (\kappa \mathbf{L}_H + \mathbf{L}_G)^{\frac{\dagger}{2}} (w_e b_e b_e^\top) (\kappa \mathbf{L}_H + \mathbf{L}_G)^{\frac{\dagger}{2}} \text{ for all } e \in E.$$

It is straightforward to see that $\mathbf{N} + \sum_{e \in E} \mathbf{M}_e = \mathbf{I}_{V \setminus \mathbb{1}}$. Further, the distortion definition guarantees

$$\mathrm{Tr}\left(\sum_{e \in E} \mathbf{M}_e\right) = \mathrm{Tr}\left((\kappa \mathbf{L}_H + \mathbf{L}_G)^\dagger \mathbf{L}_G\right) \leq \mathrm{Tr}\left((\kappa \mathbf{L}_H)^\dagger \mathbf{L}_G\right) \leq \tau := \frac{\sigma}{\kappa}.$$

Finally, we note that (spectral approximations of) the square roots of Laplacian pseudoinverses may be applied in nearly-linear time, using the same strategy as in Section 5.3. Therefore, Theorem 6 (with our other tools) shows that we can efficiently obtain a graph on $n - 1 + O(\frac{\sigma}{\kappa})$ edges whose Laplacian is an $O(1)$-spectral approximation to $\kappa \mathbf{L}_H + \mathbf{L}_G$, which itself is a $O(\kappa)$-spectral approximation to $\mathbf{L}_G$. By taking H as in Proposition 8 or 9 and κ large enough, and reparameterizing the problem in terms of ℓ (the edge reduction factor), we achieve a $\min(\breve{O}(\ell \log(n)), \ell^{1+o(1)} \log^{o(1)}(n))$-spectral approximation to $\mathbf{L}_G$ on $n - 1 + \frac{n}{\ell}$ edges in nearly-linear time. We record this construction formally in Corollary 1, restated for convenience: the runtime comes from combining Theorem 6 with Section 5.3, as the runtimes of Propositions 8, 9 do not dominate.

Corollary 1. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\ell \geq 1$, there is a randomized algorithm which in time $\breve{O}(m \log^4(m))$ returns a $(\min(\breve{O}(\ell \log(n)), \ell^{1+o(1)} \log^{o(1)}(n)), \ell)$-ultrasparsifier of G, with high probability in n.*

In particular, Corollary 1 yields the main ultrasparsifier results of [KMST10, JS21] with an alternative proof (without going through the [BSS14] analysis), and much faster algorithms.

7 Degree-preserving sparsifiers

In this section, we consider a degree-preserving variant of the specialization of Section 5 to graphs. For an undirected graph $G = (V, E, w_G)$ and a parameter $\varepsilon \in (0, 1)$ (both fixed throughout), we define associated matrices and vectors $\mathbf{L}$, $|\mathbf{B}|$, $\mathbf{B}$, $\{b_e\}_{e \in E}$ and $\mathbf{W}$ as in Section 2. Our goal is to obtain a sparse $w \in \mathbb{R}_{\geq 0}^E$ with

$$(1 - \varepsilon)\mathbf{L} \preceq \sum_{e \in E}[w \circ w_G]_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L} \text{ and } |\mathbf{B}|\mathbf{W}(w - \mathbb{1}_E) = 0,$$

i.e. the degrees of the reweighted graph are unchanged from G. In Section 7.1 and Section 7.2 we develop our main optimization subroutines for implementing the sparsification framework of [RR20] with linear constraints. We then give a degree-preserving rounding procedure in Section 7.3, and prove our main result Theorem 2 in Section 7.4. Finally, in Section 7.5 we show how to slightly improve the runtime of Theorem 2 at the cost of a logarithmic overhead in the sparsity.

7.1 Approximate Gaussian rounding with linear constraints

In this section, we provide a subroutine for optimizing the subproblems required by Proposition 2, under additional linear constraints denoted $\mathbf{C} \in \mathbb{R}^{d \times m}$. We assume $d \leq (1 - 2c_{\text{tight}})m$ (where c_{tight} is the constant from Proposition 1). Consideration of this setting is motivated by the tolerance of Proposition 1 to constraining a sufficiently small number of linear directions. In applications, $\mathbf{C}$ is a reweighted signed edge-vertex incidence matrix of a subgraph of G, with $d \leq n \ll m$.

As in Section 5.1, we assume we are given $\{\mathbf{A}_i\}_{i \in [m]} \subseteq \mathbb{S}_{\succeq \mathbf{0}}^n$ with associated operator $\mathcal{A}$, a vector $g \in \mathbb{R}^m$ satisfying Assumption 1, and a parameter ρ fixed throughout. We also explicitly assume $\|\mathcal{A}(\mathbb{1}_m)\|_{\text{op}} \leq 2$, and define the discrepancy body $\mathcal{K}_{\rho, \mathcal{A}}$ as in (2) with respect to the given matrices. Our goal in this section will be to approximate the value $r_\star^{\mathbf{C}}$ and point $x_\star^{\mathbf{C}}$ defined as

$$r_\star^{\mathbf{C}} := \|x_\star - g\|_2, \text{ where } x_\star^{\mathbf{C}} := \arg \min_{\substack{x \in [-1,1]^m \cap \mathcal{K}_{\rho, \mathcal{A}} \\ \mathbf{C}x = \mathbb{0}_d}} \|x - g\|_2. \tag{12}$$

We recall that f_λ is defined in (5) (where $f(x) = \|\mathcal{A}(x)\|_{\text{op}}$ in this section), and we define

$$\mathcal{X}^{\mathbf{C}} := \{x \in [-1, 1]^m \mid \mathbf{C}x = \mathbb{0}_d\}. \tag{13}$$

In the remainder of this section we will discuss approximately solving f_λ over $\mathcal{X}^{\mathbf{C}}$. Our algorithm will use a variant of a Frank-Wolfe method, prompting the following oracle definition.

Definition 3 (Linear optimization oracle). *Let $\mathcal{X} \subseteq \mathbb{R}^m$ be a symmetric convex set, and let $\Delta \in (0, 1)$. We say $\mathcal{O}_{\text{lin-opt}}$ is a Δ-approximate linear optimization oracle over $\mathcal{X}$ if on input $g \in \mathbb{R}^m$ it returns $z \in \mathcal{X}$ satisfying $c^\top z \leq (1 - \Delta) \min_{x \in \mathcal{X}} c^\top x$.*

We note that because $\mathcal{X}$ is symmetric, Definition 3 is sensible as the minimum is always nonpositive. Before giving our optimization procedure for f_λ, we give a helper fact, where we collect several properties of a well-studied smooth approximation to the operator norm.

Fact 7 ([Nes07]). *Let $\mu > 0$, let $\widetilde{\mathcal{A}}$ to be the operator associated with $\{\widetilde{\mathbf{A}}_i\}_{i \in [m]} \in \mathbb{R}^{2n \times 2n}$ as defined in (5), and let $A_\mu(x) := \mu \log \operatorname{Tr} \exp(\frac{1}{\mu}\widetilde{\mathcal{A}}(x))$. Then $\|\mathcal{A}(x)\|_{\text{op}} \leq A_\mu(x) \leq \|\mathcal{A}(x)\|_{\text{op}} + \mu \log(2n)$ for all $x \in \mathbb{R}^m$, and A_μ is $\frac{1}{\mu}$-smooth in the ℓ_∞ norm, i.e. for all $x, x' \in \mathbb{R}^m$,*

$$A_\mu(x') \leq A_\mu(x) + \langle \nabla A_\mu(x), x' - x \rangle + \frac{1}{2\mu} \|x' - x\|_\infty^2.$$

Further, $\nabla A_\mu(x) = \widetilde{\mathcal{A}}^*(\mathbf{Y}(x))$, where $\mathbf{Y}(x) = \exp(\frac{1}{\mu}\widetilde{\mathcal{A}}(x)) \cdot (\operatorname{Tr} \exp(\frac{1}{\mu}\widetilde{\mathcal{A}}(x)))^{-1}$.

We also define the notion of an approximate gradient oracle, and recall a (standard) efficient construction of such an oracle from prior work based on approximations to the exponential.

Definition 4 (Approximate gradient oracle). *Following the notation of Fact 7, we say $g : [-1, 1]^m \to \mathbb{R}^m$ is a Δ-approximate gradient oracle for A_μ if $\|g(x) - \nabla A_\mu(x)\|_1 \le \Delta$ for all $x \in [-1, 1]^m$.*

The following claim applies Proposition 2 of [JT23] with $\varepsilon \leftarrow O(\Delta)$ and $\gamma \leftarrow O(\frac{\Delta}{n})$.

Lemma 11 (Proposition 2, [JT23]). *We can implement a Δ-approximate gradient oracle for A_μ which succeeds with probability $\ge 1 - \delta$ in time*

$$O\left(\left(\sum_{i\in[m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right) \cdot \frac{\log^2\left(\frac{mn}{\mu\Delta\delta}\right)}{\sqrt{\mu}\Delta^2}\right)$$

We are now ready to give our main Frank-Wolfe subroutine.

Lemma 12. *Let $\beta, \rho, \delta \in (0, 1)$, $\lambda \ge \frac{\rho}{8m}$, and let $\mathcal{O}_{\mathrm{lin\text{-}opt}}$ be a Δ-approximate linear optimization oracle over $\mathcal{X}^{\mathbf{C}}$ for $\Delta := \frac{c_{\mathrm{tight}}\beta^2}{240}$. There is an algorithm which solves f_λ to additive error $\frac{\lambda c_{\mathrm{tight}}m\beta^2}{16}$ over $\mathcal{X}^{\mathbf{C}}$ using $N = O(\frac{\log n}{\rho^2\beta^4})$ calls to $\mathcal{O}_{\mathrm{lin\text{-}opt}}$ and*

$$O\left(\left(\sum_{i\in[m]} \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)\right) \cdot \left(\frac{\log^{3.5}\left(\frac{mn}{\rho\beta\delta}\right)}{\rho^{4.5}\beta^9}\right)\right)$$

additional time, with probability $\ge 1 - \delta$.

