

Faster MAX-CUT on Bounded Threshold Rank Graphs

Prashanti Anderson^{*} Samuel B. Hopkins[†] Amit Rajaraman[‡] David Steurer[§]

November 14, 2025

Abstract

We design new algorithms for approximating 2CSPs on graphs with bounded threshold rank, that is, whose normalized adjacency matrix has few eigenvalues larger than ε , smaller than $-\varepsilon$, or both. Unlike on worst-case graphs, 2CSPs on bounded threshold rank graphs can be $(1 + O(\varepsilon))$ -approximated efficiently. Prior approximation algorithms for this problem run in time exponential in the threshold rank and $1/\varepsilon$. Our algorithm has running time which is *polynomial* in $1/\varepsilon$ and exponential in the threshold rank of the *label-extended* graph, and near-linear in the input size. As a consequence, we obtain the first $(1 + O(\varepsilon))$ approximation for MAX-CUT on bounded threshold rank graphs running in $\text{poly}(1/\varepsilon)$ time. We also improve the state-of-the-art running time for 2CSPs on bounded threshold rank graphs from polynomial in input size to near-linear via a new comparison inequality between the threshold rank of the label-extended graph and base graph. Our algorithm is a simple yet novel combination of subspace enumeration and semidefinite programming.

^{*}MIT EECS and CSAIL. Email: paanders@mit.edu. Supported by NSF award no. 2238080.

[†]MIT EECS and CSAIL. Email: samhop@mit.edu. Supported by NSF award no. 2238080, MLA@CSAIL, Fin-TechAI@CSAIL, MIT Research Support Committee, MIT-Google Program for Computing Innovation.

[‡]MIT EECS and CSAIL. Email: amit_r@mit.edu. Supported by a MathWorks Fellowship.

[§]ETH Zurich. Email: dsteurer@inf.ethz.ch.

1 Introduction

Constraint satisfaction problems (CSPs) lie at the heart of algorithms, discrete optimization, computational complexity, and more. Classic examples like 3-SAT and MAX-CUT appeared in Karp’s original list of 21 NP-complete problems; the study of both algorithms and computational complexity of CSPs for the last 50 years has been inextricably linked to breakthroughs like the PCP theorem, constructions of error-correcting codes, algorithm design techniques like linear and semidefinite programming, the dichotomy theorem, and the unique games theory, to name a few.

One fruitful avenue for algorithm development in the face of NP-hardness of general CSPs has been the design of algorithms which perform well on CSPs whose constraints form a graph or hypergraph with some beyond-worst-case structure. Early examples in this spirit are dynamic programming algorithms for CSPs on bounded tree-width graphs [Kum92], message-passing algorithms inspired by statistical physics for random CSPs [BMZ05], and greedy algorithms for dense CSPs [AKK95, MS08].

A more recent line of work focuses on CSPs whose constraint graphs exhibit spectral structure—one or two-sided expansion, or more generally, having low *threshold rank*, that is, having few large eigenvalues. This generalizes the dense graph setting, and so far led to the subexponential algorithm for unique games [ABS15], efficient algorithms for (list-)decoding high-rate error-correcting codes [JQT20, JST21], new approximate counting and sampling algorithms [Ris16, JKR19], new regularity lemmas [OGT13], and more generally the insight that CSPs are computationally tractable when their constraint graphs exhibit “local to global” phenomena [BRS11]. These works focus on 2CSPs to make things conceptually simple; [AJT19] shows that insights from 2CSPs transfer to higher-arity CSPs. The high-level questions addressed by these works and ours are:

*For which MAX-2CSP instances can efficient algorithms find close-to-optimal assignments?
What are the fastest algorithms to do so?*

To go further we need some notation. To an instance ϕ of an n -variable 2CSP we associate an n -vertex undirected graph G with an edge $i \sim j$ in G iff variables i and j participate in a constraint together in ϕ . Let A be the normalized adjacency matrix of G , with eigenvalues in $[-1, 1]$. The ε threshold rank of A , denoted $\text{rank}_{\geq \varepsilon}(A)$, is the number of eigenvalues of value at least ε . We discuss some variants of threshold rank below.

Our Contributions. Our main contribution is a very simple but powerful approach to minimization of quadratic forms $x^\top Ax$ over the Boolean hypercube, solid hypercube, and more generally over arbitrary convex subsets of \mathbb{R}^n admitting an efficient separation oracle, whose running time is parameterized by the number of negative eigenvalues of A . Note that if $A \succeq 0$, then minimization of $x^\top Ax$ is a convex problem which can be solved in polynomial time, so running time parameterized by the number of negative eigenvalues of A is sensible.

We apply this approach to obtain two main algorithmic results for 2CSPs on low threshold rank constraint graphs. The first result concerns perhaps the simplest 2CSP: MAX-CUT. MAX-CUT on low threshold rank graphs was studied in [BRS11, OGT13], and the special case of dense graphs was extensively studied in [AKK95, FDLV96, YZ14, FLP15, MS08].

Our first main result is a $(1 + O(\varepsilon))$ approximation algorithm for MAX-CUT on n -vertex, m -edge graphs which runs in time $(1/\varepsilon)^{O(\text{rank}_{\leq -\varepsilon})} \cdot \tilde{O}(m + n)$, where $\text{rank}_{\leq -\varepsilon}$ is the number of eigenvalues of the normalized adjacency matrix less than $-\varepsilon$. No prior work for MAX-CUT on graphs satisfying any eigenvalue-based condition obtains a $(1 + O(\varepsilon))$ approximation in $\text{poly}(1/\varepsilon)$ time; prior algorithms always incur an exponential dependence on $1/\varepsilon$.

Our second main result is a new $(1 + O(\varepsilon))$ approximation algorithm for general 2CSPs with finite alphabet $[q]$, where $q \in \mathbb{N}$, with running time $(1/\varepsilon)^{O(\text{rank}_{\geq \varepsilon^2}/\varepsilon^4)} \cdot \tilde{O}(m + n)$, where n is the number of variables and m is the number of constraints/clauses. The previous state-of-the-art algorithm by Barak–Raghavendra–Steurer [BRS11] obtains the same guarantee only in time $\exp(\tilde{O}(\text{rank}_{\geq \varepsilon^2}/\varepsilon^4)) \cdot (mn)^{O(1)}$, relying on the sum-of-squares hierarchy. Thus, our algorithm is the first for general 2CSPs on bounded threshold rank graphs to obtain near-linear running time; indeed, this is true even in the setting of expanders, where $\text{rank}_{\geq \varepsilon^2} = 1$.¹ We discuss below some interpretation of the parameter $\text{rank}_{\geq \varepsilon^2}$, via higher-order Cheeger inequalities [LGT14].

Our Techniques, in Brief Our algorithms combine two key ingredients: subspace enumeration and semidefinite programming (SDP). Both have been used extensively in algorithms for CSPs, but the way we combine them is novel. Prior subspace enumeration algorithms were limited to nearly-satisfiable instances, and prior SDP-based algorithms (a) did not achieve $\text{poly}(1/\varepsilon)$ running time for MAX-CUT and (b) did not achieve near-linear running time for $(1 + O(\varepsilon))$ approximation with respect to m and n . Unlike prior algorithms for CSPs based on SDPs, ours solves a *series* of very simple SDPs, one for each element of a discretization of a subspace constructed from the underlying CSP.

We view the simplicity of our algorithm and analysis as a contribution in itself. The entire algorithm and analysis for a variant running in $n^{O(1)}$ time rather than $\tilde{O}(n + m)$ time spans only a few pages, and the running time can be improved to $\tilde{O}(n + m)$ by appeal to standard matrix multiplicative weights techniques [Ste10a]. We manage to avoid the use of higher levels of convex hierarchies such as sum-of-squares.

Our analysis relies on a new inequality relating the threshold rank of G to the threshold rank of the so-called *label-extended* graph of a CSP instance ϕ . We defer the formal definition, but in brief, the label-extended graph for a CSP with n variables and alphabet size q is a graph with vertex set $[n] \times [q]$, where vertex (i, a) represents “variable i is set to value a ”. Prior works relating the threshold rank of the label-extended graph to the threshold rank of G either apply only to restricted classes of CSPs or lose polynomial factors in n ; ours incurs quantitative losses depending only on the threshold value ε and the alphabet size q . Our proof of this inequality is actually a simple generalization of an argument of [BRS11], recently reinterpreted in a closely-related setting by [BHSV25].

¹ We discuss below the relationship between our work and algorithms for restricted classes of 2CSPs based on regularity lemmas. In short, while near-linear-time regularity decompositions exist in prior work and give a plausible alternative route to a near-linear time algorithm for general 2CSPs on low threshold rank graphs, no prior work carries this out.

