Subexponential Algorithms for Unique Games and Related Problems

Sanjeev Arora* Boaz Barak† David Steurer‡

April 14, 2010

Abstract

We give subexponential time approximation algorithms for Unique Games and the Small-Set Expansion. Specifically, for some absolute constant c, we give:

1. An $\exp(kn^{\varepsilon})$-time algorithm that, given as input a k-alphabet unique game on n variables that has an assignment satisfying $1-\varepsilon^c$ fraction of its constraints, outputs an assignment satisfying $1-\varepsilon$ fraction of the constraints.
2. An $\exp(n^{\varepsilon}/\delta)$-time algorithm that, given as input an n-vertex regular graph that has a set S of δn vertices with edge expansion at most ε^c, outputs a set S' of at most δn vertices with edge expansion at most ε.

We also obtain a subexponential algorithm with improved approximation for Multi Cut, as well as subexponential algorithms with improved approximations to Max Cut, Sparsest Cut and Vertex Cover on some interesting subclasses of instances.

Khot's Unique Games Conjecture (UGC) states that it is **NP**-hard to achieve approximation guarantees such as ours for Unique Games. While our results stop short of refuting the UGC, they do suggest that Unique Games is significantly easier than **NP**-hard problems such as 3-Sat, Max 3-Lin, Label Cover and more, that are believed not to have a subexponential algorithm achieving a non-trivial approximation ratio.

The main component in our algorithms is a new result on graph decomposition that may have other applications. Namely we show that for every $\delta > 0$ and a regular n-vertex graph G, by changing at most δ fraction of G's edges, one can break G into disjoint parts so that the stochastic adjacency matrix of the induced graph on each part has at most n^{ε} eigenvalues larger than $1-\eta$ (where ε, η depend polynomially on δ). Our results are based on combining this decomposition with previous algorithms for Unique Games on graphs with few large eigenvalues (Kolla and Tulsiani 2007, Kolla 2010).

*Department of Computer Science and Center for Computational Intractability, Princeton University. Supported by NSF Grants CCF-0832797, 0830673, and 0528414.

†Department of Computer Science and Center for Computational Intractability, Princeton University boaz@cs.princeton.edu. Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, and the Packard and Sloan fellowships.

‡Department of Computer Science and Center for Computational Intractability, Princeton University. Supported by NSF Grants CCF-0832797, 0830673, and 0528414.

1 Introduction

Among the important open questions of computational complexity, Khot's Unique Games Conjecture (UGC) [Kho02] is one of the very few that looks like it could "go either way." The conjecture states that for a certain constraint satisfaction problem, called Unique Games, it is **NP**-hard to distinguish between instances that are *almost satisfiable*— at least $1-\varepsilon$ of the constraints can be satisfied— and *almost completely unsatisfiable*— at most ε of the constraints can be satisfied. (See Section 5 for a formal definition.)

A sequence of works have shown that this conjecture has several important implications [Kho02, KR08, KKMO07, MOO05, KV05, CKK+05, Aus07, Rag08, MNRS08, GMR08], in particular showing that for many important computational problems, the currently known approximation algorithms have *optimal* approximation ratios. Perhaps most strikingly, Raghavendra [Rag08] showed that the UGC, if true, implies that every constraint satisfaction problem (CSP) has an associated *sharp approximation threshold* τ: for every $\varepsilon > 0$ one can achieve a $\tau-\varepsilon$ approximation in polynomial (and in fact even quasilinear [Ste10]) time, but obtaining a $\tau+\varepsilon$ approximation is **NP**-hard. Thus the UGC certainly has profound implications. But of course, profound implications by themselves need not be any evidence for truth of the conjecture. The deeper reason for belief in UGC is that in trying to design algorithms for it using current techniques, such as semi-definite programs (SDPs), one seems to run into the same bottlenecks as for all the other problems alluded to above, and indeed there are results showing limitations of SDPs in solving Unique Games [KV05, RS09, KS09a]. Moreover, recently it was shown that solving unique games is at least as hard as some other hard-looking problem— the *small set expansion* problem [RS10]. Another reason one might believe the Unique Games problem is hard is that it shares a superficial similarity with the Label Cover problem, which is known to be **NP**-hard to approximate. Indeed a relation between Label Cover and approximating unique games in a different parameter regime is known [FR04]. However, our work gives more evidence that the two problems are in fact quite different.

In this work we give a subexponential algorithm for unique games as well as small set expansion, as explained in the next two theorems. (Sometimes "subexponential" is meant to refer to $\exp(n^{o(1)})$ time, which we do not obtain when ε is a fixed constant. If we did, that would disprove the UGC under the ETH assumption explained below.)