Proof. Our algorithm will be a Frank-Wolfe method applied to the smooth approximation from Fact 7. Throughout the proof, define for convenience $x^\star$ to be the minimizer of f_λ over $\mathcal{X}^{\mathbf{C}}$, and

$$\mu := \frac{\lambda c_{\mathrm{tight}}m\beta^2}{48\log(2n)}, \quad F(x) := A_\mu(x) + \lambda\|x - g\|_2^2.$$

By the fact that $\|\cdot\|_2^2 \le m\|\cdot\|_\infty^2$, we have that F is L-smooth in the ℓ_∞ norm for

$$L := \frac{1}{\mu} + 2\lambda m.$$

Finally, we let g be a Δ'-approximate gradient oracle for A_μ, for

$$\Delta' := \frac{L}{N+1}.$$

Our algorithm defines $\eta_t := \frac{2}{t+1}$ for all $t \in [N]$ and initializes at $x_1 = \mathbb{0}_m$. It then repeatedly updates $x_{t+1} \leftarrow (1 - \eta_t)x_t + \eta_t z_t$ where z_t is the result of calling $\mathcal{O}_{\mathrm{lin\text{-}opt}}$ on $g(x_t) + 2\lambda(x_t - g)$. In each iteration, since $g(x_t) + 2\lambda(x_t - g)$ approximates $\nabla F(x_t)$ up to Δ' in the ℓ_1 norm,

$$F(x_{t+1}) \le F(x_t) + \eta_t\langle\nabla F(x_t), z_t - x_t\rangle + \frac{L\eta_t^2}{2}\|z_t - x_t\|_\infty^2 + 2\eta_t\Delta'$$
$$\le F(x_t) + \eta_t\langle\nabla F(x_t), (1 - \Delta)x^\star - x_t\rangle + 3L\eta_t^2,$$

since $\|z_t - x_t\|_\infty \le 2$ for $z_t, x_t \in \mathcal{X}^{\mathbf{C}}$, and we applied Definition 3 with $x_\star \in \mathcal{X}^{\mathbf{C}}$. Now defining $E_t := F(x_t) - F((1 - \Delta)x_\star)$, the above display and convexity of F implies

$$E_{t+1} \le E_t + \eta_t\langle\nabla F(x_t), (1 - \Delta)x^\star - x_t\rangle + 2L\eta_t^2 \le (1 - \eta_t)E_t + 3L\eta_t^2. \tag{14}$$

24

We next claim that inductively, $E_t \leq \frac{12L}{t+1}$ for all $t \in [N]$. The base case of the induction follows from nonnegativity of F everywhere, so Fact 7 and Assumption 1 imply

$$E_1 \leq F(\mathbb{0}_m) \leq \mu \log(2n) + 4\lambda m \leq 4L.$$

Further, inductively applying (14) preserves this invariant:

$$E_{t+1} \leq \frac{t-1}{t+1} \cdot \frac{12L}{t+1} + \frac{12L}{(t+1)^2} < \frac{12L}{t+2}.$$

By taking the constant in front of N large enough, and using $\lambda \geq \frac{\rho}{8m}$, this implies

$$\begin{aligned}
f_\lambda(x_N) - f_\lambda(x_\star) &\leq F(x_N) - F(x_\star) + \mu \log(2n) \\
&\leq F(x_N) - F((1-\Delta)x_\star) + \Delta F(\mathbb{0}_m) + \mu \log(2n) \\
&\leq \frac{\lambda c_{\text{tight}} m\beta^2}{48} + \Delta F(\mathbb{0}_m) + \frac{\lambda c_{\text{tight}} m\beta^2}{48} \leq \frac{\lambda c_{\text{tight}} m\beta^2}{16}.
\end{aligned}$$

where the first inequality used Fact 7, the second used nonnegativity and convexity of F, the third used our bound on E_N, and the last used $F(\mathbb{0}_m) \leq \mu \log(2n) + 4\lambda m \leq 5\lambda m$. The runtime comes from applying Lemma 11 N times, with $\delta \leftarrow \frac{\delta}{N}$, and union bounding for the failure probability. Here, we note that it suffices to take $\Delta' = \Theta(\rho\beta^2)$ and $\frac{1}{\mu} = O(\frac{\log n}{\rho\beta^2})$. $\qquad \square$

To complete our use of Lemma 12, we will give an efficient linear optimization oracle in Section 7.2 when $\mathcal{X} = \mathcal{X}^{\mathbf{C}}$ and the constraint matrix $\mathbf{C}$ is graph-structured.

7.2 Degree-preserving linear optimization oracle

We recall the following standard facts from prior work (e.g. [BCL94]) used in our analysis.

Fact 8. *For $p \geq 2$, $\frac{1}{2}\|\cdot\|_p^2$ is $(p-1)$-smooth in the ℓ_p norm where the domain is $\mathbb{R}^d$ for any $d \in \mathbb{N}$. Additionally, for any $x \in \mathbb{R}^d$ and $p \geq q \geq 1$, we have $\|x\|_p \leq \|x\|_q \leq d^{\frac{1}{q}-\frac{1}{p}} \|x\|_p$.*

Further, in this section we will require the notion of an oblivious routing.

Definition 5. *We say $\mathbf{R} \in \mathbb{R}^{E \times V}$ is an α-oblivious routing for a graph $G = (V, E, w_G)$ if $\mathbf{B}^\top \mathbf{R} x = x$ for all $x \in \mathbb{R}^V$, and $\|\mathbf{W}^{-1}\mathbf{R}\mathbf{B}^\top \mathbf{W} y\|_\infty \leq \alpha \|y\|_\infty$ for all $y \in \mathbb{R}^E$. We let $\mathcal{T}_{\text{route}}(\alpha)$ be the time required to construct and apply an α-oblivious routing when G is clear from context.*

We refer the reader to [KLOS14] for more exposition on the history and uses of oblivious routings. We also recall the following construction of an oblivious routing from that paper.

Proposition 10 (Theorem 19, [KLOS14])**.** *Let $G = (V, E, w_G)$ have $m = |E|$ and $n = |V|$. Then $\mathcal{T}_{\text{route}}(\alpha) = mn^{o(1)}$ for some $\alpha = n^{o(1)}$, with high probability in n.*

Finally, we give two simple helper claims used in our proof.

Lemma 13. *For some $p \geq 1$, $v \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m \times m}$, let $\mathrm{OPT}_{v,p} := \min_{\|\mathbf{A}z\|_p \leq 1} \langle v, \mathbf{A}z \rangle$. Then*

$$\min_{z \in \mathbb{R}^m} \langle v, \mathbf{A}z \rangle + \frac{1}{2} \|\mathbf{A}z\|_p^2 = -\frac{1}{2}\mathrm{OPT}_{v,p}^2.$$

Further, for any $\Delta \in (0, 1)$ and $w \in \mathbb{R}^m$ with

$$\min_{w \in \mathbb{R}^m} \langle v, \mathbf{A}w \rangle + \frac{1}{2} \|\mathbf{A}w\|_p^2 \leq -\frac{(1-\Delta)^2}{2}\mathrm{OPT}_{v,p}^2,$$

we can obtain w', a rescaling of w, satisfying $\|\mathbf{A}w'\|_p \leq 1$ and $\langle v, \mathbf{A}w' \rangle \leq (1-\Delta)\mathrm{OPT}_{v,p}$.

Proof. Letting $z^\star$ achieve value $\mathrm{OPT}_{v,p} \leq 0$, by setting $z \leftarrow \lambda z^\star$ for $\lambda = -\mathrm{OPT}_{v,p}$, we see that z attains value $-\frac{1}{2}\mathrm{OPT}_{v,p}^2$ in the above display; here we use $\|\mathbf{A}z^\star\|_p = 1$, as otherwise we could scale up $z^\star$. To see that no z attains smaller value, suppose otherwise, and let z minimize the above display achieving value $-\frac{1}{2}C^2 < -\frac{1}{2}\mathrm{OPT}_{v,p}^2$. Since rescaling z by any λ does not improve the value,

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\lambda \langle v, \mathbf{A}z \rangle + \frac{\lambda^2}{2}\|\mathbf{A}z\|_p^2\right) = 0 \text{ at } \lambda = 1 \iff \langle v, \mathbf{A}z \rangle = -\|\mathbf{A}z\|_p^2 = -C^2.$$

Therefore, $\frac{1}{C}\|\mathbf{A}z\|_\infty \leq 1$, and $\frac{z}{C}$ achieves value $-C < \mathrm{OPT}_{v,p}$ for the original problem, a contradiction. To see the second claim, suppose without loss of generality we have first rescaled so that $\langle v, \mathbf{A}w \rangle = -\|\mathbf{A}w\|_p^2 \leq -(1-\Delta)^2\mathrm{OPT}_{v,p}^2$, as otherwise rescaling by a constant λ improves the function value. We then rescale $w' \leftarrow \frac{w}{\|\mathbf{A}w\|_p}$, and the conclusion follows immediately. $\qquad\square$

Lemma 14. *Let $p \geq 1$, and suppose*

$$\langle v, \mathbf{A}z \rangle + \frac{1}{2}\|\mathbf{A}z\|_p^2 \leq 0,$$

where $\mathrm{OPT}_{v,p}$ is defined as in Lemma 13. Then $\|\mathbf{A}z\|_\infty \leq -2\mathrm{OPT}_{v,p}$.

Proof. First, we claim any z with $\|\mathbf{A}z\|_p > -2\mathrm{OPT}_{v,p}$ has $\langle v, \mathbf{A}z \rangle + \frac{1}{2}\|\mathbf{A}z\|_p^2 > 0$. To see this,

$$\langle v, \mathbf{A}z \rangle + \frac{1}{2}\|\mathbf{A}z\|_p^2 \geq \|\mathbf{A}z\|_p\left(\left\langle v, \mathbf{A}\frac{z}{\|\mathbf{A}z\|_p}\right\rangle\right) + \frac{1}{2}\|\mathbf{A}z\|_p^2$$

$$\geq \|\mathbf{A}z\|_p\left(\mathrm{OPT}_{v,p} + \frac{1}{2}\|\mathbf{A}z\|_p\right) > 0,$$

where we used the definition of $\mathrm{OPT}_{v,p}$. The conclusion follows from $\|\mathbf{A}z\|_\infty \leq \|\mathbf{A}z\|_p$. $\qquad\square$

We are now ready to give our linear optimization oracle.

Lemma 15. *We can implement a Δ-approximate linear optimization oracle over $\mathcal{X}^{\mathbf{B}^\top \mathbf{W}}$, where $\mathbf{B}$, $\mathbf{W}$ are associated with $G = (V, E, w_G)$, in time $O(\mathcal{T}_{\mathrm{route}}(\alpha) \cdot \frac{\alpha^2 \log|E|}{\Delta^2})$ for any $\alpha \geq 1$.*

Proof. Throughout we let $\mathbf{R}$ be an α-oblivious routing, identify V and E with $[n]$ and $[m]$ in the standard way for $n = |V|$ and $m = |E|$, and define $\mathbf{C} := \mathbf{I}_m - \mathbf{R}\mathbf{B}^\top$ and $\mathbf{A} := \mathbf{W}^{-1}\mathbf{C}\mathbf{W}$. By the definition of an oblivious routing, $\mathbf{B}^\top \mathbf{C}\mathbf{W}z = \mathbb{0}_n$ for all $z \in \mathbb{R}^m$. Hence, we may solve

$$\min_{\|\mathbf{A}z\|_\infty \leq 1} \langle c, \mathbf{A}z \rangle$$

to Δ-multiplicative accuracy, and output $x = \mathbf{A}z$, which satisfies

$$\mathbf{B}^\top \mathbf{W}x = \mathbf{B}^\top \mathbf{C}\mathbf{W}z = \mathbb{0}_n.$$

Next, note that for $p := \frac{5\log m}{\Delta}$, Fact 8 implies

$$\left(1 - \frac{\Delta}{4}\right)\mathrm{OPT}_{c,\infty} > \mathrm{OPT}_{c,p} > \mathrm{OPT}_{c,\infty},$$

so Lemma 13 implies it suffices to return a solution to

$$\min \langle c, \mathbf{A}z \rangle + \frac{1}{2}\|\mathbf{A}z\|_p^2 \tag{15}$$

26

up to $\frac{\Delta}{4}\mathrm{OPT}_{c,p}^2$ additive error. Next, the oblivious routing definition and triangle inequality imply

$$\max_{\|v\|_\infty=1} \|\mathbf{A}v\|_\infty \le 1 + \alpha \le 2\alpha,$$

so Fact 8 shows that the objective in (15) is $4\alpha^2 p$-smooth in the ℓ_∞ norm. Our linear optimization oracle iteratively takes a step of the ℓ_∞ gradient descent algorithm in Theorem 1 of [KLOS14] (on the objective (15)), and multiplies the iterate by $\mathbf{A}$. The objective value for iterates of ℓ_∞ gradient descent is monotone non-increasing, and multiplication by $\mathbf{A}$ preserves the objective value since it is simple to verify $\mathbf{A}^2 z = \mathbf{A}z$ for all z. Hence, Lemma 14 implies that the ℓ_∞ radius of all iterates is bounded by $O(\mathrm{OPT}_{v,p})$. A direct application of Theorem 1 in [KLOS14] finally implies the algorithm returns a point with function error $\frac{\Delta}{4}\mathrm{OPT}_{c,p}^2$ to (15) in

$$k = O\left(\frac{\alpha^2 p\,\mathrm{OPT}_{c,p}^2}{\Delta\,\mathrm{OPT}_{c,p}^2}\right) = O\left(\frac{\alpha^2 \log m}{\Delta^2}\right)$$

iterations, each of which is dominated by multiplication through $\mathbf{A}$, giving the runtime. $\qquad\square$

7.3 Degree-preserving rounding via link/cut trees

In this section, we give a self-contained subroutine solving the following degree-preserving rounding problem via an application of the link/cut tree dynamic data structure of [ST83].