1.1 Results

Quadratic Optimization over the Hypercube. Our first result is an algorithm for quadratic optimization over the hypercube when the coefficient matrix A has few negative eigenvalues.

Theorem 1.1 (Quadratic optimization). *Let $A \in \mathbb{R}^{n \times n}$ be an arbitrary symmetric matrix, scaled to have $\|A\|_{\text{op}} \leq 1$, and let $\text{OPT} = \max_{x \in \{\pm 1\}^n} x^\top A x$. Set $k = \text{rank}_{\geq \varepsilon}(A)$. Then, there is an algorithm running in time $\text{poly}(n) \cdot \left(\frac{1}{\varepsilon}\right)^{O(k)}$ that outputs a (random) vector $\hat{x} \in \{\pm 1\}^n$ such that*

$$\mathbb{E} \hat{x}^\top A \hat{x} \geq \text{OPT} - O(\varepsilon n).$$

Remark 1.2. While we state [Theorem 1.1](#) for the case of optimization over $\{\pm 1\}$, our main insights can be generalized to optimizing $x^\top A x$ over convex domains with efficient separation oracles when A has few negative eigenvalues.

Note that if A has no positive eigenvalues and has a 0 diagonal, then $\max_{x \in \{\pm 1\}^n} x^\top A x$ can be performed via maximization of a concave function over a convex set, since it suffices to optimize over x in the solid cube $[-1, 1]^n$ and round the result greedily. Our result can be viewed as showing that even if the optimization problem $\max_{x \in [-1, 1]^n} x^\top A x$ is not convex, but the non-convexity is restricted to a small-dimensional subspace, then we can still efficiently find an \hat{x} which is close to optimal. Intuitively, this is because one can perform limited brute-force search over the span of the positive eigenvectors of A , and for each point in the span solve an associated *convex* optimization problem.

MAX-CUT on Low Threshold Rank Graphs. One special case of the quadratic optimization problem is when A is the negation of the normalized adjacency matrix of a regular graph. In this case, we have the MAX-CUT problem. In fact, the ideas in [Theorem 1.1](#) also extend to irregular graphs, and the algorithm can be sped up using near-linear-time SDP solvers [\[Ste10a\]](#), to obtain the following.

Theorem 1.3 ($(1 + O(\varepsilon))$ Approximation for MAX-CUT in $\text{poly}(1/\varepsilon)$ Time). *Let G be a graph on vertex set $[n]$ with m edges, with adjacency matrix A and diagonal degree matrix D . Let OPT be the size of the largest cut in G . Set $k = \text{rank}_{\leq -\varepsilon}(D^{-1/2} A D^{-1/2})$. In time $\left(\frac{1}{\varepsilon}\right)^{O(k)} \cdot \tilde{O}(n + m)$, one can output a (random) cut attaining value $(1 - O(\varepsilon)) \cdot \text{OPT}$ in expectation.*

We now compare [Theorem 1.3](#) to prior work. The problem of MAX-CUT on low threshold rank graphs has been studied in several prior works, with various runtimes and notions of threshold rank. These are summarized in the table below. Note that all parameters k are defined with respect to the normalized adjacency matrix. The algorithms discussed in the table all produce $(1 + O(\varepsilon))$ approximations to the maximum cut.

² [\[OGT13\]](#) provides the running time $2^{\tilde{O}(k^{1.5}/\varepsilon^3)} + \text{poly}(n)$ via a regularity decomposition of the graph. Using matrix multiplicative weights to find the regularity decomposition via a fast SDP solver yields the running time we quote. Matrix multiplicative weights-based SDP solvers appear first in [\[AK16\]](#); the variant for regularity decompositions appears in [\[QST20\]](#).

Table 1: MAX-CUT on Low Threshold Rank Graphs

Paper	Runtime	Parameter k
[BRS11]	$2^{O(k/\varepsilon^4)} \cdot n^{O(1)}$	# of eigs in $[\varepsilon^2, 1]$
[GS11]	$n^{O(k/\varepsilon^2)}$	# of eigs $\leq -\varepsilon$
[GS12]	$2^{k/\varepsilon^3} \cdot n^{O(1/\varepsilon)}$	# of eigs $\leq -\varepsilon^2$
[OGT13] + [JST21] ²	$2^{\tilde{O}(k^{1.5}/\varepsilon^3)} + \tilde{O}(n+m)$	sum of squares of eigs not in the range $[-\varepsilon, \varepsilon]$
This work	$(\frac{1}{\varepsilon})^{O(k)} \cdot \tilde{O}(n+m)$	# of eigs $\leq -\varepsilon$

Similar to [GS11, GS12], our runtime is parameterized by the number of eigenvalues of the (normalized) adjacency matrix which are smaller than $-\text{poly}(\varepsilon)$. However, compared to [GS11, GS12], our running time has an exponentially better dependence on $1/\varepsilon$. Regularity lemma-based algorithms for MAX-CUT on low threshold rank graphs such as [OGT13] are instead parameterized with k being as the sum of the squares of eigenvalues not in the range $[-\varepsilon, \varepsilon]$. Since all the eigenvalues of the normalized adjacency matrix lie in the range $[-1, 1]$, this could be much smaller than the number of such eigenvalues. However, the sum of the squares of the eigenvalues not in the range $[-\varepsilon, \varepsilon]$ is at least $O(\varepsilon^2) \cdot (\# \text{ of eigenvalues not in the range } [-\varepsilon, \varepsilon])$; we also require only control on the negative eigenvalues of A rather than positive and negative eigenvalues. As a result, our exponential dependence on k and ε also improve on regularity lemma-based approaches [OGT13, JST21], although we incur an multiplicative dependence in runtime on the $\tilde{O}(n+m)$, $(\frac{1}{\varepsilon})^{O(k)}$ terms, as opposed to an additive dependence.

General 2CSPs. In addition to fast algorithms for MAX-CUT, we give the first near-linear time algorithm for general CSPs on low threshold rank graphs.

Theorem 1.4. *Let G be a graph on $[n]$ with m edges, with adjacency matrix A and degree diagonal D . For each edge ij in G , let $\varphi_{ij} : [q]^2 \rightarrow \{0, 1\}$ be a predicate on alphabet size q that is not identically 0. Also let $\text{OPT} = \max_{x \in [q]^n} \Phi(x) := \max_{x \in [q]^n} \sum_{ij \in E(G)} \varphi_{ij}(x_i, x_j)$. Set $k = \text{rank}_{\geq \varepsilon^2}(D^{-1/2}AD^{-1/2})$. In time $(\frac{1}{\varepsilon})^{O(k/\varepsilon^4)} \cdot \tilde{O}(n+m) \cdot \text{poly}(q)$, one can find a random assignment \hat{x} with $\mathbb{E}\Phi(\hat{x}) \geq \text{OPT} - O(\varepsilon q m)$.*

The main prior work that gave algorithms for general CSPs under threshold rank conditions is [BRS11]. Our work matches the exponential dependence on k, ε (up to factors of $\log(1/\varepsilon)$) of [BRS11] while improving the polynomial dependence on n and m to near-linear. The main barrier to achieving a near-linear running time via prior algorithmic techniques was that the algorithmic approaches that lead to $\tilde{O}(n+m)$ time (e.g., regularity lemmas [OGT13, JST21]) also lead to exponential dependence on the threshold rank of the label-extended graph (see Section 2). Our main technical contribution to overcome this barrier is to bound the threshold rank of the label-extended graph in terms of the threshold rank of G ; combining this inequality with fast SDP solvers using matrix multiplicative weights leads to the algorithm of Theorem 1.4.

Which Graphs Have Small Threshold Rank? As noted in prior work, many interesting families of graphs have low threshold rank. Random graphs with degree d have threshold rank 1 for $\varepsilon > c/\sqrt{d}$; more generally, one-sided spectral expanders have positive threshold rank 1, and two-sided spectral expanders have negative threshold rank equal to zero.

What about graphs beyond expanders? Focusing on the positive threshold rank case, graphs in which all small sets are highly expanding have bounded threshold rank [Ste10b]. And higher-order Cheeger inequalities like [LGT14] say that if a graph has no sparse k -way cut, then the k -th largest eigenvalue of the normalized adjacency matrix is bounded from above, which yields a bound on the threshold rank. Thus, our algorithm for 2CSPs is efficient on graphs satisfying this combinatorial property.