Theorem 1.1 (See Theorem 5.1)**.** *There is some absolute constant $\alpha > 0$ and an $\exp(kn^{\varepsilon^\alpha})$-time algorithm that given a $(1-\varepsilon)$-satisfiable unique game of alphabet size[1] k, outputs an assignment satisfying $1-\varepsilon^\alpha$ fraction of the constraints.*

Theorem 1.2 (See Theorem 2.1)**.** *There is some absolute constant $\alpha > 0$ and an $\exp(n^{\varepsilon^\alpha}/\delta)$-time algorithm that given $\varepsilon, \delta > 0$ and a graph that has a set of measure at most δ and edge expansion at most ε, finds a set of measure at most δ and edge expansion at most ε^α.*

(Our results for small set expansion are slightly better quantitatively; see Theorem 2.1 for more details.) In fact, our algorithm for the unique games problem is obtained by extending the algorithm for the small set expansion problem, thus giving more evidence for the connection between these two problems.

What do these results imply for the status of the UGC? In a nutshell, they still don't rule out the UGC, but imply that (1) unique-game hardness results cannot be used to establish full exponential hardness of a computational problem regardless of the truth of the UGC, and (2) even if the UGC is true then (assuming 3-Sat has fully exponential complexity) the corresponding reduction from 3-Sat to Unique Games would have to run in $n^{1/\varepsilon^{0.01}}$ time, where ε is the completeness parameter of the unique games instance; in particular the UGC cannot be proved via a gadget reduction from Label Cover of the type pioneered by Hastad [Hås97].

[1] The *alphabet size* of a unique game is the number of symbols that each variable can be assigned. In the context of the UGC one can think of ε as some arbitrarily small constant, and k as some arbitrarily large constant depending on ε.

Thus unique games are qualitatively different from many **NP**-complete problems, which seem to require fully exponential time, as pointed out by Stearns and Hunt [SHI90] and Impagliazzo, Paturi and Zane [IPZ01]. The latter paper formulated the *Exponential Time Hypothesis* (ETH) —there is no $\exp(o(n))$ algorithm for solving n-variable 3-SAT—and showed that it implies that many computational problems such as CLIQUE, k-COLORABILITY,and VERTEX COVER require $2^{\Omega(n)}$ time as well. (n here refers not to the input size but the size of the solution when represented as a string.)

In fact, there are very few problems whose complexity is known to be subexponential but believed not to be polynomial— two famous examples are FACTORING and GRAPH ISOMORPHISM problems, which can be solved in time roughly $\exp(n^{1/3})$ [LL93] and $\exp(\sqrt{n}\log n)$ [Luk82] respectively[2]. Because of this paucity of counterexamples, researchers' intuition has been that "natural" problems exhibit a dichotomy —they are either in **P** or require fully exponential time (ie have essentially no nontrivial algorithms). For example the *algebraic dichotomy conjecture* of Bulatov, Jeavons and Krokhin [BJK00] (see also [KS09b]) says that under the ETH *every* constraint satisfaction problem is either in **P** or requires $2^{\Omega(n)}$ time. *Fixed parameter intractability* also tries to formalize the same phenomenon in another way.

Accumulating evidence in recent years suggested that a similar dichotomy might hold for *approximation*. To give an example, it is now known (due to efficient-PCP constructions, the last one by Moshkovitz and Raz [MR08]) that ETH implies that achieving $7/8+\varepsilon$-approximation to MAX 3-SAT requires $2^{n^{1-o(1)}}$ time for every fixed $\varepsilon > 0$, and similar statements are true for MAX-3LIN, and LABEL COVER. Thus it would be natural to interpret the spate of recent UGC-hardness results, especially Raghavendra's result for all CSPs, as suggesting that the same is true for many natural classes of **NP**-hard optimization problems such as CSPs: there are no approximation algorithms for these problems that run in subexponential time but achieve a better approximation ratio than current poly-time algorithms. Our results show that this interpretation would be incorrect and in fact is inconsistent with the UGC since UNIQUE GAMES itself— an important example of a constraint satisfaction problem in Raghavendra's class— has a subexponential time approximation algorithm that beats the poly-time algorithms if the UGC is true. Similarly our result also refutes the **NP**-hardness of variants of the UGC, such as those considered by Chawla et al [CKK+05], where the completeness parameter ε is a function tending to 0 with the input length. (Curiously, our subexponential algorithm really depends on completeness parameter being close to 1; the result of Feige and Reichman [FR04] mentioned above rules out under the ETH a subexponential approximation algorithm for games with completeness bounded away from 1.)

While (with the exception of the MULTI CUT problem) our ideas do not yet apply to problems "downstream of unique games" (e.g., 0.87-approximation to MAX CUT), we do indicate in Section 6 how to use them to get better algorithms on subfamilies of interesting instances for these problems.