Lemma 16. *Let* $\mathbf{B}$, $\mathbf{W}$ *be associated with* $G = (V, E, w_G)$, *define* $u_e := 2[w_G]_e$ *for all* $e \in E$, *and let* $n = |V|$, $m = |E|$. *There is an algorithm which runs in time* $O(m \log n)$ *and outputs* $w \in \prod_{e \in E}[0, u_e]$ *such that* $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$, *and* $w_e = 0$ *for at least* $\frac{m-(n-1)}{2}$ *of the* $e \in E$.

Proof. The algorithm is very similar to the cycle-cancelling data structure via link/cut trees in Appendix C of [AJJ$^+$22], so we first give a high-level description of the algorithm, and then discuss how to modify Appendix C of [AJJ$^+$22] to implement the algorithm in $O(m \log n)$ time. At the beginning of the algorithm, all edges in E are marked "alive," and we set $w \leftarrow w_G$. We then iteratively find a cycle C in the graph restricted to alive edges, and let

$$\Delta := \min_{e \in C} \min\left(u_e - w_e, w_e\right).$$

In other words, Δ is the closest distance of any current weight to saturating either the lower bound 0 or the upper bound u_e. We then alternately add and subtract Δ from w along the cycle so that one edge is saturated (i.e. either 0 or u_e), and mark the saturated edge as "dead." Inductively, $w \in \prod_{e \in E}[0, u_e]$ at all times, and $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$ is preserved. Further, at the end of the algorithm at most $n - 1$ edges are not saturated. If more saturated edges are set to 0, then we output w. Otherwise, we output $2w_G - w \in \prod_{e \in E}[0, u_e]$, which preserves the degrees as

$$|\mathbf{B}|^\top(2w_G - w) = 2|\mathbf{B}|^\top w_G - |\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G.$$

We now discuss how to implement this algorithm efficiently, following the notation of Appendix C of [AJJ$^+$22]. We will maintain four link/cut forests with the same topology, denoted T_{up}^+, T_{lo}^+, T_{up}^-, and T_{lo}^-, and process the edges of G sequentially. When we process an edge e, we add it to all four forests. The role of the forests T_{up}^- and T_{lo}^- is to maintain the distances of w_e from u_e and 0 respectively, for all e with odd depth; similarly, T_{up}^+ and T_{lo}^+ maintain distances of the weights of all even-depth edges from saturation. When we process an edge $e = (u, v)$, if its depth in the tree

containing u is odd, we set its weight in T_{up}^- and T_{lo}^- to $[w_G]_e$, and set its weight in T_{up}^+ and T_{lo}^+ to $2m\|w_G\|_\infty$; we symmetrically handle even-depth edges. Next, if the endpoints of e belong to different trees in the forest, we link these trees. Otherwise, we have found a cycle, and we can find the minimum distance to saturation Δ along the path between u and v in $O(\log n)$ time. By the weight setting of $2m\|w_G\|_\infty$ on odd-depth edges in T_{up}^+ and T_{lo}^+ at initialization, these edges will never minimize the weight on any path throughout the algorithm, since they stay at least $2\|w_G\|_\infty$; even-depth edges are handled identically. We then add or subtract Δ from the path in all four trees appropriately in $O(\log n)$ time, and delete the saturated edge; hence, the runtime is $O(m\log n)$. $\qquad\square$

7.4 Degree-preserving sparsifiers in almost-linear time

We now prove Theorem 2 after stating one helper lemma.

Lemma 17. *Let $G = (V, E, w_G)$ be bipartite with bipartition $V = L \cup R$, and let $\mathbf{B} \in \mathbb{R}^{E \times V}$ be its edge-incidence matrix signed such that for each $e = (u, v) \in E$ for $u \in L$ and $v \in R$, $\mathbf{B}_{e:}$ has a 1 in the u coordinate and a -1 in the v coordinate. Then $\mathbf{B}^\top x = 0$ implies $|\mathbf{B}|^\top x = 0$.*

Proof. It suffices to note that $|\mathbf{B}|^\top = \mathbf{D}\mathbf{B}^\top$, where $\mathbf{D}$ is a diagonal matrix with

$$\mathbf{D}_{vv} = \begin{cases} 1 & v \in L \\ -1 & v \in R \end{cases}.$$

$\qquad\square$

Theorem 2. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0, 1)$, there is a randomized algorithm which returns $w \in \mathbb{R}_{\geq 0}^E$ satisfying $\mathrm{nnz}(w) = O(\frac{n}{\varepsilon^2})$, $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$ in time*

$$O\left((m + \mathcal{T}_{\mathrm{route}}(\alpha)) \cdot \alpha^2 \cdot \mathrm{poly}\left(\frac{\log m}{\varepsilon}\right)\right),$$

for any $\alpha \geq 1$ following the notation of Definition 5, such that with high probability in n,

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

Proof. Our algorithm follows analogously to the proof of Proposition 6 and its specialization in Section 5.3, so we follow their notation, and identify V and E with $[n]$ and $[m]$ in the standard way. We initialize $w_0 \leftarrow \mathbb{1}_m$, and define

$$\mathbf{M}_e := \mathbf{W}^{\frac{1}{2}} \mathbf{B}\widetilde{\mathbf{L}}([w_G]_e b_e b_e^\top)\widetilde{\mathbf{L}}\mathbf{B}^\top \mathbf{W}^{\frac{1}{2}} \text{ for all } e \in E,$$

where we follow the notation of Section 5.3 in defining $\widetilde{\mathbf{L}}$. Our algorithm proceeds in $K = O(\log m)$ phases indexed by $k \in [K]$. For all $k \in [K]$, we let $m_{k-1} := \mathrm{nnz}(w_{k-1})$, and

$$\rho_k := \max\left(2C_{\mathrm{set}}\sqrt{\frac{n}{m_{k-1}}}, \frac{\varepsilon}{(K - k + 1)\log^2(K - k + 1)}\right).$$

We again terminate if $m_k \leq \frac{C_{\mathrm{sparse}}n}{\varepsilon^2}$ for a universal C_{sparse}. Otherwise, each phase k computes x_k such that $|\mathbf{B}|^\top(w_{k-1} \circ x_k) = \mathbb{0}_n$, and for at least $\frac{c_{\mathrm{tight}}}{30}m_{k-1}$ coordinates i, $[x_k]_i = -1$, and

$$\|\mathcal{M}(w_{k-1} \circ x_k)\|_{\mathrm{op}} = O(\rho_k).$$

28

We then update $w_k \leftarrow w_{k-1} \circ (\mathbb{1}_m + x_k)$, and the remainder of the proof follows analogously to Proposition 6 and Section 5.3. It remains to argue how to compute such a vector x_k.

First, we randomly partition the vertices in the support of w_{k-1} into two sets, and accept if the number of edges crossing the bipartition is at least $\frac{m_{k-1}}{3}$; this occurs with high probability in n after $O(\log n)$ random bipartitions, since by linearity of expectation and Markov's inequality, the number of edges not in the bipartition is at most $\frac{2m_{k-1}}{3}$ with constant probability. Let $G_k = (V, E_k, w_{k-1} \circ w_G)$ where E_k are the edges crossing the bipartition, let $\mathbf{B}_k$ be its signed edge-vertex incidence matrix, and let $\mathbf{A}_e = [w_{k-1}]_e \mathbf{M}_e$ for all $e \in E_k$. By Lemma 17, the condition $|\mathbf{B}_k|^\top([w_{k-1} \circ w_G \circ x_k]_{E_k}) = \mathbb{0}_n$ is implied by $\mathbf{B}_k^\top([w_{k-1} \circ w_G \circ x_k]_{E_k}) = \mathbb{0}_n$. Hence, letting $\mathbf{W}_k = \mathbf{diag}\left([w_{k-1} \circ w_G]_{E_k}\right)$, we draw $g \sim \mathcal{N}(\mathbb{0}_{m_{k-1}}, \mathbf{I}_{m_{k-1}})$, and solve both of the problems

$$\min_{x \in \mathcal{K}_{\rho_k, \mathcal{A}} \cap \mathcal{X}^{\mathbf{B}_k^\top \mathbf{W}_k}} \|x - g\|_2^2, \qquad \min_{x \in \mathcal{K}_{\rho_k, \mathcal{A}} \cap \mathcal{X}^{\mathbf{B}_k^\top \mathbf{W}_k}} \|x + g\|_2^2,$$

to additive error $\tau = \frac{c_{\text{tight}} m_{k-1} \rho_k^2}{4}$, where we follow the notation (13). By the same induction argument as in Proposition 6, we maintain $\left\|\mathcal{A}(\mathbb{1}_{m_{k-1}})\right\|_{\text{op}} \leq 2$ at all times, so combining Lemmas 12 and 15 (with $\beta \leftarrow \rho_k$) shows that we can achieve additive error τ with high probability in n in time

$$O\left(m \log^{1+o(1)}(m) \log\left(\frac{1}{\varepsilon}\right) \cdot \left(\frac{\log^{3.5}(m)}{\rho_k^{13.5}} + \frac{\log^2(m)}{\rho_k^{10}} \cdot \alpha^2 \mathcal{T}_{\text{route}}(\alpha)\right)\right),$$

for any $\alpha \geq 1$, where we bounded $\sum_{e \in E_k} \mathcal{T}_{\text{mv}}(\mathbf{A}_e) = O(m \log^{1+o(1)}(m) \log \frac{1}{\varepsilon})$ as in Section 5.3.

Let $\tilde{x}_k$ be the resulting approximate minimizer with more coordinates $\leq -1 + \rho_k$, and note that there is a set S_k with size $|S_k| \geq \frac{c_{\text{tight}} m_{k-1}}{12}$ such that $[\tilde{x}_k]_e \leq -1 + \rho_k$ for all $e \in S_k$. By applying Lemma 16 to the graph $(V, S_k, [(\mathbb{1}_{m_{k-1}} + \tilde{x}_k) \circ w_{k-1} \circ w_G]_{S_k})$, we can produce x_k which equals $\tilde{x}_k$ outside S_k, has at least $\frac{c_{\text{tight}} m_{k-1}}{30}$ coordinates equal to -1, maintains $\mathbf{B}_k^\top([w_{k-1} \circ x_k]_{E_k}) = \mathbb{0}_n$, and

$$\left\|\mathcal{A}([x_k - \tilde{x}_k]_{S_k})\right\|_{\text{op}} \leq \rho_k \left\|\mathcal{A}(\mathbb{1}_{m_k})\right\|_{\text{op}} = O(\rho_k).$$

The above inequality used that $x_k - \tilde{x}_k$ is bounded by $\pm \rho_k$ on S_k by the guarantees of Lemma 16. Combining this with $\|\mathcal{A}(\tilde{x}_k)\|_{\text{op}} = O(\rho_k)$ completes the proof. $\qquad \square$

Remark 1. *We remark that in the proof of Theorem 2, if the input graph G was bipartite, then there is no need to perform the step of randomly sampling a bipartition and restricting our algorithm to the edges crossing the bipartition (as Lemma 17 applies generically in this case, and bipartiteness is preserved by our reweightings). As a result, all of the graphs for which we construct oblivious routings for via the machinery in Section 7.2 are 1.1-multiplicative spectral approximations to G (taking ε sufficiently small), by the same induction argument as in Proposition 6.*

7.5 Runtime-sparsity tradeoffs

In this section, we describe how to achieve a slightly worse sparsity guarantee than Theorem 2 in near-linear time. Our technique in this section leverages an improved oblivious routing procedure in the case that our input graph is an expander. For completeness, we briefly define expander graphs.