Which graphs have small negative threshold rank—the setting of our MAX-CUT algorithm—is less explored. Intuitively, a graph G with small $\text{rank}_{\leq -\varepsilon}(A)$ should have “few” cuts which cut significantly more than half of the edge set. But to our knowledge, *dual* Cheeger inequalities like [Tre12] and higher-order dual Cheeger inequalities like [Liu15] only describe combinatorial properties of G which imply bounds on $\text{rank}_{\leq -(1-\varepsilon)}(A)$, the number of eigenvalues close to -1 , which could be much smaller than the number of eigenvalues less than $-\varepsilon$. Thus we pose the following problem: *provide a combinatorial characterization of graphs G with few eigenvalues less than $-\varepsilon$.*

Returning to general 2CSPs, if we apply a standard reduction [Kol10] from a general 2CSP to a quadratic optimization problem, we get running times in terms of the threshold rank of the so-called *label-extended* graph of the CSP. One of our key technical insights is in relating the threshold rank of this graph to the threshold rank of the constraint graph A . We discuss this in more detail in Section 4. A separate argument shows that for dense graphs G , the threshold rank of the label-extended graph is at most $O(q^2/\varepsilon^2)$, meaning that our algorithm also recovers the best known running time for a PTAS for dense 2CSPs on constant-sized alphabets [Yar14, YZ14].

1.2 Techniques

Algorithm for Boolean Quadratic Optimization and MAX-CUT. Our algorithm is a novel combination of prior SDP-based approaches and subspace enumeration techniques. We will sketch the proof of Theorem 1.1 in the case where A has $\text{diag}(A) = \mathbf{0}$.³ Recall that the canonical SDP relaxation of the problem $\max_{x \in \{\pm 1\}^n} x^\top A x$ is as follows:

$$\begin{aligned} \max_{X \in \mathbb{R}^{n \times n}, X \succeq 0} \quad & \langle A, X \rangle \\ \text{s.t.} \quad & \text{diag}(X) = \mathbf{1}. \end{aligned}$$

Equivalently, for reasons we will see in a moment, we could optimize over $X \in \mathbb{R}^{n+1 \times n+1}$:

$$\begin{aligned} \max_{X \in \mathbb{R}^{n+1 \times n+1}, X \succeq 0} \quad & \langle A, X_{[1, n+1], [1, n+1]} \rangle \\ \text{s.t.} \quad & \text{diag}(X) = \mathbf{1}, \end{aligned}$$

³ This assumption is without loss of generality, since the diagonal contributes a constant amount to the objective on any $x \in \{\pm 1\}^n$.

where $X_{[1,n+1],[1,n+1]}$ is the bottom right $n \times n$ submatrix of X .

A solution to the second relaxation can be interpreted as an linear operator $\tilde{\mathbf{E}} : \mathbb{R}[x_1, \dots, x_n]_{\leq 2} \rightarrow \mathbb{R}$, which maps polynomials in x_1, \dots, x_n of degree at most 2 to scalars; we call $\tilde{\mathbf{E}}$ a *pseudoexpectation*: we can take $\tilde{\mathbf{E}}x_i = X_{0,i}$ and $\tilde{\mathbf{E}}x_i x_j = X_{i,j}$ and extend linearly. In this language, the SDP above becomes $\max_{\tilde{\mathbf{E}}} \tilde{\mathbf{E}}\langle x, Ax \rangle$. While we will not use the full machinery associated with this point of view and the corresponding Sum-of-Squares semidefinite hierarchy, we still use the pseudoexpectation language for ease of exposition. Note that the PSD constraint ensures that the pseudoexpectation of any square polynomial is non-negative, and notably that $\tilde{\mathbf{E}}xx^\top - (\tilde{\mathbf{E}}x)(\tilde{\mathbf{E}}x)^\top$ is a PSD matrix.

As a first attempt at an algorithm for quadratic optimization, consider the following naive rounding scheme applied to the optimal $\tilde{\mathbf{E}}$ – round each x_i independently by flipping a ± 1 -valued coin with expectation $\tilde{\mathbf{E}}x_i$. Of course, by symmetry there is an optimal solution $\tilde{\mathbf{E}}$ for which $\tilde{\mathbf{E}}x_i = 0$ for all i , meaning that this rounding scheme will not give any nontrivial guarantees unless we modify the SDP. But, to see how to modify the SDP, it will be helpful to further analyze this rounding scheme.

Note that the solution produced has (in expectation) objective value $(\tilde{\mathbf{E}}x)^\top A (\tilde{\mathbf{E}}x)$ and thus we can write difference between the objective value of the SDP and the rounded objective value as:

$$\langle A, \tilde{\mathbf{E}}xx^\top \rangle - \langle A, (\tilde{\mathbf{E}}x)(\tilde{\mathbf{E}}x)^\top \rangle = \langle A, \widetilde{\mathbf{Cov}} \rangle,$$

where $\widetilde{\mathbf{Cov}} = \tilde{\mathbf{E}}xx^\top - (\tilde{\mathbf{E}}x)(\tilde{\mathbf{E}}x)^\top$. We call this quantity the *rounding error*. Our goal is to bound it from above.

We can decompose the rounding error in two parts: that incurred on the subspace spanned by the eigenvalues of A that are less than ε (which we refer to as $A_{<\varepsilon}$), and that incurred on the part of A with eigenvalues at least ε (which we refer to as $A_{\geq\varepsilon}$):

$$\langle A, \widetilde{\mathbf{Cov}} \rangle = \langle A_{<\varepsilon}, \widetilde{\mathbf{Cov}} \rangle + \langle A_{\geq\varepsilon}, \widetilde{\mathbf{Cov}} \rangle.$$

First off, the $\langle A_{<\varepsilon}, \widetilde{\mathbf{Cov}} \rangle$ part of the rounding error cannot be too large for any SDP solution $\tilde{\mathbf{E}}$. Since $\widetilde{\mathbf{Cov}}$ is positive semidefinite, we can bound $\langle A_{<\varepsilon}, \widetilde{\mathbf{Cov}} \rangle \leq \varepsilon \cdot \text{Tr}(\widetilde{\mathbf{Cov}}) \leq \varepsilon n$. Thus, for the rounding to be successful – incurring error at most $O(\varepsilon n)$ – we only require that the SDP solution $\tilde{\mathbf{E}}$ satisfies $\langle A_{\geq\varepsilon}, \widetilde{\mathbf{Cov}} \rangle \leq \varepsilon n$. However, in general, the SDP solution associated with this optimization problem might not have this property.

This brings us to a modified SDP relaxation of $\max_{x \in \{\pm 1\}^n} x^\top A x$. Suppose that we knew a vector v such that $\|v - \Pi x^*\|^2 \leq \varepsilon n$, where Π is the projector to image of $A_{\geq\varepsilon}$ and x^* is an optimal solution to $\max_{x \in \{\pm 1\}^n} x^\top A x$. This is not too much to assume, since by enumerating over at most $(1/\varepsilon)^{O(\text{rank}_{\geq\varepsilon}(A))}$ vectors v in the subspace $\text{Im}(A_{\geq\varepsilon})$, we can guess such a v . Then the following SDP would still be a relaxation:

$$\max \tilde{\mathbf{E}}\langle x, Ax \rangle \text{ s.t. } \tilde{\mathbf{E}}\|\Pi x - v\|^2 \leq \varepsilon n. \quad (1)$$

Crucially, if $\tilde{\mathbf{E}}$ is any pseudoexpectation for which there exists v such that $\tilde{\mathbf{E}}\|\Pi x - v\|^2 \leq \varepsilon n$, then

$$\langle A_{\geq\varepsilon}, \widetilde{\mathbf{Cov}} \rangle = \langle A_{\geq\varepsilon}, \Pi \widetilde{\mathbf{Cov}} \Pi \rangle \leq \|A_{\geq\varepsilon}\| \cdot \text{Tr} \Pi \widetilde{\mathbf{Cov}} \Pi \leq \tilde{\mathbf{E}}\|\Pi(x - \tilde{\mathbf{E}}x)\|^2.$$

Furthermore, using triangle inequality and PSD-ness of $\tilde{\mathbf{E}}$, we have

$$\tilde{\mathbf{E}}\|\Pi(x - \tilde{\mathbf{E}}x)\|^2 \leq 2(\tilde{\mathbf{E}}\|\Pi x - v\|^2 + \|v - \tilde{\mathbf{E}}\Pi x\|^2) \leq 4\tilde{\mathbf{E}}\|\Pi x - v\|^2.$$

Putting it together, we get that for any feasible $\tilde{\mathbf{E}}$ for (1), the rounding error is at most $O(\epsilon n)$.