1.1 Comparison with prior algorithms for unique games

Several works have given algorithms for approximating unique games. Most of these can be broadly divided into two categories: (1) polynomial-time algorithms giving relaxed notions of approximation (i.e., deteriorating as the alphabet size grows) for *all* instances [Kho02, Tre05, GT06, CMM06a, CMM06b] and (2) polynomial-time algorithms for *certain families* of instances such games whose constraint graphs are random graphs, graphs with expansion properties, and random geometric graphs [AKK+08, KT07, BHHS10, AIMS10, Kol10]. An exception is the recent work of Arora, Impagliazzo, Matthews, and Steurer [AIMS10] that gave an $\exp(2^{-\Omega(1/\varepsilon)}n)$ algorithm for unique games that are $1-\varepsilon$ satisfiable.

[2]Using padding one can show **NP**-hard problems with these property as well. A more interesting example (pointed out to us by Russell Impagliazzo) is subset sum on n integers each of $\sqrt{n}$ bits, which is **NP**-hard, but has an $\exp(\sqrt{n})$ time algorithm. Another example (pointed out to us by Anupam Gupta) is obtaining a $\log^{1.99} n$ approximation to the group Steiner tree problem. This was shown to be **NP**-hard via a *quasipolynomial* time reduction in [HK03], but to be in $\exp(n^{o(1)})$ time in [CP05], hence showing that the blowup in the reduction size is *inherent* (see discussion in [CP05, §4.3]).

Compared to papers from the first category, our algorithms run in subexponential as opposed to polynomial time, but give an approximation guarantee that is independent of the alphabet size. At the moment the constant α Theorem 1.1 is about $1/6$, and although it could perhaps be optimized further, there are some obstacles in making it smaller than $1/2$, which means that for very small alphabet size our approximation guarantee will be worse than that of [CMM06a], that gave an algorithm that on input a k-alphabet $(1-\varepsilon)$-satisfiable unique game, outputs a solution satisfying $1 - O(\sqrt{\varepsilon \log k})$ fraction of the constraints.

1.2 Overview: Threshold rank and graph decompositions

Our basic approach for the unique games algorithm is divide and conquer (similarly to Arora et al [AIMS10]): Partition the constraint graph of the Unique Games instance into disjoint blocks, throw out all constraints corresponding to edges that go between blocks, and solve independently within each block. However, the underlying "divide" step involves a new notion of graph decomposition, which we now explain.

Consider the adjacency matrix of an undirected regular graph G, whose every row/column is scaled to 1. (In other words, a *symmetric stochastic matrix*.) Our algorithm will use the fact that graphs with only a few eigenvalues close to 1 are "simple" because exhaustive enumeration in the subspace spanned by the corresponding eigenvalues will quickly give a good-enough solution, as explained below. Thus "complex" graphs are those with many eigenvalues close to 1. The core of our result is a new way of partitioning *every* graph into parts that are "simple." This decomposition result seems different from existing notions of partitions such as *Szemeredi's regularity lemma* [Sze76], *low-width cut decomposition of matrices* [FK99], *low-diameter decompositions* [LS93] and *padded decompositions* [GKL03]. The first two of the above notions really only apply to dense or pseudo-dense graphs, not all graphs. The latter two apply to all graphs but involve a "penalty term" of $O(\log n)$ that is too expensive in our setting, as explained in the paragraph after Theorem 1.4.

For $\tau \in [0,1)$ let the *τ-threshold rank* of G, denoted $\mathrm{rank}_\tau(G)$, be the number (with multiplicities) of eigenvalues λ of G satisfying $|\lambda| > \tau$. Thus $\mathrm{rank}_0(G)$ coincides with the usual rank of the matrix G, i.e., number of non-zero eigenvalues. We will usually be interested in τ close to 1, say 0.9. The higher the parameter $\mathrm{rank}_\tau(G)$, the "more complex" G is for us. Unlike many existing notions of "rank" or "complexity", $\mathrm{rank}_\tau(G)$ is small –actually, 1 —for a random graph, and more generally is 1 for any good expander. This should not be viewed as a bug in the definition: after all, expander graphs and random graphs are easy instances for problems such as unique games and small set expansion [AKK+08]. In fact, very recently Kolla [Kol10], building on [KT07], generalized this result to show an algorithm for unique games that runs in time exponential in the threshold rank of the corresponding constraint graph (assuming a certain bound on the ℓ_∞ norm of the eigenvectors).[3] A key step in our algorithm uses a very simple version of the key step in [Kol10, KT07], see Section 2.1.