Definition 6 (Expander graphs). *Let $G = (V, E, w_G)$ be a graph. For a set $S \subseteq V$, we let $\text{Vol}(S) = \sum_{v \in S} \delta_v$ be the sum of weighted degrees in S and $\partial S = \{e \in E : e \notin S \text{ and } e \notin E/S\}$ to be the edge boundary of S. The cut value of S is $w(\partial S) = \sum_{e \in \partial S}[w_G]_e$. The conductance of S is*

$$\Phi(S) = \frac{w(\partial S)}{\min\{\text{Vol}(S), \text{Vol}(V/S)\}}.$$

We say G is a ϕ-expander if $\Phi(S) \geq \phi$ for all $S \subseteq V$.

The main fact we use is that electric flows induce good oblivious routings on expanders. This is summarized in the following lemma from prior work, where $\mathbf{R}$ is an electric routing matrix.

Lemma 18 (Lemma 28, [KLOS14]). *Let $G = (V, E, w_G)$ be a ϕ-expander. Define*

$$\mathbf{R} = \mathbf{W}\mathbf{B}\mathbf{L}^\dagger.$$

Then $\mathbf{R}$ is an α-oblivious routing with $\mathcal{T}_{\text{route}}(\alpha) = \widetilde{O}(m)$ for $\alpha = O(\frac{\log m}{\phi^2})$.

Leveraging Lemma 18, we are now ready to give a $\widetilde{O}(m)$ time algorithm for constructing linear-sized degree-preserving sparsifiers of bipartite expanders.

Lemma 19. *Let $G = (V, E, w)$ be a bipartite ϕ-expander with $m = |E|$ and $n = |V|$. Given a parameter $\varepsilon \in (0, 1)$, there is a randomized algorithm which returns $w \in \mathbb{R}^E_{\geq 0}$ satisfying $\text{nnz}(w) = O(\frac{n}{\varepsilon^2})$, $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$ in time*

$$O\left(\frac{m}{\phi^2} \cdot \text{poly}\left(\frac{\log m}{\varepsilon} \right) \right)$$

such that with high probability in n,

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

Proof. We employ Theorem 2 on G, and use the routings from Lemma 18. In light of Remark 1, all graphs we compute oblivious routings on are 1.1-sparsifiers of G and are hence $\Omega(\phi)$-expanders. As such, Lemma 18 yields $\mathcal{T}_{\text{route}}(\alpha) = \widetilde{O}(m)$ with $\alpha = O(\frac{\log m}{\phi^2})$ for all graphs we construct oblivious routings for: substituting this into Theorem 2 yields the result. $\square$

We now upgrade this sparsification result on expander graphs to general graphs. Our main tool for this is a expander decomposition result from prior work [SW19].

Lemma 20 (Theorem 4.1, [SW19]). *There is a randomized algorithm which takes as input $\phi \in (0, 1)$ and a graph $G = (V, E, w_G)$ with $m = |E|$, where w_G has multiplicative range $\text{poly}(m)$, and total weight $W := \sum_{e \in E}[w_G]_e$. With high probability in n, it partitions V into $\{V_i\}_{i \in [k]}$ in time $O(\frac{m}{\phi} \log^5 m)$ such that the following properties hold.*

- *The induced subgraph of G on each V_i is a ϕ-expander.*

- *The total edge weight cut by the V_i is $O(\phi W \log^3 m)$.*

We combine this primitive with Lemma 19 to obtain the main result of this section.

Theorem 7. *Given a graph $G = (V, E, w_G)$ with $m = |E|$, $n = |V|$, and $\varepsilon \in (0, 1)$, assume w_G has multiplicative range $\text{poly}(m)$. There is a randomized algorithm which returns $w \in \mathbb{R}^E_{\geq 0}$ satisfying $\text{nnz}(w) = O(\frac{n \log(m)}{\varepsilon^2})$, $|\mathbf{B}|^\top w = |\mathbf{B}|^\top w_G$ in time $\widetilde{O}(m \cdot \varepsilon^{-O(1)})$ such that with high probability in n,*

$$(1 - \varepsilon)\mathbf{L}_G \preceq \sum_{e \in E} w_e b_e b_e^\top \preceq (1 + \varepsilon)\mathbf{L}_G.$$

Proof. We begin applying Lemma 20 to find bipartite expanders covering a constant fraction of G's edge weight. Let S be a set which includes each vertex in G with probability $\frac{1}{2}$: note that since each edge is cut by S with probability $\frac{1}{2}$, $\mathbb{E}[w(\partial S)] = \frac{1}{2}W$ by linearity of expectation. By repeating this $O(\log n)$ times, we find a set S with $w(\partial S) \geq \frac{1}{3}W$ with high probability in n. Let H be the graph containing all edges cut by S, and note that it is bipartite. Applying Lemma 20 to H with $\phi = O(\log^{-3}(m))$, we obtain induced subgraphs $\{H[V_i]\}_{i \in [k]}$ of H satisfying the following.

- Each $H[V_i]$ is a ϕ-expander.

- The $H[V_i]$ collectively contain $\frac{3}{4}$ of the edge-weight of H.

As subgraphs of bipartite graphs are bipartite, these $H[V_i]$ are bipartite ϕ-expanders which collectively cover $\frac{3}{4} \cdot \frac{1}{3} = \frac{1}{4}$ of the edge weight of G. By iterating this procedure on the remaining edges, we thus obtain disjoint subgraphs $\{G_i\}_{i \in [K]}$ of G for $K = O(\log m)$ satisfying the following.

- $G = \bigcup_{i \in [K]} G_i$.

- Each G_i is a disjoint union of bipartite $O(\log^{-3}(m))$-expanders.

We apply Lemma 19 to each expander in each G_i and combine the outputs. As the union of sparsifiers is a sparsifier of the union, the resulting graph is an ε-spectral sparsifier of our original G. Additionally, as the total number of vertices in all of the G_i is at most $nK = O(n \log m)$, the number of edges in the resulting sparsifier is bounded as $O(\frac{n \log m}{\varepsilon^2})$. Finally, as the G_i are disjoint they collectively contain m edges: the total runtime of the calls to Lemma 19 is thus the desired

$$O\left(m \cdot \mathrm{poly}\left(\frac{\log m}{\varepsilon}\right)\right).$$

$\square$

8   Spencer's theorem

In this section, we describe how we recover the main result of [JSS23] within our framework. Specifically, we prove the following result which is identical to Theorem 1.1 of [JSS23] after explicitly dropping any empty rows and columns of $\mathbf{A}$. We remark that for this section only, we will reverse the roles of "m" and "n" for consistency with [JSS23] (so there are n colors to choose).

Theorem 8. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfy $\max_{i \in [m], j \in [n]} |\mathbf{A}_{ij}| \leq 1$. There is an algorithm running in $\breve{O}(\mathrm{nnz}(\mathbf{A}) \cdot \log^5(n))$ time which, with probability at least $\frac{1}{2}$, returns $x \in \{\pm 1\}^n$ such that, for an absolute constant C,*

$$\|\mathbf{A}x\|_\infty \leq C\sqrt{n \log\left(\frac{m}{n} + 2\right)}.$$

As with the [JSS23] result, the failure probability may be boosted using independent repetitions. Our strategy to prove Theorem 8 is the framework in Section 3. We need the following standard fact about the Gaussian measure of the set of Spencer partial colorings, see e.g. Lemma 8.9, [Rot16].

Lemma 21. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ have $\max_{i \in [m], j \in [n]} |\mathbf{A}_{ij}| \leq 1$. Then $\gamma_n(\mathcal{K}) \geq \exp(-\frac{n}{40})$ where*

$$\mathcal{K} := \left\{ x \in \mathbb{R}^n \mid \|\mathbf{A}x\|_\infty \leq \sqrt{8n \log\left(\frac{m}{n} + 2\right)} \right\}. \tag{16}$$

To handle error induced by near-tight constraints, we use a randomized rounding procedure.

Lemma 22. *For $\mathcal{K}$ in (16) and universal constants c, C, suppose we have $x \in C\mathcal{K} \cap [-1, 1]^n$ with*

$$\min(|1 - x_i|, |1 + x_i|) \leq \frac{1}{\sqrt{\log(m)}} \text{ for all } i \in S,$$

where $S \subseteq [n]$ has $|S| \geq cn$. We can compute $x' \in C'\mathcal{K} \cap [-1, 1]^n$ for a universal constant C' with at least $\frac{cn}{3}$ coordinates in $\{-1, 1\}$ in $O(n)$ time, with high probability in n.

Proof. Set $x'_i = x_i$ for $i \notin S$ and, for $i \in S$, sample uniformly at random $x'_i \sim \{1, 2x_i - 1\}$ if $x_i > 0$ and $x'_i \sim \{-1, 2x_i + 1\}$ if $x_i < 0$. Then $\mathbb{E}[x'_i] = x_i$ and by the triangle inequality,

$$\left\| \sum_{i \in [n]} x'_i A_i \right\|_\infty \leq \left\| \sum_{i \in S} (x'_i - x_i) A_i \right\|_\infty + \left\| \sum_{i \in [n]} x_i A_i \right\|_\infty .$$

By a Chernoff bound, at least $\frac{1}{3}$ of the coordinates in S will be rounded to ± 1 with probability $1 - \exp(-\Omega(n))$, and clearly $x' \in [-1, 1]^n$. Moreover, the $x'_i - x_i$ are Rademacher random variables scaled by a factor of at most $\frac{1}{\sqrt{\log(m)}}$, so Hoeffding's inequality yields the claim $x' \in C'\mathcal{K}$. $\square$

Finally, we give an efficient algorithm based on stochastic mirror descent for solving subproblems required by our framework. The next result follows straightforwardly from arguments in [CJST20] (building on [CHW12]), but we summarize how to obtain it for completeness in Appendix B.

Lemma 23. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ have $\leq k$ nonzero entries in every column and $\max_{i \in [m]} \|\mathbf{A}_{i:}\|_2 \leq R$, and let $v \in \mathbb{R}^n$, $\lambda, \varepsilon \geq 0$, and $\delta \in (0, 1)$. There is an algorithm which runs in time*

$$O \left(m + \frac{(k+n)nR^2 \log^2(m) \log(\frac{1}{\delta})}{\varepsilon^2} \right),$$

and returns $x \in [-1, 1]^n$ such that with probability $\geq 1 - \delta$,

$$\|\mathbf{A}x\|_\infty + \lambda \|x - v\|_2^2 \leq \min_{x' \in [-1,1]^n} \|\mathbf{A}x'\|_\infty + \lambda \|x' - v\|_2^2 + \varepsilon.$$

Now we have all the tools to prove Theorem 8.