This brings us to our final algorithm for Boolean quadratic optimization. Given A , enumerate the image of $A_{\geq \epsilon}$. For each v in the enumeration, solve (1) and round by sampling coordinates independently. Output the best solution found in this way.

Since (up to an additive shift and rescaling) MAX-CUT on regular graphs is just $\max x^\top (-A)x$, where A is the normalized adjacency matrix, this gives our $\text{poly}(1/\epsilon)$ time algorithm for MAX-CUT on regular graphs with few negative eigenvalues (Theorem 1.3). We note that this argument can be generalized to irregular graphs and we defer this to Section 3.

Algorithm for General 2CSPs, and Threshold Ranks of Label-Extended Graphs Now we move on to our algorithm for general 2CSPs with alphabet size $[q]$. It is by now standard to associate an n -variable 2CSP instance φ with variables x_1, \dots, x_n and constraints $\varphi_{ij} : [q] \times [q] \rightarrow \{0, 1\}$ a *label-extended graph* M on vertex set $[n] \times [q]$, where (i, a) is adjacent to (j, b) if $x_i = a, x_j = b$ satisfies φ_{ij} [Kol10]. Using largely the same algorithm and argument as in the Boolean case, we can obtain a $(1 + O(\epsilon))$ approximation algorithm in time $(1/\epsilon)^{O(\text{rank}_{\geq \epsilon}(M))} \cdot (n)^{O(1)}$.

The prior work [BRS11] gives a $(1 + O(\epsilon))$ approximation algorithm for 2CSPs with running time depending only on the threshold rank of the base graph G ; their algorithm runs in time $\exp(\tilde{O}(\text{rank}_{\geq \epsilon^2}(G)/\epsilon^4))(n)^{O(1)}$. We recover this running time (and later, unlike [BRS11], speed up to near-linear time) by proving a new inequality relating $\text{rank}_{\geq \epsilon}(M)$ to $\text{rank}_{\geq \epsilon^2}(G)$. Namely, we show that $\text{rank}_{\geq 2q\epsilon}(M) \leq \text{rank}_{\geq \epsilon^2}(G)/\epsilon^4$ (Corollary 4.2). We actually observe that this bound follows from a small adaptation of an argument by [BHSV25], itself adapted from [BRS11].

From Polynomial to Near-Linear Time The main contributor to the running time of our algorithm is the time to solve the SDPs. To speed this up from $(mn)^{O(1)}$ time to $\tilde{O}(m + n)$, we rely on the matrix multiplicative weights framework as developed in [Ste10a], which gives a near-linear time SDP solver for the basic SDP relaxation of CSPs. To adapt this approach to our setting, we just need to enforce the additional constraint $\tilde{\mathbf{E}}\|\Pi x - v\|^2 \leq \epsilon n$ when solving the SDP. This turns out not to be too difficult. The key quantity governing the difficulty of enforcing a constraint in the multiplicative weights framework is its “width”; in this case, the operator norm of the matrix which forms the constraint. Enforcing this extra constraint turns out to be possible with only $\text{poly}(q/\epsilon)$ width, meaning that the framework of [Ste10a] can solve our SDPs in time $\text{poly}(q/\epsilon) \cdot \tilde{O}(m + n)$.

1.3 Related Work

SDP-based algorithms. Semidefinite programming and SDP hierarchies have been used to give algorithms for CSPs under varying threshold rank conditions [BRS11, GS11, GS12]. The work of [BRS11] achieves similar approximation guarantees to that of our work, but incurs a polynomial as opposed to near-linear dependence in runtime on the size of the input. For restricted classes of CSPs (including MAX-CUT), the work of Guruswami–Sinop [GS11, GS12] yields algorithms with

similar approximation guarantees. Furthermore, in the case of 3-coloring, recent work by Hsieh utilizes SDP-based approaches to find proper 3-colorings on almost half the edges in 3-colorable graphs with small threshold rank [Hsi25].

Subspace enumeration algorithms. Subspace enumeration has been an influential algorithm design technique for CSPs, specifically Unique Games, and related problems such as small set expansion [Kol10, ABS15]. These works give algorithms whose runtime depends on the threshold rank of the *label-extended* graph. A key challenge in these approaches is bounding this quantity, since the label-extended graph having bounded threshold rank is a stronger condition than the constraint graph having bounded threshold rank. We discuss threshold rank bounds further in the paragraph below.

Regularity lemmas. Regularity lemmas have been a key tool for giving fast algorithms for constraint satisfaction and related problems. Their use dates back to early approximation schemes for dense CSPs in the 90s and early 2000s [FK99, AdIVKK02], and more recent work extended these ideas by giving a regularity lemma for graphs with low threshold rank [OGT13]. Extensions of this work to higher-arity CSPs have also been influential in the design of fast decoders for error correcting codes with high rate [JST21].

Dense CSPs. Many prior works have studied approximation schemes for CSPs on graphs which are either dense or pseudo-dense [FK99, AdIVKK02, ACOH⁺10, COCF10]. Note that all dense and pseudo-dense graphs have low threshold rank, and thus our results and prior results on approximation schemes for graphs with low threshold rank apply to a larger class of instances.

Dual higher-order Cheeger inequalities. Higher-order dual Cheeger inequalities [Tre12, Liu15] give a relationship between the magnitude of λ_{n-k} and combinatorial properties of the graph. In particular, they can be used to show that graphs with this combinatorial property—that all vertex-disjoint non-empty subsets V_1, \dots, V_k must have at least one V_i which is far from being a bipartite connected component of the original graph—have eigenvalues other than $\lambda_{n-k}, \dots, \lambda_n$ bounded away from -1 . However, this does not yield a combinatorial class of graphs which our algorithm runs fast on since higher-order dual Cheeger inequalities are too quantitatively loose to give bounds on $\text{rank}_{\leq -\epsilon}$.⁴

Threshold rank bounds. As discussed above, a line of prior work yielded algorithms for Unique Games which runs in time parameterized by the threshold rank of the label-extended graph [Kol10, ABS15]. Such work also studied the relationship between the threshold rank of the constraint graph and the threshold rank of the label-extended graph. However, existing prior work all lose polynomial factors in n [ABS15].

⁴ This is due to the fact that the inequality becomes trivial when considering eigenvalues which are negative but very close to 0.

2 Preliminaries

Threshold Rank and Label Extension. We first formally define the threshold rank of a matrix as well as the label extended graph of a CSP.

Definition 2.1 (Threshold Rank). Let $A \in \mathbb{R}^{n \times n}$ be a matrix with real eigenvalues, and let $0 < \varepsilon < 1$. The (one-sided) ε -threshold rank of A , denoted $\text{rank}_{\geq \varepsilon}(A)$ or $\text{rank}_{\leq -\varepsilon}$, is the number of eigenvalues of A which are at least ε or less than $-\varepsilon$ respectively.

Definition 2.2 (Label Extended Graph). Let G be a graph on $[n]$ with adjacency matrix A . For each edge ij in G let $\phi_{ij} : [q]^2 \rightarrow \{0, 1\}$ be a predicate on alphabet size q . The label extended graph G' is a graph with vertices $\{v_{i,\alpha}\}_{i \in [n], \alpha \in [q]}$ such that there is an edge between (i, α) and (j, β) if and only if $ij \in G$ and $\phi_{ij}(\alpha, \beta) = 1$.

Semidefinite programming and pseudodistributions. Although we do not use the full power of the sum-of-squares programming hierarchy, we will still use the pseudoexpectation notation for ease of exposition.

Definition 2.3 (Pseudodistribution). A *pseudodistribution* D of degree t is a function from \mathbb{R}^n to \mathbb{R} with finite support such that $\sum_{x \in \text{supp}(D)} D(x) = 1$ and $\sum_{x \in \text{supp}(D)} D(x) p(x)^2 \geq 0$ for all polynomials $p(x)$ with $\deg(p(x)^2) \leq t$.

Definition 2.4 (Pseudoexpectation). Given a pseudodistribution D of degree t , the associated *pseudoexpectation* $\tilde{\mathbf{E}}_D$ is the linear operator from the space of functions to \mathbb{R} that is defined by f mapping to $\tilde{\mathbf{E}}_D f(x) := \sum_{x \in \text{supp}(D)} D(x) f(x)$.

The non-negativity of the pseudoexpectation on squared polynomials implies the following easy fact.

Fact 2.5. Let $\tilde{\mathbf{E}}$ be a degree t pseudoexpectation and let its pseudocovariance matrix be $\widetilde{\mathbf{Cov}} = \tilde{\mathbf{E}}_{xx^\top} - (\tilde{\mathbf{E}}x)(\tilde{\mathbf{E}}x)^\top$. Then $\widetilde{\mathbf{Cov}} \succeq 0$.

Definition 2.6 (Constrained pseudodistributions). Let $\mathcal{A} = \{p_1 \geq 0, p_2 \geq 0, \dots, p_r \geq 0\}$ be a system of r polynomial inequality constraints of degree at most d in m variables. Let μ be a degree- ℓ pseudodistribution over \mathbb{R}^m . We say that μ *satisfies* \mathcal{A} at degree $\ell \geq 1$ if for every subset $\mathcal{S} \subset [r]$ and every sum-of-squares polynomial q such that $\deg(q) + \sum_{i \in \mathcal{S}} \max(\deg(p_i), d) \leq \ell$, $\tilde{\mathbf{E}}_\mu q \prod_{i \in \mathcal{S}} p_i \geq 0$. Further, we say that μ *approximately satisfies* \mathcal{A} if the above inequalities are satisfied up to additive error $\tilde{\mathbf{E}}_\mu q \prod_{i \in \mathcal{S}} p_i \geq -2^{-n^\ell} \|q\| \prod_{i \in \mathcal{S}} \|p_i\|$, where $\|\cdot\|$ denotes the Euclidean norm of the coefficients of the polynomial, represented in the monomial basis.

Note that when we have a pseudodistribution of degree 2 then there is an easy weak separation oracle for the convex set of moment tensors of constrained pseudodistributions via checking that: (1) the moment matrix is PSD, and (2) $\tilde{\mathbf{E}} p_i \geq 0$ for all $p_i \in \mathcal{A}$ ⁵. Furthermore, if \mathcal{A} is *explicitly bounded*, in that it contains a constraint of the form $\{\|x\|^2 \leq 1\}$, the above observation in combination with [GLS81] yields the following fact.

⁵ There is also a weak separation oracle for pseudoexpectation of higher degree, see [Sho87, Par00].

Theorem 2.7 (Efficient optimization over pseudodistributions of degree 2). *There exists an $(m + r)^{O(1)}$ -time algorithm that, given any explicitly bounded and satisfiable system \mathcal{A} of r polynomial constraints in m variables, outputs a degree-2 pseudodistribution that satisfies \mathcal{A} approximately, in the sense of Definition 2.6.⁶*

3 Algorithm and Analysis

3.1 Improving the runtime dependence on ε

In this section, we shall describe our primary algorithm, with a running time depending on the threshold rank of the label-extended graph.

Theorem 3.1. *Let $M \in \mathbb{R}^{nq \times nq}$ be a symmetric matrix such that $\|M\|_{\text{op}} \leq 1$, and the n diagonal $q \times q$ blocks of M are all identically 0. Also, let $D \in \mathbb{R}^{n \times n}$ be an arbitrary diagonal matrix with non-negative entries and $E = D \otimes \text{Id}_q$. Let $k = \text{rank}_{\geq \varepsilon}(M)$. Consider the optimization problem*

$$\text{OPT} := \max_{y \in \mathcal{C}_q^n} \Phi(y) = \max_{y \in \mathcal{C}_q^n} y^\top E^{1/2} M E^{1/2} y,$$

where \mathcal{C}_q^n is the subset of $y \in \{0, 1\}^{nq}$ such that each of the n blocks of y (each of size q) has exactly one 1. There exists an algorithm running in time $\text{poly}(n) \cdot \left(\frac{1}{\varepsilon}\right)^{O(k)}$ that returns a random $y \in \mathcal{C}_q^n$ with

$$\mathbb{E}\Phi(y) \geq \text{OPT} - O(\varepsilon \text{Tr } D).$$

Given this, our result on quadratic optimization over the hypercube follows near-immediately. We restate it for convenience.

Theorem 1.1 (Quadratic optimization). *Let $A \in \mathbb{R}^{n \times n}$ be an arbitrary symmetric matrix, scaled to have $\|A\|_{\text{op}} \leq 1$, and let $\text{OPT} = \max_{x \in \{\pm 1\}^n} x^\top A x$. Set $k = \text{rank}_{\geq \varepsilon}(A)$. Then, there is an algorithm running in time $\text{poly}(n) \cdot \left(\frac{1}{\varepsilon}\right)^{O(k)}$ that outputs a (random) vector $\hat{x} \in \{\pm 1\}^n$ such that*

$$\mathbb{E}\hat{x}^\top A \hat{x} \geq \text{OPT} - O(\varepsilon n).$$

Proof. We may assume without loss of generality that the diagonal entries of A are all 0, since they contribute a constant when optimizing over the hypercube. The result then immediately follows

from Theorem 3.1 on setting $q = 2$, $D = \text{Id}_n$, and $M = A \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. ■

The algorithm for Theorem 3.1 is described in Algorithm 1.

For the remainder of this section, suppose one has a pseudoexpectation $\tilde{\mathbb{E}}$ over boolean variables $(y_{i,\alpha})_{i \in [n], \alpha \in [q]}$, satisfying the constraint that exactly one of the $(y_{i,\alpha})_{\alpha \in [q]}$ is 1 for every i (this enforces the requirement that $y \in \mathcal{C}_q^n$).

Finally, consider the product distribution μ^\otimes over $[q]^n$ such that $\mu(\alpha_1, \dots, \alpha_n) = \prod_{j \in [n]} \tilde{\mathbb{E}} y_{j,\alpha_j}$. We also have the associated distribution ν over \mathcal{C}_q^n induced by μ^\otimes : in other words, the different blocks are independent, and within each block, the probabilities of the different symbols are given by the marginals under the pseudodistribution.

⁶ Here, we assume that the bit complexity of the constraints in \mathcal{A} is $(m + t)^{O(1)}$.

Algorithm 1: Solving CSPs faster on bounded threshold rank graphs

Input: $M \in \mathbb{R}^{nq \times nq}$ with $\text{rank}_{>\varepsilon}(M) = k$ and $\|M\|_{\text{op}} \leq 1$, and PSD diagonal $D \in \mathbb{R}^{n \times n}$

- 1 $\Pi \leftarrow$ projection onto the top k -dimensional eigenspace of M
- 2 $\mathcal{S} \leftarrow \sqrt{\varepsilon \cdot \text{Tr } D}$ -net of $\sqrt{\text{Tr } D} \cdot \mathbb{B}^k$.
- 3 $E \leftarrow D \otimes \text{Id}_q$.
- 4 **for** $v \in \mathcal{S}$ **do**
- 5 Find a degree-2 pseudoexpectation $\tilde{\mathbf{E}}$ over $[q]^n$ optimizing $\tilde{\mathbf{E}}y^\top E^{1/2} M E^{1/2} y$, subject to the constraints $y_i(1 - y_i) = 0$, $\sum_{j \in [q]} y_{qi+j} = 1$, and $\tilde{\mathbf{E}}\|\Pi E^{1/2} y - v\|_2^2 \leq \varepsilon \text{Tr } D$.
- 6 Round $\tilde{\mathbf{E}}$ to a (random) $y_v \in [q]^n$ by independently sampling each index $(y_v)_i$ according to the marginals prescribed by $\tilde{\mathbf{E}}$.
- 7 **return** the y_v that maximizes $\Phi(y_v)$

Lemma 3.2. Let Π be the projector onto the eigenspace of M with eigenvalues larger than ε , and $v \in \mathbb{R}^{nq}$ an arbitrary vector. Then,

$$\tilde{\mathbf{E}}\Phi(y) - \mathbf{E}_{y \sim v}\Phi(y) \leq \varepsilon \text{Tr } D + 4\tilde{\mathbf{E}}\left\|\Pi E^{1/2}(y - v)\right\|_2^2.$$

Proof. Because the diagonal blocks of M are 0, we are interested in

$$\tilde{\mathbf{E}}y^\top E^{1/2} M E^{1/2} y - \left(\tilde{\mathbf{E}}y\right)^\top E^{1/2} M E^{1/2} \left(\tilde{\mathbf{E}}y\right) = \left\langle E^{1/2} M E^{1/2}, \widetilde{\mathbf{Cov}} \right\rangle,$$

where $\widetilde{\mathbf{Cov}} = \tilde{\mathbf{E}}y y^\top - (\tilde{\mathbf{E}}y)(\tilde{\mathbf{E}}y)^\top$ is the “pseudocovariance” of $\tilde{\mathbf{E}}$. We now decompose A into its different eigenspaces and bound each separately. We have

$$\begin{aligned} \left\langle E^{1/2} M E^{1/2}, \widetilde{\mathbf{Cov}} \right\rangle &= \left\langle M_{<\varepsilon}, E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \right\rangle + \left\langle M_{\geq \varepsilon}, E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \right\rangle \\ &\leq \varepsilon \text{Tr} \left(E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \right) + \left\langle M_{\geq \varepsilon}, E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \right\rangle \\ &\leq \varepsilon \text{Tr} \left(E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \right) + \text{Tr} \left(\Pi E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \Pi \right). \end{aligned}$$