Proof of Theorem 8. We first state several simplifying reductions from [JSS23]. First, the proof of Theorem 1.1 shows it suffices to prove Theorem 8 when $\frac{n}{\log^2 n} \leq m \leq n^2$; for small m, [ALS21] proves the result, and for large m, a random coloring suffices. Next, the reduction from Theorem 1.1 to Theorem 2.1 in [JSS23] uses [ALS21] to handle columns with less than $n \cdot \log^{-2}(n)$ nonzero entries, so after applying this reduction we may assume $n^2 = O(\text{nnz}(\mathbf{A}) \cdot \log^2(n))$. The reduction from Theorem 2.1 to Theorem 2.2 in [JSS23] shows that it suffices to solve the partial coloring variant of the problem $O(\log \log n)$ many times, i.e. to produce a point in $O(1)\mathcal{K} \cap [-1, 1]^n$ with $\Omega(n)$ tight constraints, where $\mathcal{K}$ is defined as in (16). Finally, the proof of Theorem 2.2 in [ALS21] shows that columns with a large number of nonzero entries may be colored separately, so we can reduce to no column having more than $k = O(\text{nnz}(\mathbf{A}) \cdot \frac{\log(n)}{n})$ nonzero entries.

We now solve the partial coloring problem under the restrictions that the maximum column sparsity is $k = O(\text{nnz}(\mathbf{A}) \cdot \frac{\log(n)}{n})$, and $n^2 = O(\text{nnz}(\mathbf{A}) \cdot \log^2(n))$. We apply Proposition 2 with $\Theta = n$, $\beta = \frac{1}{\sqrt{\log m}}$, and $\rho = C_{\text{set}} \sqrt{8n \log(\frac{m}{n} + 2)}$, where C_{set} is the constant from Proposition 1. Proposition 2 then requires solving $O(\log \log n)$ subproblems of the form

$$\arg \min_{x \in [-1,1]^m} \|\mathbf{A}x\|_\infty + \lambda \|x - g\|_2^2$$

to error $\varepsilon := \Omega(\frac{\sqrt{n}}{\log n})$. We apply Lemma 23 to do so with error probability $O(\log^{-1}(n))$ in time

$$O \left(n^2 + (k+n)n \log^3(n) \log \log(n) \right) = \breve{O} \left(\text{nnz}(\mathbf{A}) \cdot \log^5(n) \right)$$

where we use $R = \sqrt{n}$ and our assumed bounds on k and n^2. A union bound over $\breve{O}(1)$ subproblems (with the overhead of $\breve{O}(1)$ partial coloring phases) gives the failure probability. The partial coloring

32

follows by applying Lemma 21, Proposition 1, Lemma 1, and Lemma 2 to show the output of Proposition 2 has $\Omega(n)$ nearly-tight constraints, and then using Lemma 22 to round these nearly-tight constraints to tight ones. Finally, the runtime follows since there are $\breve{O}(1)$ calls to Lemma 23.

$\square$

Acknowledgements

References

[AAGM15] Shiri Artstein-Avidan, Apostolos Giannopoulos, and Vitali D. Milman. *Asymptotic geometric analysis. Part I*, volume 202 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2015.

[AJJ+22] Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semistreaming 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.

[AL17] Zeyuan Allen-Zhu and Yuanzhi Li. Follow the compressed leader: Faster online learning of eigenvectors and faster MMWU. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 116–125, 2017.

[ALS21] Ryan Alweiss, Yang P. Liu, and Mehtaab Sawhney. Discrepancy minimization via a self-balancing walk. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021*, pages 14–20. ACM, 2021.

[AN19] Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *SIAM J. Comput.*, 48(2):227–248, 2019.

[Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010*, pages 3–10. IEEE Computer Society, 2010.

[BBN13] Michel Baes, Michael Bürgisser, and Arkadi Nemirovski. A randomized mirror-prox method for solving structured large-scale matrix saddle-point problems. *SIAM Journal on Optimization*, 23(2):934–962, 2013.

[BCL94] Keith Ball, Eric A. Carlen, and Elliott H. Lieb. Sharp uniform convexity and smoothness estimates for trace norms. *Inventiones mathematicae*, 115(1):463–482, 1994.

[BDG19] Nikhil Bansal, Daniel Dadush, and Shashwat Garg. An algorithm for Komlós conjecture matching Banaszczyk's bound. *SIAM J. Comput.*, 48(2):534–553, 2019.

[BDGL19] Nikhil Bansal, Daniel Dadush, Shashwat Garg, and Shachar Lovett. The Gram-Schmidt walk: A cure for the Banaszczyk blues. *Theory Comput.*, 15:1–27, 2019.

[BJM23] Nikhil Bansal, Haotian Jiang, and Raghu Meka. Resolving matrix Spencer conjecture up to poly-logarithmic rank. In *STOC '23: 55th Annual ACM SIGACT Symposium on Theory of Computing, 2023*. ACM, 2023.

[BJSS20] Nikhil Bansal, Haotian Jiang, Sahil Singla, and Makrand Sinha. Online vector balancing and geometric discrepancy. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 1139–1152. ACM, 2020.

[BLV22] Nikhil Bansal, Aditi Laddha, and Santosh S. Vempala. A unified approach to discrepancy minimization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022*, volume 245 of *LIPIcs*, pages 1:1–1:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BSS14]   Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan spar-
           sifiers. *SIAM Rev.*, 56(2):315–334, 2014.

[CDST19]   Yair Carmon, John C. Duchi, Aaron Sidford, and Kevin Tian. A rank-1 sketch for
           matrix multiplicative weights. In *Conference on Learning Theory, COLT 2019*, pages
           589–623, 2019.

[CGP+18]   Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junx-
           ing Wang. Graph sparsification, spectral sketches, and faster resistance computation,
           via short cycle decompositions. In *59th IEEE Annual Symposium on Foundations of
           Computer Science, FOCS 2018*, pages 361–372. IEEE Computer Society, 2018.

[Cha01]   Bernard Chazelle. *The Discrepancy Method: Randomness and Complexity.* Cambridge
           University Press, 2001.

[CHW12]   Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for
           machine learning. *J. ACM*, 59(5):23:1–23:49, 2012.

[CJST19]   Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix
           games. In *Advances in Neural Information Processing Systems 32: Annual Conference
           on Neural Information Processing Systems 2019*, pages 11377–11388, 2019.

[CJST20]   Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix
           games. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS
           2020*, pages 283–293. IEEE, 2020.

[CKP+16]   Michael B. Cohen, Jonathan A. Kelner, John Peebles, Richard Peng, Aaron Sidford, and
           Adrian Vladu. Faster algorithms for computing the stationary distribution, simulating
           random walks, and more. In *IEEE 57th Annual Symposium on Foundations of Computer
           Science, FOCS 2016*, pages 583–592. IEEE Computer Society, 2016.

[CKST19]   Charles Carlson, Alexandra Kolla, Nikhil Srivastava, and Luca Trevisan. Optimal lower
           bounds for sketching graph cuts. In *Proceedings of the Thirtieth Annual ACM-SIAM
           Symposium on Discrete Algorithms, SODA 2019*, pages 2565–2569. SIAM, 2019.

[Coh16]   Michael B. Cohen. Ramanujan graphs in polynomial time. In Irit Dinur, editor, *IEEE
           57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 276–
           281. IEEE Computer Society, 2016.

[DJR22]   Daniel Dadush, Haotian Jiang, and Victor Reis. A new framework for matrix discrep-
           ancy: Partial coloring bounds via mirror descent. In *Proceedings of the 54th Annual
           ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 649–658. Asso-
           ciation for Computing Machinery, 2022.

[DNTT18]   Daniel Dadush, Aleksandar Nikolov, Kunal Talwar, and Nicole Tomczak-Jaegermann.
           Balancing vectors in any norm. In *59th IEEE Annual Symposium on Foundations of
           Computer Science, FOCS 2018*, pages 1–10. IEEE Computer Society, 2018.

[DSW22]   Yichuan Deng, Zhao Song, and Omri Weinstein. Discrepancy minimization in input-
           sparsity time. *CoRR*, abs/2210.12468, 2022.

[ES18]   Ronen Eldan and Mohit Singh. Efficient algorithms for discrepancy minimization in
           convex sets. *Random Struct. Algorithms*, 53(2):289–307, 2018.

[GHM15]   Dan Garber, Elad Hazan, and Tengyu Ma. Online learning of eigenvectors. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 560–568, 2015.

[HSS14]   Nicholas J. A. Harvey, Roy Schwartz, and Mohit Singh. Discrepancy without partial colorings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014*, volume 28 of *LIPIcs*, pages 258–273. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014.

[HSSZ19]   Christopher Harshaw, Fredrik Sävje, Daniel A. Spielman, and Peng Zhang. Balancing covariates in randomized experiments using the Gram-Schmidt walk. *CoRR*, abs/1911.03071, 2019.

[JS21]   Arun Jambulapati and Aaron Sidford. Ultrasparse ultrasparsifiers and faster laplacian system solvers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 540–559. SIAM, 2021.

[JSS23]   Vishesh Jain, Ashwin Sah, and Mehtaab Sawhney. Spencer's theorem in nearly input-sparsity time. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 3946–3958. SIAM, 2023.

[JT23]   Arun Jambulapati and Kevin Tian. Revisiting area convexity: Faster box-simplex games and spectrahedral generalizations. *CoRR*, abs/2303.15627, 2023.

[KLOS14]   Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multi-commodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 217–226. SIAM, 2014.

[KMP11]   Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-$m \log n$ time solver for SDD linear systems. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 590–598, 2011.

[KMST10]   Alexandra Kolla, Yury Makarychev, Amin Saberi, and Shang-Hua Teng. Subgraph sparsification and nearly optimal ultrasparsifiers. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 57–66. ACM, 2010.

[KV05]   Adam Tauman Kalai and Santosh S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

[Lar23]   Kasper Green Larsen. Fast discrepancy minimization with hereditary guarantees. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 276–289. SIAM, 2023.

[LM15]   Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.*, 44(5):1573–1582, 2015.

[LS17]   Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 678–687. ACM, 2017.

[LS18]   Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *SIAM J. Comput.*, 47(6):2315–2336, 2018.

[LSY19]   Yang P. Liu, Sushant Sachdeva, and Zejun Yu. Short cycles via low-diameter decompositions. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 2602–2615. SIAM, 2019.

[Mat99]   Jiri Matousek. *Geometric Discrepancy: An Illustrated Guide.* Springer-Verlag, 1999.

[MM15]   Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 1396–1404, 2015.

[Nes07]   Yurii Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming, Series A*, 110:245–259, 2007.

[Pen16]   Richard Peng. Approximate undirected maximum flows in $O(m\text{polylog}(n))$ time. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 1862–1867. SIAM, 2016.

[Pis89]   Gilles Pisier. *The Volume of Convex Bodies and Banach Space Geometry.* Cambridge Tracts in Mathematics. Cambridge University Press, 1989.

[PS22]   Richard Peng and Zhuoqing Song. Sparsified block elimination for directed laplacians. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022*, pages 557–567. ACM, 2022.

[PV23]   Lucas Pesenti and Adrian Vladu. Discrepancy minimization via regularization. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 1734–1758. SIAM, 2023.

[PY19]   Merav Parter and Eylon Yogev. Optimal short cycle decomposition in almost linear time. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132 of *LIPIcs*, pages 89:1–89:14, 2019.

[Rot16]   Thomas Rothvoss. Lecture notes on integer optimization and lattices, 2016.

[Rot17]   Thomas Rothvoss. Constructive discrepancy minimization for convex sets. *SIAM J. Comput.*, 46(1):224–234, 2017.

[RR20]   Victor Reis and Thomas Rothvoss. Linear size sparsifier and the geometry of the operator norm ball. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 2337–2348. SIAM, 2020.

[RR22]   Victor Reis and Thomas Rothvoss. Vector balancing in Lebesgue spaces. *Random Structures and Algorithms*, 2022.

[Spe85]   Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985.

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

[ST83]   Daniel Dominic Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26(3):362–391, 1983.

[ST04]   Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004*, pages 81–90. ACM, 2004.

[SW09]   Daniel A. Spielman and Jaeoh Woo. A note on preconditioning by low-stretch spanning trees. *CoRR*, abs/0903.2816, 2009.

[SW19]   Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2616–2635. SIAM, 2019.

[Tro15]   Joel A. Tropp. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1–2):1–230, 2015.

[ZBG⁺22]   Sepehr Abbasi Zadeh, Nikhil Bansal, Guru Guruganesh, Aleksandar Nikolov, Roy Schwartz, and Mohit Singh. Sticky Brownian rounding and its applications to constraint satisfaction problems. *ACM Trans. Algorithms*, 18(4):33:1–33:50, 2022.

[ZLO15]   Zeyuan Allen Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pages 237–245. ACM, 2015.

A   Proof of Theorem 4

In this section we provide a proof of Theorem 4. As much of the proof is the same, we defer exposition to [RR20] for brevity and only provide technical details here. First we justify that we may assume $\tau \geq 1$. If the result holds for $\tau = 1$, then for any $\tau < 1$ we may consider instead the matrices $\mathbf{A}'_i := \frac{1}{\tau}\mathbf{A}_i$ which still satisfy $\sum_{i=1}^m |\mathbf{A}'_i| \preceq \mathbf{I}_n$ and $\sum_{i=1}^n \mathrm{Tr}(|\mathbf{A}'_i|) = 1$. Then in order to show a measure lower bound for the discrepancy body associated with $\mathbf{A}_i$, it suffices to do so for the (smaller) body associated with $\mathbf{A}'_i$. Hence we will assume $\tau \geq 1$ from now on.

To parallel [RR20], we let

$$\mathbf{S} := \mathcal{A}(\mathbb{1}_m), \quad \widetilde{\mathbf{S}} := \frac{1}{2}\Big(\mathbf{S} + \frac{\tau}{n} \cdot \mathbf{I}_n\Big), \quad \mathbf{A}_{C,D} := (C + D\|x\|_2^2) \cdot \mathbf{I}_n - \mathcal{A}(x). \tag{17}$$

Note that $\widetilde{\mathbf{S}} \succeq \frac{1}{2n}\mathbf{I}_n$ by definition. The potential function we will use is:

$$\Phi_{C,D}(x) := \mathrm{Tr}(\tilde{\mathbf{S}} \cdot \mathbf{A}_{C,D}(x)^{-1}).$$

In order to analyze changes to the potential, we require a generalization of Lemma 11 in [RR20].

Lemma 24. *Let* $\mathbf{A}, \mathbf{B}, \mathbf{S} \in \mathbb{R}^{n \times n}$ *be symmetric with* $\mathbf{A}, \mathbf{S} \succ \mathbf{0}_n$ *and* $\|\delta \mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\|_{\mathrm{op}} \leq \frac{1}{2}$. *Then* $\mathrm{Tr}((\mathbf{A} - \delta\mathbf{B})^{-1}\mathbf{S}) = \mathrm{Tr}(\mathbf{A}^{-1}\mathbf{S}) + \delta\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}) + c\delta^2\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S})$ *for some* $c \in [0, 2]$.

Proof. We abbreviate $\mathbf{M} := \delta\mathbf{A}^{-1}\mathbf{B}$. As $\|\mathbf{M}\|_{\mathrm{op}} \leq \frac{1}{2}$, the matrix $\mathbf{I}_n - \mathbf{M}$ is non-singular. We obtain

$$(\mathbf{A} - \delta\mathbf{B})^{-1} = (\mathbf{A}(\mathbf{I}_n - \mathbf{M}))^{-1} = (\mathbf{I}_n - \mathbf{M})^{-1}\mathbf{A}^{-1} = \sum_{k=0}^{\infty} \mathbf{M}^k \mathbf{A}^{-1},$$

so that

$$\begin{aligned}
\mathrm{Tr}((\mathbf{A} - \delta\mathbf{B})^{-1}\mathbf{S}) &= \sum_{k=0}^{\infty} \mathrm{Tr}(\mathbf{M}^k\mathbf{A}^{-1}\mathbf{S}) = \sum_{k=0}^{2} \mathrm{Tr}(\mathbf{M}^k\mathbf{A}^{-1}\mathbf{S}) + \sum_{k=3}^{\infty} \mathrm{Tr}(\mathbf{M}^k\mathbf{A}^{-1}\mathbf{S}) \\
&= \mathrm{Tr}(\mathbf{A}^{-1}\mathbf{S}) + \delta\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}) + \delta^2\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}) \\
&\quad + \sum_{k=3}^{\infty} \mathrm{Tr}(\mathbf{M}^k\mathbf{A}^{-1}\mathbf{S}).
\end{aligned}$$

We proceed to bound the last sum above. For any $k \geq 3$,

$$\begin{aligned}
|\mathrm{Tr}(\mathbf{M}^k\mathbf{A}^{-1}\mathbf{S})| &= \left|\delta^k \left\langle \left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right)^{k-2}, \mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right\rangle\right| \\
&\leq \left\|(\delta\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}})^{k-2}\right\|_{\mathrm{op}} \cdot \delta^2\mathrm{Tr}\left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right) \\
&\leq \left\|\delta\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right\|_{\mathrm{op}}^{k-2} \cdot \delta^2\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}) \\
&\leq \frac{1}{2^{k-2}}\delta^2\mathrm{Tr}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{S}),
\end{aligned}$$

where the first line used the cyclic property of trace, the second was the matrix Hölder inequality, and the third used $\|\mathbf{A}^k\|_{\mathrm{op}} = \|\mathbf{A}\|_{\mathrm{op}}^k$ for symmetric $\mathbf{A}$. The conclusion then follows from combining the above two displays and applying the triangle inequality. $\qquad\square$

We next give a variant of how the potential changes in a single step (Lemma 12 in [RR20]).

Lemma 25. *Fix $\alpha \in (0, \frac{1}{2})$, $D \leq 1$, and $m = \Omega(\alpha^{-2})$ for a sufficiently large constant. Let $x \in \mathbb{R}^m$ and suppose $\mathbf{A}_{C,D}(x) \succ \mathbf{0}$, $\Phi_{C,D}(x) \leq \frac{Dm^2\alpha^2}{10}$, and $\delta := \frac{\alpha}{45m^5n^2}$. Define $F(y)$ so that*

$$\Phi_{C+\delta^2 F(y),D}(x + \delta y) = \Phi_{C,D}(x). \tag{18}$$

Then there is a $\mathbf{X} \in \mathbb{R}^{m \times m}$ with $\mathbf{0} \preceq \mathbf{X} \preceq \mathbf{I}_m$ and $\mathrm{Tr}(\mathbf{X}) \geq (1-\alpha^2) \cdot m$ so that $\mathbb{E}_{y \sim \mathcal{N}_{\leq m}(\mathbb{0}_m, \mathbf{X})}[F(y)] \leq 0$ while always $|F(y)| \leq 7Dm^4 n$. Further, $\mathbf{A}_{C+\delta^2 F(y),D}(x + \delta y) \succ \mathbf{0}$.

Proof of Lemma 25. To simplify notation, we abbreviate matrices

$$\mathbf{A} := \mathbf{A}_{C,D}(x), \quad \widetilde{\mathbf{B}} := \mathcal{A}(y), \quad \mathbf{B} := \widetilde{\mathbf{B}} - \delta(D\|y\|_2^2 + F(y))\mathbf{I}_n.$$

Next, we define an index set

$$\mathcal{I} := \left\{ i \in [m] \mid \mathrm{Tr}(\mathbf{A}^{-1}\mathbf{A}_i) \leq \frac{1.1}{\alpha^2 m} \cdot \mathrm{Tr}(\mathbf{A}^{-1}\mathbf{S}) \right\}.$$

By Markov's inequality, we have $|\mathcal{I}| \geq (1 - \frac{\alpha^2}{1.1})m$. Consider the subspace

$$H := \Big\{ y \in \mathbb{R}^m \mid y_i = 0 \,\forall i \notin \mathcal{I}, \langle x, y \rangle = 0, \left\langle \mathcal{A}(y), \mathbf{A}^{-1}\widetilde{\mathbf{S}}\mathbf{A}^{-1} \right\rangle = 0, \left\langle \mathcal{A}(y), \mathbf{A}^{-1}\widetilde{\mathbf{S}}\mathbf{A}^{-2} \right\rangle = 0,$$
$$\left\langle \mathcal{A}(y), \mathbf{A}^{-2}\widetilde{\mathbf{S}}\mathbf{A}^{-1} \right\rangle = 0 \Big\}$$

so $\dim(H) \geq |\mathcal{I}| - 4 \geq (1 - \alpha^2)m$. We choose $\mathbf{X}$ as the projection matrix onto H. The remaining proof is organized in two claims: the first bounds $|F(y)|$, and the second controls $\mathbb{E}F(y)$.

Claim 1. *If $y \in H$ and $\|y\|_2 \leq m$, $|F(y)| \leq 7Dm^4 n$, $\|\mathbf{B}\|_{\mathrm{op}} \leq 4m$, and $\|\delta \mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\|_{\mathrm{op}} \leq \frac{1}{2}$.*

Proof. Since (18) holds, monotonicity of the inverse shows the difference matrix

$$\mathbf{A}_{C+\delta^2 F(y),D}(x + \delta y) - \mathbf{A}_{C,D}(x) = \delta^2(D\|y\|_2^2 + F(y)) \cdot \mathbf{I}_n - \delta \mathcal{A}(y)$$

has at least one positive and one negative eigenvalue. Since $\delta \mathcal{A}(y) \succeq -\delta \|y\|_\infty \mathbf{I}_n \succeq -\delta m \mathbf{I}_n$, in order for the difference matrix to have a positive eigenvalue we must have $F(y) \geq -\frac{2m}{\delta}$, and similarly $\delta \mathcal{A}(y) \preceq \delta m \mathbf{I}_n$ implies $F(y) \leq \frac{m}{\delta}$. Hence, $|F(y)| \leq \frac{2m}{\delta}$, which implies the bound:

$$\|\mathbf{B}\|_{\mathrm{op}} \leq \|\mathcal{A}(y)\|_{\mathrm{op}} + \delta\left(Dm + \frac{m}{\delta}\right) \leq 4m. \tag{19}$$