Note that the first term is

$$\varepsilon \left\langle E, \widetilde{\mathbf{Cov}} \right\rangle = \varepsilon \sum_{\substack{i \in [n] \\ \alpha \in [q]}} D_{ii} \tilde{\mathbf{E}}y_{(i,\alpha)}^2 \leq \varepsilon \sum_{i \in [n]} D_{ii} \sum_{\alpha \in [q]} \tilde{\mathbf{E}}y_{(i,\alpha)} = \varepsilon \text{Tr}(D),$$

utilizing that $\tilde{\mathbf{E}}y_{qi+j}^2 \leq \tilde{\mathbf{E}}y_{qi+j}$ and $\sum_{j \in [q]} \tilde{\mathbf{E}}y_{qi+j} = 1$. Let $m = \tilde{\mathbf{E}}y$. The second trace may be bounded as

$$\begin{aligned} \text{Tr} \left(\Pi E^{1/2} \widetilde{\mathbf{Cov}} E^{1/2} \Pi \right) &= \tilde{\mathbf{E}} \left\| \Pi E^{1/2} y - \Pi E^{1/2} m \right\|_2^2 \\ &\leq 2 \cdot \left(\tilde{\mathbf{E}} \left\| \Pi E^{1/2} (y - v) \right\|_2^2 + \left\| \Pi E^{1/2} (m - v) \right\|_2^2 \right) \\ &\leq 4 \cdot \tilde{\mathbf{E}} \left\| \Pi E^{1/2} (y - v) \right\|_2^2, \end{aligned}$$

where the last inequality is Jensen’s, and the first is the almost-triangle inequality. ■

We may now prove the result in this section with suboptimal running time.

Proof of Theorem 3.1. We first observe that there exists some $v \in \mathcal{S}$ such that the associated optimizing pseudoexpectation has objective value at least OPT : indeed, consider an optimizer $y \in \mathcal{C}_q^n$, and observe that $\|\Pi E^{1/2} y\|_2^2 \leq \|D^{1/2} \mathbf{1}_n\|_2^2 \leq \text{Tr } D$ since y has exactly one non-zero entry in each q -length block, so there indeed exists a $v \in \mathcal{S}$ such that $\|\Pi E^{1/2} y - v\|_2^2 \leq \varepsilon \cdot \text{Tr } D$. Then, the Dirac distribution supported only on y has objective value OPT and satisfies our constraints for this v .

Lemma 3.2 implies that Algorithm 1 returns a point x such that $\mathbb{E}\Phi(x) \geq \text{OPT} - O(\varepsilon \text{Tr } D)$, as claimed. For the running time guarantee, we may use a standard SDP solver that runs in $\text{poly}(n)$ time to find the pseudoexpectation of interest for each v . Thus, the running time is just $|\mathcal{S}| \cdot \text{poly}(n)$. The bound on the size of \mathcal{S} , and thus the running time, is folklore. ■

3.2 Improving the runtime's dependence on n for 2CSPs

In this section, we describe how to improve the $\text{poly}(n, q) \cdot (\frac{1}{\varepsilon})^{\tilde{O}(k)}$ runtime in the previous section to a near-linear time algorithm when M corresponds to the normalized adjacency matrix of a label-extended graph. This yields a near-linear time algorithm for arbitrary 2CSPs when the threshold rank of the label-extended graph is small⁷. Furthermore, with a small modification, we will show how to extend this to give a near-linear time algorithm for MAX-CUT on graphs where the normalized negative adjacency matrix has small threshold rank.

Theorem 3.3 (Fast Algorithm for 2CSPs with Small Label-Extended Threshold Rank). *Let G be a graph on $[n]$ with m edges, with adjacency matrix A and degree diagonal D . For each edge ij in G , let $\varphi_{ij} : [q]^2 \rightarrow \{0, 1\}$ be a predicate on alphabet size q that is not identically 0 and let M be the label-extended graph associated with the constraint graph A and constraints φ_{ij} . Let $E = D \otimes \text{Id}_q$. Also let $\text{OPT} = \max_{x \in [q]^n} \Phi(x) := \max_{x \in [q]^n} \sum_{ij \in E(G)} \varphi_{ij}(x_i, x_j)$. Set $k = \text{rank}_{\geq \varepsilon}(E^{-1/2} M E^{-1/2})$. Then, Algorithm 1 can be implemented in time $\tilde{O}(n + m) \cdot (\frac{1}{\varepsilon})^{O(k)} \cdot \text{poly}(q)$ and finds an assignment \hat{x} with $\Phi(\hat{x}) \geq \text{OPT} - O(\varepsilon q^2 m)$.*

Note that the main bottleneck for runtime in Algorithm 1 is solving the SDP. The approximation guarantees of our algorithm will follow via the analysis in the proof of Theorem 3.1 and thus it suffices to show that we can find an (approximate) solution to the SDP in Algorithm 1 in near-linear time. We utilize existing techniques from [Ste10a], which give a fast SDP solver that runs in time $\text{poly}(k, q, 1/\varepsilon) \cdot \tilde{O}(n + m)$. Since our setup is very similar to theirs and our improved runtime largely follows from prior work, we will only sketch the main differences in the proof.

Proof of Theorem 3.3. Note that the approximation guarantees of the theorem follow from Theorem 3.1, provided that we can quickly find an approximate SDP solution where on average constraints are approximately satisfied⁸ and our additional constraint holds up to accuracy $\varepsilon \text{Tr}(D)/100$. Below, we will briefly sketch the algorithm and analysis of the fast SDP solver.

⁷ In Section 4 we will give runtime guarantees in terms of the threshold rank of the constraint graph by comparing these two quantities.

⁸ As noted in [Ste10a] if they are approximately satisfied on average, then by [RS09] we can modify the solution to satisfy all constraints exactly at the expense of decreasing the objective value by $\varepsilon q^2 m$.

We will find an (approximate) solution to the relevant SDP in near-linear time via matrix multiplicative weights, as in [Ste10a]. Note that we can represent all the relevant moments of our pseudo expectation via the following matrix:

$$\tilde{\mathbf{E}} \begin{pmatrix} 1 & \frac{1}{\sqrt{qn}}y \end{pmatrix}^\top \begin{pmatrix} 1 & \frac{1}{\sqrt{qn}}y \end{pmatrix}.$$

Observe that the bottom right $qn \times qn$ submatrix is a PSD matrix with trace 1 and corresponds directly to the SDP solution computed in prior work [Ste10a]. Note that via [Ste10a] we can find such a matrix which optimizes the objective function in near-linear time, without taking into account our additional constraint. The main remaining piece is to show that we can also enforce our new constraint in the matrix multiplicative weights solver. The two ingredients towards this will be to show that (a) the new constraint is low-width and (b) we can efficiently detect violations of the constraint only given access to a sketch of the proposed “feasible” solution.

First, $\tilde{\mathbf{E}}_\mu \|\Pi E^{1/2}x - v\|^2 \leq \varepsilon \text{Tr}(D)$ is indeed a low-width constraint. The hyperplane corresponding to this constraint has operator norm $O(\text{Tr}(E))$, while we need the constraint to be satisfied up to $\varepsilon/100 \cdot \text{Tr}(D)$ slack. Thus, the width of the constraint is $O(q/\varepsilon)$.

We now describe how to efficiently detect a violation of the constraint given a sketched version of the solution. Recall that every iteration we are given $\tilde{W}_t \in \mathbb{R}^{\log(qn) \times (qn)}$ such that $\tilde{W}^\top \tilde{W}$ is the sketched SDP solution. Note that in order to detect the violation of the constraint in the sketched SDP, we only need to know $\Pi'(E')^{1/2} \tilde{W}^\top \tilde{W} (E')^{1/2} \Pi'$, where Π' is the projection of the last n coordinates to the relevant subspace and E' is a $(n+1) \times (n+1)$ dimensional block matrix with E in the lower right block, Id_1 in the top left block, and zeros elsewhere⁹ Note that this is a low rank matrix, since Π' has rank $k+1$. Thus, if we just compute $B \Pi' \tilde{W}_t$, where $B \in \mathbb{R}^{(k+1) \times n}$ simply is a change of basis, then by looking at $(B \Pi' \tilde{W}_t)^\top (B \Pi' \tilde{W}_t)$ we can determine if $\tilde{W}^\top \tilde{W}$ satisfies the constraint. The fact that it suffices to check whether the solution would be satisfied by the sketch follows from prior analysis.