Further, since $\|\widetilde{\mathbf{S}}\mathbf{A}^{-1}\|_{\mathrm{op}} \leq \Phi_{C,D}(x) \leq \frac{Dm^2}{10}$, we have

$$\|\delta \mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\|_{\mathrm{op}} \leq \delta\|\mathbf{A}^{-\frac{1}{2}}\|_{\mathrm{op}}^2\|\mathbf{B}\|_{\mathrm{op}} \leq 4m\delta\left\|\mathbf{A}^{-1}\right\|_{\mathrm{op}}$$
$$\leq 4m\delta\|\widetilde{\mathbf{S}}^{-1}\|_{\mathrm{op}}\|\widetilde{\mathbf{S}}\mathbf{A}^{-1}\|_{\mathrm{op}} \leq \frac{2Dm^3\delta}{5}\|\widetilde{\mathbf{S}}^{-1}\|_{\mathrm{op}} \leq \frac{4Dm^3 n\delta}{5} \leq \frac{1}{2}, \tag{20}$$

where we used our bounds on $\mathbf{B}$ and $\widetilde{\mathbf{S}}\mathbf{A}^{-1}$, and recalled from (17) that $\widetilde{\mathbf{S}} \succeq \frac{1}{2n}\mathbf{I}_n$. It remains to obtain a stronger bound on $|F(y)|$. By using Lemma 24 and the definition of the subspace H,

$$0 = \Phi_{C+\delta^2 F(y),D}(x + \delta y) - \Phi_{C,D}(x) = \mathrm{Tr}\left((\mathbf{A} - \delta\mathbf{B})^{-1}\widetilde{\mathbf{S}}\right) - \mathrm{Tr}\left(\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)$$
$$= \delta\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right) + c\delta^2\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right) \tag{21}$$
$$= \delta^2\left(-\left(D\|y\|_2^2 + F(y)\right)\mathrm{Tr}(\mathbf{A}^{-2}\widetilde{\mathbf{S}}) + c\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)\right),$$

for some $c \in [0, 2]$, since $\mathrm{Tr}(\mathbf{A}^{-1}\mathcal{A}(y)\mathbf{A}^{-1}\widetilde{\mathbf{S}}) = 0$. This implies

$$|F(y)| \le \frac{c\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)}{\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right)} + D\|y\|_2^2 \le 2\left\|\mathbf{A}^{-1}\right\|_{\mathrm{op}}\|\mathbf{B}\|_{\mathrm{op}}^2 + Dm^2 \le 7Dm^4n,$$

where we used the matrix Hölder inequality, (19), and (20) which implies $\|\mathbf{A}^{-1}\|_{\mathrm{op}} \le \frac{Dm^2n}{5}$. $\qquad\square$

Claim 2. *We have*

$$\mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}\left[\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)\right] \le \left(\frac{Dm}{200} + \frac{1.1}{\alpha^2 m}\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{S}\right)\right)\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right).$$

Proof. Let $\lambda := \delta(D\|y\|_2^2 + F(y)) \le 7.7\delta Dm^4n$. By the last two constraints in the definition of H, and using that $\mathbf{B} = \widetilde{\mathbf{B}} - \lambda\mathbf{I}_n$, for any $y \in H$ with $\|y\|_2 \le m$,

$$\left|\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right) - \mathrm{Tr}\left(\mathbf{A}^{-1}\widetilde{\mathbf{B}}\mathbf{A}^{-1}\widetilde{\mathbf{B}}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)\right| = \lambda^2\mathrm{Tr}\left(\mathbf{A}^{-3}\widetilde{\mathbf{S}}\right) \le \lambda^2\left\|\mathbf{A}^{-1}\right\|_{\mathrm{op}}\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right)$$

$$\le 12\delta^2 D^3 m^{10}n^3\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right).$$

Moreover, defining $\mathbf{W}_i := \widetilde{\mathbf{S}}^{\frac{1}{2}}\mathbf{A}^{-1}\mathbf{A}_i\mathbf{A}^{-\frac{1}{2}}$ for all $i \in [n]$,

$$\mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}\left[\mathrm{Tr}\left(\mathbf{A}^{-1}\widetilde{\mathbf{B}}\mathbf{A}^{-1}\widetilde{\mathbf{B}}\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)\right] = \mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}\left[\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}}y_i y_j\langle\mathbf{W}_i, \mathbf{W}_j\rangle\right]$$

$$= \mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}\left[\left\|\sum_{i\in\mathcal{I}}y_i\mathbf{W}_i\right\|_{\mathrm{F}}^2\right] \le \sum_{i\in\mathcal{I}}\|\mathbf{W}_i\|_{\mathrm{F}}^2$$

$$= \sum_{i\in\mathcal{I}}\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)$$

$$\le \sum_{i\in\mathcal{I}}\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{A}_i\mathbf{A}^{-1}\widetilde{\mathbf{S}}\right)\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{A}_i\right)$$

$$\le \frac{1.1}{\alpha^2 m}\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{S}\right)\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right).$$

The second line used Lemma 9 of [RR20] with the fact that when y is truncated, the norm is zero. The fourth line used Lemma 10 of [RR20], and the last used the definition of $\mathcal{I}$ and $\mathcal{A}(\mathbb{1}_m) \preceq \mathbf{I}_n$. The conclusion follows by combining the above two displays with our choice of δ. $\qquad\square$

Taking expectations over (21) and applying Claim 2 shows

$$0 \le \left(-0.44Dm - \mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}[F(y)] + \frac{2.2}{\alpha^2 m}\mathrm{Tr}\left(\mathbf{A}^{-1}\mathbf{S}\right)\right)\mathrm{Tr}\left(\mathbf{A}^{-2}\widetilde{\mathbf{S}}\right),$$

where we used $\dim(H) \ge \frac{1}{2}m$ and Corollary 5 of [RR20] to lower bound $\mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}[\|y\|_2^2] \ge 0.45Dm$. Rearranging, and using $\mathbf{S} \preceq 2\widetilde{\mathbf{S}}$ and our potential bound, then yields

$$\mathbb{E}_{y\sim\mathcal{N}_{\le m}(\mathbb{0}_m,\mathbf{X})}[F(y)] \le \frac{4.4}{\alpha^2 m}\mathrm{Tr}(\mathbf{A}^{-1}\widetilde{\mathbf{S}}) - 0.44Dm \le 0.$$

Finally, note that since $\mathrm{Tr}(\mathbf{A}^{-1}\widetilde{\mathbf{S}}) \le \frac{Dm^2\alpha^2}{10}$ and $\widetilde{\mathbf{S}} \succeq \frac{1}{2n}\mathbf{I}_n$, no eigenvalue of $\mathbf{A}$ can be smaller than $\frac{5\tau}{Dm^2\alpha^2n}$. Hence, the last conclusion on positive semidefiniteness follows from

$$\mathbf{A}_{C+\delta^2 F(y),D}(x + \delta y) - \mathbf{A} = \delta^2\left(D\|y\|_2^2 + F(y)\right)\mathbf{I}_n - \delta\mathcal{A}(y) \succeq -3m\delta\mathbf{I}_n \succ -\frac{1}{2n}\mathbf{I}_n. \qquad\square$$

Theorem 4. *Suppose $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}^n_{\succeq\mathbf{0}}$ satisfy $\mathcal{A}(\mathbb{1}_m) \preceq \mathbf{I}_n$ and $\mathrm{Tr}(\mathcal{A}(\mathbb{1}_m)) \leq \tau$ for sufficiently large m and $n \leq 2^{\frac{m}{5}}$. Following notation (2), there is a constant C_0 such that*

$$\gamma_m\left(\frac{C_0}{\alpha}\mathcal{K}_{\sqrt{\frac{\tau}{m}},\mathcal{A}} + \alpha\sqrt{m}\mathbb{B}_2^m\right) \geq \frac{1}{2}, \text{ for all } \alpha \in (0,1).$$

Proof. Assume the bounds in Lemma 25 are met, and define $\varepsilon := \sqrt{\frac{\tau}{m}}$. We overload

$$\mathbf{A}_i \leftarrow \begin{pmatrix} \mathbf{A}_i & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_i \end{pmatrix} \in \mathbb{R}^{2n\times 2n} \text{ for all } i \in [n].$$

We run a hypothetical algorithm for $T := \frac{1}{\delta^2}$ iterations, with parameter choices

$$C := \frac{10\varepsilon}{\alpha}, \ D := \frac{2\varepsilon}{\alpha m}, \ \delta := \frac{\alpha}{45m^5n^2}.$$

Our algorithm initalizes $x^{(0)} \leftarrow \mathbb{0}_m$ and for $t \in [T]$, lets $\mathbf{X}^{(t)}$ be the covariance matrix given by Lemma 25 applied to $x^{(t-1)}$, samples $y^{(t)} \sim \mathcal{N}(\mathbb{0}_m, \mathbf{X}^{(t)})$ and $z^{(t)} \sim \mathcal{N}(\mathbb{0}_m, \mathbf{I}_m - \mathbf{X}^{(t)})$, and updates $x^{(t)} \leftarrow x^{(t-1)} + \delta y^{(t)}$. Analagously to the proof of Theorem 1 in [RR20], Azuma's inequality and Gaussian concentration show that all of the following hold with probability at least $\frac{1}{2}$.

1. $\left\|y^{(t)}\right\|_2 \leq m$ for all $t \in [T]$.

2. $\delta^2 \sum_{t\in[T]} F(y^{(t)}) \leq \frac{C}{10}$.

3. $\|Y\|_2^2 \leq 5m$ and $\|Z\|_2^2 \leq 5\alpha^2 m$, where $Y := \sum_{t\in[T]} y^{(t)}$ and $Z := \sum_{t\in[T]} z^{(t)}$.

We remark that the first claim above results in the requirement that $n \leq 2^{\frac{m}{5}}$, since we are union bounding over $n^4\mathrm{poly}(m)$ steps and Gaussian concentration fails with probability 2^{-m} in each step by Corollary 5 of [RR20]. Further, the initial value of the potential function is $2C^{-1}\tau = \frac{Dm^2\alpha^2}{10}$, so the potential bound in Lemma 25 is met if we update C by $\delta^2 F(y^{(t)})$ each iteration. Under these events, since $Y + Z$ is a draw from $\mathcal{N}(\mathbb{0}_m, \mathbf{I}_m)$, we have shown

$$\|\mathcal{A}(Y)\|_{\mathrm{op}} \leq (1.1C + D\|Y\|_2^2) \leq \frac{21\varepsilon}{\alpha}.$$

Hence, at least half of draws from $\mathcal{N}(\mathbb{0}_m, \mathbf{I}_m)$ are within distance $\alpha\sqrt{5m}$ from $\frac{21}{\alpha}\mathcal{K}_{\varepsilon,\mathcal{A}}$. Reparameterizing α by a constant and adjusting C_0 then yields the claim, up to the restriction $m = \Omega(\alpha^{-2})$ in Lemma 25. If instead $\alpha \geq \frac{C_0}{\sqrt{2m}}$, since $\|\mathcal{A}(x)\|_{\mathrm{op}}^2 \leq \|\mathcal{A}(x)\|_F^2 \leq 2\tau$ with probability $\geq \frac{1}{2}$ by Markov's inequality, $\frac{C_0}{\alpha}\mathcal{K}_{\varepsilon,\mathcal{A}}$ already covers the Gaussian measure without the addition of $\alpha\sqrt{m}\mathbb{B}_2^m$. $\square$