Furthermore, we can efficiently compute $(B \Pi' \tilde{W}_t)^\top (B \Pi' \tilde{W}_t)$. It suffices to be able to compute $B \Pi'$ in near-linear time, and this matrix can be constructed quickly given the relevant top eigenvectors of M . Furthermore, note that instead of having Π be the projection to the eigenspaces with eigenvalues at least ε , it also suffices to have Π be the projection to some subspace which includes all eigenvectors with eigenvalues at least 2ε and has dimension at most $\text{rank}_{\geq \varepsilon}(M)$.¹⁰ Thus, it suffices to find a collection of at most $\text{rank}_{\geq \varepsilon}(M)$ vectors such that every eigenvector with eigenvalue at least 2ε is in close to the span. We can compute this via the power method—note that between 2ε and ε there must exist a gap between two eigenvectors, and thus it suffices to take $\log n$ iterations of a top- k power method. Each iteration takes time $O(kq^2m)$ since M has at most q^2m non-zero entries.

Finally, we will argue that we can implement our rounding algorithm when only given access to the sketched solution, without computing the full $qn \times qn$ solution matrix. In order to round the solution, it suffices to know $\tilde{\mathbf{E}}y$, since this defines the marginal distribution on each coordinate, and

⁹ Π' is not equal to Π since our submatrix also includes an initial row and column for the expectation.

¹⁰ We simply lose a factor of 2 in the approximation guarantee, which still yields a $O(\varepsilon q^2 m)$ additive approximation.

we can then independently round each coordinate. However, this is a $1 \times qn$ submatrix of $\tilde{W}^\top \tilde{W}$ which can be reconstructed in $\tilde{O}(qn)$ time from the Gram vectors. ■

Theorem 1.3 ($(1 + O(\varepsilon))$ Approximation for MAX-CUT in $\text{poly}(1/\varepsilon)$ Time). *Let G be a graph on vertex set $[n]$ with m edges, with adjacency matrix A and diagonal degree matrix D . Let OPT be the size of the largest cut in G . Set $k = \text{rank}_{\leq -\varepsilon}(D^{-1/2}AD^{-1/2})$. In time $(\frac{1}{\varepsilon})^{O(k)} \cdot \tilde{O}(n + m)$, one can output a (random) cut attaining value $(1 - O(\varepsilon)) \cdot \text{OPT}$ in expectation.*

Given the proof of [Theorem 3.3](#), the proof of [Theorem 1.3](#) follows via some small modifications.

Proof of Theorem 1.3. We aim to apply [Theorem 3.3](#), but with a small twist. Note that we can write the objective value corresponding to the MAX-CUT in [Theorem 3.1](#) as follows:

$$\frac{1}{2} \left[y^\top M_A^1 y + y^\top \left(-A \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \right) y \right],$$

where M_A^1 is the label-extended graph of the CSP with constraint graph A and every constraint being trivially satisfied. Since the first term has a fixed value for every $y \in \mathcal{C}_2^n$, it is equivalent to optimizing the second term, or the quadratic optimization problem associated with $M = -D^{-1/2}AD^{-1/2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ and $E = D \otimes \text{Id}_2$. Thus, by [Theorem 3.1](#) we have that [Algorithm 1](#) incurs a small rounding error for $k = \text{rank}_{\geq \varepsilon}(M) = O(\text{rank}_{\leq -\varepsilon}(A))$ when utilizing the matrices (and associated projections to the top eigenspaces) described in the prior line. Furthermore, since the original formulation of the objective corresponds to a quadratic optimization associated with the label-extended graph of a CSP, we can apply [Theorem 3.3](#) to show that [Algorithm 1](#) can be implemented in $\tilde{O}(n + m) \cdot (\frac{1}{\varepsilon})^{O(k)} \cdot \text{poly}(q)$ time since the relevant SDPs are equivalent (up to this $y^\top M_A^1 y$ fixed additive shift) to optimization on the label-extended graph of a CSP. ■

4 Threshold Rank of the Signed and Label-Extended Adjacency Matrix

We prove the following theorem relating the threshold rank of real-valued signings of adjacency matrices to the threshold rank of the underlying adjacency matrix. The proof is a small adaptation of an argument from [\[BHSV25\]](#), which is itself an adaptation of an argument from [\[BRS11\]](#).

Theorem 4.1. *Suppose $A \in \mathbb{R}_{\geq 0}^{n \times n}$ is symmetric with $\|A\| \leq 1$. Let $B \in \mathbb{R}^{n \times n}$ be symmetric with $|B_{ij}| \leq A_{ij}$ for all i, j . If $\text{rank}_{\geq \tau}(A) \leq s$, then for any $\sigma > 0$,*

$$\text{rank}_{\geq \sqrt{\tau(1-\sigma)} + \sigma}(B) \leq \frac{s}{\sigma^2}.$$

Corollary 4.2. *Let $A \in \mathbb{R}_{\geq 0}^{n \times n}$ with $\|A\| \leq 1$. For some $q \in \mathbb{N}$, let $B \in \mathbb{C}^{nq \times nq}$ be such that for any $i, j \in [n]$ and $\alpha, \beta \in [q]$, $B_{(i,\alpha),(j,\beta)} = 0$ if $A_{ij} = 0$, and $|B_{(i,\alpha),(j,\beta)}| \leq A_{ij}$. If $\text{rank}_{\geq \tau}(A) \leq s$, then*

$$\text{rank}_{\geq q\sqrt{\tau(1-\sigma)} + \sigma}(B) \leq \frac{s}{\sigma^2}.$$

In particular, for any $\varepsilon > 0$, setting $\sigma = \tau = \varepsilon^2$,

$$\text{rank}_{\geq 2q\varepsilon}(B) \leq \frac{\text{rank}_{\geq \varepsilon^2}(A)}{\varepsilon^4}.$$

Proof. This immediately follows on instantiating [Theorem 4.1](#) with $\frac{1}{q}A \otimes \mathbf{1}_q \mathbf{1}_q^\top$ and $\frac{1}{q}B$. \blacksquare

Given this final ingredient, we may prove our result [Theorem 1.4](#) on near-linear time algorithms for general 2CSPs on bounded threshold rank graphs, restated for convenience.

Theorem 1.4. *Let G be a graph on $[n]$ with m edges, with adjacency matrix A and degree diagonal D . For each edge ij in G , let $\varphi_{ij} : [q]^2 \rightarrow \{0, 1\}$ be a predicate on alphabet size q that is not identically 0. Also let $\text{OPT} = \max_{x \in [q]^n} \Phi(x) := \max_{x \in [q]^n} \sum_{ij \in E(G)} \varphi_{ij}(x_i, x_j)$. Set $k = \text{rank}_{\geq \varepsilon^2}(D^{-1/2}AD^{-1/2})$. In time $(\frac{1}{\varepsilon})^{O(k/\varepsilon^4)} \cdot \tilde{O}(n + m) \cdot \text{poly}(q)$, one can find a random assignment \hat{x} with $\mathbf{E}\Phi(\hat{x}) \geq \text{OPT} - O(\varepsilon qm)$.*

Proof. Let $B \in \mathbb{R}^{nq \times nq}$ be the normalized adjacency matrix of the label-extended graph: For $i, j \in [n]$ and $\alpha, \beta \in [q]$, $B_{(i,\alpha),(j,\beta)} = (D^{-1/2}AD^{-1/2})_{ij}$ if $\{i, j\}$ is an edge and $\varphi_{ij}(\alpha, \beta) = 1$, and is 0 otherwise. Now instantiate [Theorem 3.3](#) with $M = \frac{1}{q}B$, the scaled degree diagonal $E = qD \otimes \text{Id}_q$, and the error parameter as 2ε . To conclude, the 2ε -threshold rank of $\frac{1}{q}B$ can be bounded in terms of the ε^2 -threshold rank of A using [Corollary 4.2](#). \blacksquare

[[BHSV25](#)] proves [Theorem 4.1](#) in the case that B is $-A$. We show that with minor changes the same proof allows B to have complex entries. The only change needed to the argument which proves [[BHSV25](#), Corollary 4.2], is the following lemma which substitutes for [[BHSV25](#), Lemma 4.4].

Lemma 4.3 (Adapted from [[BHSV25](#), Lemma 4.4]). *Suppose $A \in \mathbb{R}_{\geq 0}^{n \times n}$ is symmetric with $\|A\| \leq 1$. Let $B \in \mathbb{R}^{n \times n}$ be symmetric with $|B_{ij}| \leq A_{ij}$ for all i, j : in particular, $B_{ij} = 0$ if $A_{ij} = 0$. Suppose that $\text{rank}_{\geq \lambda}(B) \geq t$ for some $\lambda \geq 0$ and t a positive integer. Then there exists $V \in \mathbb{R}^{t^2 \times n}$ such that*

$$\langle A, V^\top V \rangle \geq \lambda^2, \quad \|V^\top V\|_F^2 \leq \frac{1}{t}, \quad \text{Tr}(V^\top V) = 1.$$

Proof, adapted from [[BHSV25](#)]. Let $\{u_1, \dots, u_t\}$ be t orthonormal eigenvectors of B whose corresponding eigenvalues are at least λ . Let $U \in \mathbb{R}^{n \times t}$ have s th column equal to u_s / \sqrt{t} . We record for later that

$$\begin{aligned} \langle B, UU^\top \rangle &= \sum_{s \leq t} \frac{1}{t} \langle B, u_s u_s^\top \rangle \geq \lambda, \\ \|UU^\top\|_F^2 &= \|U^\top U\|_F^2 = \frac{1}{t}, \\ \text{Tr}(UU^\top) &= \text{Tr}(U^\top U) = 1. \end{aligned}$$

We denote the i -th row of U by w_i , and let $v_i = w_i^{\otimes 2} / \|w_i\|$ if $w_i \neq 0$ and $v_i = 0$ otherwise. Let V be the matrix whose i -th column is v_i . We will show that V satisfies the conclusions of the lemma.

For the first conclusion, note that

$$\left(\sum_{i,j} B_{ij} \langle w_i, w_j \rangle \right)^2 = \left(\sum_{i,j} \frac{\sqrt{A_{ij}} \cdot \langle w_i, w_j \rangle}{\sqrt{\|w_i\| \|w_j\|}} \cdot \frac{B_{ij}}{\sqrt{A_{ij}}} \sqrt{\|w_i\| \|w_j\|} \right)^2$$

$$\begin{aligned}
&\leq \left(\sum_{ij} A_{ij} \frac{\langle w_i, w_j \rangle^2}{\|w_i\| \|w_j\|} \right) \cdot \left(\sum_{i,j} \frac{|B_{ij}|^2}{A_{ij}} \|w_i\| \|w_j\| \right) \\
&\leq \left(\sum_{ij} A_{ij} \frac{\langle w_i, w_j \rangle^2}{\|w_i\| \|w_j\|} \right) \cdot \left(\sum_{i,j} A_{ij} \|w_i\| \|w_j\| \right) \\
&\leq \left(\sum_{ij} A_{ij} \frac{\langle w_i, w_j \rangle^2}{\|w_i\| \|w_j\|} \right) \text{Tr}(UU^T)
\end{aligned}$$

where for the last line we used that $\|A\| \leq 1$. This gives the conclusion that $\langle A, VV^\top \rangle \geq \lambda^2$. The remaining conclusions follow exactly the arguments in [BHSV25], Lemma 4.4. \blacksquare

References

- [ABS15] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *Journal of the ACM (JACM)*, 62(5):1–25, 2015. [1](#), [8](#)
- [ACOH⁺10] Noga Alon, Amin Coja-Oghlan, Hiệp Hàn, Mihyun Kang, Vojtěch Rödl, and Mathias Schacht. Quasi-randomness and algorithmic regularity for graphs with general degree distributions. *SIAM Journal on Computing*, 39(6):2336–2362, 2010. [8](#)
- [AdIVKK02] Noga Alon, W. Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of MAX-CSP problems. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 232–239, New York, NY, USA, 2002. Association for Computing Machinery. [8](#)
- [AJT19] Vedat Alev, Fernando Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. pages 180–201, 11 2019. [1](#)
- [AK16] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM (JACM)*, 63(2):1–35, 2016. [3](#)
- [AKK95] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293, 1995. [1](#)
- [BHSV25] Rares-Darius Buhai, Yiding Hua, David Steurer, and Andor Vári-Kakas. Finding Colorings in One-Sided Expanders. *arXiv preprint arXiv:2508.02825*, 2025. [2](#), [7](#), [14](#), [15](#), [16](#)
- [BMZ05] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005. [1](#)
- [BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding Semidefinite Programming Hierarchies via Global Correlation. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 472–481. IEEE, 2011. [1](#), [2](#), [4](#), [7](#), [14](#)
- [COCF10] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010. [8](#)
- [FDLV96] W Fernandez De La Vega. Max-cut has a randomized approximation scheme in dense graphs. *Random Structures & Algorithms*, 8(3):187–198, 1996. [1](#)
- [FK99] Alan Frieze and Ravindran Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19:175–220, 02 1999. [8](#)
- [FLP15] Dimitris Fotakis, Michael Lampis, and Vangelis Th Paschos. Sub-exponential Approximation Schemes for CSPs: from Dense to Almost Sparse. *arXiv preprint arXiv:1507.04391*, 2015. [1](#)

- [GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981. [9](#)
- [GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre Hierarchy, Higher Eigenvalues, and Approximation Schemes for Graph Partitioning and Quadratic Integer Programming with PSD Objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. [4](#), [7](#)
- [GS12] Venkatesan Guruswami and Ali Kemal Sinop. Faster SDP hierarchy solvers for local rounding algorithms. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 197–206. IEEE, 2012. [4](#), [7](#)
- [Hsi25] Jun-Ting Hsieh. Coloring 3-Colorable Graphs with Low Threshold Rank, 2025. [8](#)
- [JKR19] Vishesh Jain, Frederic Koehler, and Andrej Risteski. Mean-field approximation, convex hierarchies, and the optimality of correlation rounding: a unified perspective. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, page 1226–1236, New York, NY, USA, 2019. Association for Computing Machinery. [1](#)
- [JQST20] Fernando Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique Decoding of Explicit -balanced Codes Near the Gilbert-Varshamov Bound. pages 434–445, 11 2020. [1](#), [3](#)
- [JST21] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of Ta-Shma’s codes via splittable regularity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021*, page 1527–1536, New York, NY, USA, 2021. Association for Computing Machinery. [1](#), [4](#), [8](#)
- [Kol10] Alexandra Kolla. Spectral algorithms for unique games. In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity, CCC ’10*, page 122–130, USA, 2010. IEEE Computer Society. [5](#), [7](#), [8](#)
- [Kum92] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI magazine*, 13(1):32–32, 1992. [1](#)
- [LGT14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6), December 2014. [2](#), [5](#)
- [Liu15] Shiping Liu. Multi-way dual cheeger constants and spectral bounds of graphs. *Advances in Mathematics*, 268:306–338, 2015. [5](#), [8](#)
- [MS08] Claire Mathieu and Warren Schudy. Yet Another Algorithm for Dense Max Cut: Go Greedy. In *SODA*, volume 8, pages 176–182, 2008. [1](#)

- [OGT13] Shayan Oveis Gharan and Luca Trevisan. A new regularity lemma and faster approximation algorithms for low threshold rank graphs. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 303–316. Springer, 2013. [1](#), [3](#), [4](#), [8](#)
- [Par00] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000. [9](#)
- [Ris16] Andrej Risteski. How to calculate partition functions using convex programming hierarchies: provable bounds for variational methods. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1402–1416, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. [1](#)
- [RS09] Prasad Raghavendra and David Steurer. How to Round Any CSP. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS '09*, page 586–594, USA, 2009. IEEE Computer Society. [12](#)
- [Sho87] Naum Zuselevich Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics*, 23(5):695–700, 1987. [9](#)
- [Ste10a] David Steurer. Fast sdp algorithms for constraint satisfaction problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 684–697. SIAM, 2010. [2](#), [3](#), [7](#), [12](#), [13](#)
- [Ste10b] David Steurer. Subexponential algorithms for d-to-1 two-prover games and for certifying almost perfect expansion. *Available at the author's website*, 1:2–1, 2010. [5](#)
- [Tre12] Luca Trevisan. Max cut and the smallest eigenvalue. *SIAM Journal on Computing*, 41(6):1769–1786, 2012. [5](#), [8](#)
- [Yar14] Grigory Yaroslavtsev. Going for speed: Sublinear algorithms for dense r-CSPs. *arXiv preprint arXiv:1407.7887*, 2014. [5](#)
- [YZ14] Yuichi Yoshida and Yuan Zhou. Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 423–438, 2014. [1](#), [5](#)