B Optimization subroutines

B.1 Discussion of Proposition 5

In this section, we give a discussion of how to modify the algorithm in [JT23] to obtain Proposition 5. In particular, Proposition 5 follows from modifying Theorem 3 of [JT23] applied to the regularized box-spectraplex primal-dual formulation of (9), whose notation we will follow throughout this section. We observe that $\mathcal{T}_{\mathrm{mv}}(\widetilde{\mathbf{A}}_i) = \mathcal{T}_{\mathrm{mv}}(|\widetilde{\mathbf{A}}_i|) = \mathcal{T}_{\mathrm{mv}}(\mathbf{A}_i)$, since we assumed all the $\mathbf{A}_i \in \mathbb{S}^d_{\succeq\mathbf{0}}$, and there is no "$\mathbf{B}$" term in the notation of [JT23], Theorem 3. Further,

$$\left\|\sum_{i\in[m]} \widetilde{\mathbf{A}}_i\right\|_{\mathrm{op}} = \left\|\sum_{i\in[m]} \mathbf{A}_i\right\|_{\mathrm{op}} = \|\mathcal{A}(\mathbb{1}_m)\|_{\mathrm{op}},$$

so in the notation of [JT23], Theorem 3, it suffices to set $L_{\text{tot}} = L_{\mathcal{A}} = \|\mathcal{A}(\mathbb{1}_m)\|_{\text{op}}$. It remains how to handle the regularization term, denoted in this section by

$$q(x) := \lambda \|x - g\|_2^2.$$

It is known in the literature how to modify extragradient methods to handle composite terms (e.g. Section 5.2, [CJST19]), but because the algorithm in [JT23] is somewhat nonstandard, we give a brief description here. As seen in the proofs of Lemma 4, Corollary 2 of [JT23], it suffices to modify the left-hand sides of Eq. (10) in [JT23] to read for iterates $z = (x, \mathbf{Y})$, $z' = (x', \mathbf{Y}')$,

$$\eta \left\langle g(z) + \nabla q(x'), z' - z^+ \right\rangle \leq V_z^{(\alpha+\beta)}(z^+) - V_{z'}^{(\alpha)}(z^+) - V_z^{(\alpha)}(z'),$$

$$\eta \left\langle g(z') + \nabla q(x'), z^+ - u \right\rangle \leq 2V_z^{(\alpha+\beta)}(u) - 2V_{z^+}^{(\alpha+\beta)}(u) - 2V_z^{(\alpha+\beta)}(z^+)$$
$$+ 2V_{\overline{\mathbf{Y}}}^{\gamma h}(u^{\mathsf{y}}) - 2V_{\overline{\mathbf{Y}}^+}^{\gamma h}(u^{\mathsf{y}}),$$

where u^{y} is the spectraplex component of $u \in [-1, 1]^m \times \Delta^{2n \times 2n}$, and $\overline{\mathbf{Y}}, \overline{\mathbf{Y}}^+$ are auxiliary iterates maintained by the extragradient step oracles of [JT23]. Achieving the second equation above is immediate by giving the extragradient step oracle (Algorithm 5) in [JT23] the linear term $g(z') + \nabla q(x')$, instead of just $g(z')$. To achieve the first equation, we modify the subproblems solved in Lines 3 and 5 of the gradient step oracle (Algorithm 4) in [JT23] to minimize a linear term, plus $r(\cdot, \mathbf{Y})$, plus ηq. It is simple to check that first-order optimality conditions on x' in Corollary 3 then result in the additional $\nabla q(x')$ term in the first line above, and no other proofs are changed.

B.2   Proof of Lemma 23

Lemma 23. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ have $\leq k$ nonzero entries in every column and $\max_{i \in [m]} \|\mathbf{A}_{i:}\|_2 \leq R$, and let $v \in \mathbb{R}^n$, $\lambda, \varepsilon \geq 0$, and $\delta \in (0, 1)$. There is an algorithm which runs in time*

$$O\left(m + \frac{(k+n)nR^2 \log^2(m) \log(\frac{1}{\delta})}{\varepsilon^2}\right),$$

and returns $x \in [-1, 1]^n$ such that with probability $\geq 1 - \delta$,

$$\|\mathbf{A}x\|_\infty + \lambda \|x - v\|_2^2 \leq \min_{x' \in [-1,1]^n} \|\mathbf{A}x'\|_\infty + \lambda \|x' - v\|_2^2 + \varepsilon.$$

Proof. It suffices to prove that in time $O((k+n)n^2 \log^2(m)\varepsilon^{-2})$, we can return a point satisfying

$$\mathbb{E}\left[\|\mathbf{A}x\|_\infty + \lambda \|x - v\|_2^2\right] \leq \min_{x' \in [-1,1]^n} \|\mathbf{A}x'\|_\infty + \lambda \|x' - v\|_2^2 + \frac{\varepsilon}{2},$$

since Markov's inequality results in an ε-suboptimal point with probability $\geq \frac{1}{2}$, and then we can run $O(\log \frac{1}{\delta})$ independent runs and take the best function value. To do this, we reparameterize $\mathbf{A}$ as the vertical concatenation of $\sqrt{n}\mathbf{A}$ and $-\sqrt{n}\mathbf{A}$, and recast the objective as the minimax problem

$$\min_{x \in [-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]^n} \max_{y \in \Delta^m} y^\top \mathbf{A}x + \lambda \|\sqrt{n}x - v\|_2^2. \tag{22}$$

It then suffices to apply Proposition 2 of [CJST20], following the ℓ_2-ℓ_1 local norm setup in Table 6, with the composite extension in Lemma 13 (setting $Q(x, y) = \lambda \|x - v\|_2^2$). Specifically, Proposition 2 of [CJST20] requires a local gradient estimator $\tilde{g}$ (see Definition 3 of that paper) for the operator

$$g(x, y) = \left(\mathbf{A}^\top y, -\mathbf{A}x\right)$$

corresponding to the bilinear component of our minimax problem (22). We use

$$\tilde{g}(x,y) = \left(\mathbf{A}_{i:}, -\mathbf{A}_{:j} \cdot \frac{x_j}{p_j} \right),$$

where $i \in [m]$ is randomly selected with probability y_i, and $j \in [n]$ is randomly selected with probability $p_j = x_j^2 \|x\|_2^{-2}$. This estimator $\tilde{g}$ is unbiased for g, meeting the first criterion of Definition 3 in [CJST20]. Further, since all rows of $\mathbf{A}$ have ℓ_2 norm bounded by $R\sqrt{n}$ (after rescaling by $\sqrt{n}$), Lemma 21 of [CJST20] with w_0 set to the all-zeroes vector shows $\tilde{g}$ satisfies the second criterion of Definition 3 in [CJST20] with $L = O(R\sqrt{n})$. Finally, for the ℓ_2-ℓ_1 local norm setup, Table 6 in [CJST20] shows $\Theta = O(\log m)$. In conclusion, Proposition 2 of [CJST20] shows that after

$$O\left(\frac{L^2 \Theta}{\varepsilon^2} \right) = O\left(\frac{nR^2 \log(m)}{\varepsilon^2} \right)$$

iterations, we obtain expected suboptimality gap $\frac{\varepsilon}{2}$ as desired. In each iteration, we can explicitly update the vector x and recompute the sampling distribution proportional to x^2 in $O(n)$ time. Because the update to y is k-sparse, Section 5.1 of [CJST20] (see also Lemma 5.4 of [JSS23] for a brief summary) shows how to update y and maintain a sampling distribution proportional to it in time $O(k \log(m))$ per iteration. Hence, each iteration takes time $O(n + k \log(m))$ as desired. $\qquad\square$

C Low-distortion subgraph construction

In this section, we give a proof of the low-distortion subgraph construction claimed in Proposition 9.

Proposition 9 ([JS21])**.** *Let $G = (V, E, w_G)$ have $m = |E|$ and $n = |V|$. There is an algorithm which takes a parameter $\gamma > 1$ and runs in time $O(m)$, which produces a $O(m\gamma^{o(1)}(\log n)^{o(1)})$-distortion subgraph of G with $n - 1 + \frac{m}{\gamma}$ edges.*

Proof. This construction is implicit in the proof of Theorem A.4 in [JS21]: we summarize the details here for completeness. Theorem 1.9 of [JS21] gives a "path sparsification" algorithm, which takes as input an n-node m-edge graph G and parameter $q \geq 1$ and returns a $(q, O(\log^5 n))$-path sparsifier with $O(nq \log^3 n)$ edges in $O(m + nq \log^{13} n)$ time.

In addition, Theorem 2.5 of [JS21] provides an algorithm which takes as input a graph G, parameters $k, \gamma \geq 2$, and a path sparsification algorithm which runs on n'-vertex m'-edge graphs and returns a $(10\beta, \beta)$-path sparsifier (for some value β) with $S(m', n')$ edges in $T(m', n')$ time. It outputs a subgraph H with at most

$$n - 1 + O\left(\frac{m}{\gamma} + \frac{m \log \gamma}{k^2} \right) + S\left(O(m), O\left(\frac{m}{k} \right) \right)$$

edges such that $\mathrm{Tr}(\mathbf{L}_H^\dagger \mathbf{L}_G) \leq m\hat{\alpha}$ for

$$\hat{\alpha} = O\left(\exp\left(\sqrt{8 \log \gamma \cdot \log\left(48 \log k \sqrt{\log \gamma} \right)} \right) \log k \sqrt{\log \gamma} \right).$$

Additionally, the algorithm runs in $O(m + T(O(m), O(\frac{m}{k})))$ time.

Choosing $q = O(\log^5 n')$ in Theorem 1.9 gives an algorithm which runs on n'-vertex m'-edge graphs, returns a $(10\beta, \beta)$-path sparsifier for $\beta = O(\log^5 n')$, and has $S(m', n') = O(n' \log^8 n')$ and

$T(m', n') = O(m' + n' \log^{18} n')$. Choosing $k = \gamma \log^{18} m$ in Theorem 2.5 and combining the above gives an algorithm which takes in G and outputs a subgraph H in time

$$O\left(m + T\left(O(m), O\left(\frac{m}{k}\right)\right)\right) = O\left(m + \frac{m}{k} \log^{18} m\right) = O(m)$$

with

$$n - 1 + O\left(\frac{m}{\gamma} + \frac{m \log \gamma}{k^2}\right) + S\left(O(m), O\left(\frac{m}{k}\right)\right) = n - 1 + O\left(\frac{m}{\gamma}\right) + \frac{m}{k} \log^8 m$$

$$= n - 1 + O\left(\frac{m}{\gamma}\right)$$

edges, such that $\mathrm{Tr}(\mathbf{L}_H^\dagger \mathbf{L}_G) \le m\hat{\alpha}$ with

$$\hat{\alpha} = O\left(\exp\left(\sqrt{8 \log \gamma \cdot \log\left(48 \log\left(\gamma \log^{18} n\right) \sqrt{\log \gamma}\right)}\right) \log\left(\gamma \log^{18} n\right) \sqrt{\log \gamma}\right). \qquad (23)$$

To simplify the above expression, note that

$$\log\left(\gamma \log^{18} n\right) \sqrt{\log \gamma} \le 18\sqrt{\log \gamma} \log \log n + (\log \gamma)^{\frac{3}{2}} \le 20(\log \log n)^{\frac{3}{2}} + 20(\log \gamma)^{\frac{3}{2}}.$$

Since $\log(a + b) \le 2 \max\{\log a, \log b\}$ for $a, b \ge 2$, we have

$$8 \log \gamma \cdot \log\left(48 \log\left(\gamma \log^{18} n\right) \sqrt{\log \gamma}\right) \le 16 \log \gamma \cdot \left(\log 1000 + \frac{3}{2} \max\left(\log \log \gamma, \log \log \log n\right)\right)$$

$$\le 400 + 36 \max\left(\log \gamma \cdot \log \log \gamma, \log \gamma \cdot \log \log \log n\right)$$

$$\le 400 + 36 \max\left(\log \gamma \cdot \log \log \gamma, \log \log n \cdot \log \log \log n\right)$$

where the last inequality holds since $\log \log \log n \ge \log \log \gamma$ only when $\gamma \le \log n$. Substituting the above bounds into (23) and using $\sqrt{a + b} \le \sqrt{a} + \sqrt{b}$, we have

$$\hat{\alpha} = O\left(\exp\left(20 + 6 \max\left(\sqrt{\log \gamma \cdot \log \log \gamma}, \sqrt{\log \log n \cdot \log \log \log n}\right)\right)\left((\log \log n)^{\frac{3}{2}} + (\log \gamma)^{\frac{3}{2}}\right)\right)$$

$$= \breve{O}\left(\exp(6\sqrt{\log \gamma \cdot \log \log \gamma}) \exp(6\sqrt{\log \log n \cdot \log \log \log n})(\log \gamma)^{\frac{3}{2}}\right) = O(\gamma^{o(1)} (\log n)^{o(1)}).$$

Thus the algorithm described above proves the desired claim. $\qquad \square$