Relating threshold rank and small-set expansion. The basic result underlying our graph decomposition algorithm is the following inequality that relates the threshold rank and small set expansion:

Theorem 1.3 (Rank/expansion tradeoff, see Theorem 2.3)**.** *If G is an n vertex regular graph in which every set S of at most s vertices has edge expansion at least* 0.1 *(i.e., at least* 0.1 *fraction of S's edges go to $[n]\setminus S$),*

[3] Specifically, [KT07] gave an algorithm that finds a satisfying assignment in time exponential in the threshold rank of the *label extended graph* of the unique game (see Section 5) and used it to obtain a polynomial time algorithm for unique games on expanders. [Kol10] showed how one can translate in certain cases bounds on the threshold rank of the constraint graph into bounds on the threshold rank of the label extended graph, hence allowing to use this algorithm in this more general setting. In this work we observe a more general, though quantitatively worse, relation between the threshold ranks of the label-extended and constraint graphs, see Corollary 5.3.

then

$$\mathrm{rank}_{1-\varepsilon}(G) \cdot s \leqslant n^{1+O(\varepsilon)} .$$

Furthermore, there is a polynomial-time algorithm that given any graph G will recover a set of size $n^{1+O(\varepsilon)}/\mathrm{rank}_{1-\varepsilon}(G)$ with edge expansion at most 0.1.

This result can be seen as a generalization of Cheeger's inequality [Che70, Dod84, AM85, Alo86]. The usual Cheeger's inequality would yield a nonexpanding set in the graph if there is even a single eigenvalue close to 1, but this set might be as large as half the vertices, while we obtain a set that (up to $n^{O(\varepsilon)}$ slackness) of measure inversely proportional to the number of large eigenvalues. Theorem 1.3 directly implies a simple "win-win" algorithm for the small set expansion problem. Either the $(1-\varepsilon)$-threshold rank is larger than $n^{c\varepsilon}$ for some large constant c, in which case we can find a very small (say of size less than $n^{1-\varepsilon}$) non-expanding set in polynomial time. Or, in the spirit of [KT07, Kol10], we can do in $\exp(n^{O(\varepsilon)}$-time a brute force enumeration over the span of the eigenvectors with eigenvalues larger than $1-\varepsilon$, and we are guaranteed to find if some non-expanding set S exists in the graph then we will recover S (up to a small error) via this enumeration, see Theorem 2.2.

Threshold-rank decomposition. By applying Theorem 1.3 repeatedly and recursively, we obtain our decomposition result:

Theorem 1.4 (Threshold-rank decomposition theorem, see Theorem 4.1)**.** *There is a constant c and a polynomial-time algorithm that given an n vertex regular graph G and $\varepsilon > 0$, partitions the vertices of G to sets $A_1, \ldots, A_q$ such that the induced[4] graph G_i on A_i satisfies $\mathrm{rank}_{1-\varepsilon^c}(G_i) \leqslant n^{\varepsilon}$ and at most a ε fraction of G's edges have their endpoints in different sets of this partition.*

Key to this decomposition is the advantage Theorem 1.3 has over Cheeger's inequality. Since every application of Cheeger's Inequality might leave us with half the vertices, one generally needs $\Omega(\log n)$ recursion depth to get a partition where each block has, say, size $\sqrt{n}$. This could end up removing all the edges unless $\varepsilon = O(1/\log n)$. In contrast, using Theorem 1.3 (or rather its more precise variant Theorem 2.3) we can get to such a partition with using a constant (depending on ε) depth of recursion.

Our unique games algorithm is obtained from Theorem 1.4 as follows. Given a unique games instance, we apply Theorem 1.4 to partition it (after removing a small fraction of the constraints) into disjoint parts each having small rank. We then look at the *label extended graph* corresponding to every part. (This is the graph that contains a "cloud" of k vertices for every variable of a k-alphabet unique game, and there is a matching between pairs of clouds according to the corresponding permutation, see Section 5.) We use the previously known observation that a satisfying assignment corresponds to a non-expanding set in the label extended graph, and combine it with a new observation (Lemma 5.2) relating the threshold rank of the label extended graph and the corresponding constraint graph. The result then follows by using the enumeration method over the top eigenspace to recover (up to some small noise) the satisfying assignment in every part.

Proof of the rank/expansion relation. Now we give some intuition behind Theorem 1.3, which underlies all this. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the graph's eigenvalues. Let us pick $\tau = 1-\eta$ for a small enough η and suppose $m = \mathrm{rank}_\tau(G)$. Then (the $2k$-th power of) the *Schatten* $2k$*-norm* $\mathrm{Tr}(G^{2k}) = \sum_{i \leqslant n} |\lambda_i|^{2k}$ is at least $m(1-\eta)^{2k}$. On the other hand, $\mathrm{Tr}(G^{2k})$ is also equal to $\sum_{i \leqslant n} \|G^k e_i\|_2^2$, where e_i is the unit vector whose only nonzero coordinate is the i-th. But $\|G^k e_i\|_2^2$ simply expresses the collision probability of a k-step random walk that starts from i. Then we can use a "local" version of Cheeger's inequality (in this form due to Dimitriou and Impagliazzo [DI98]), which shows that if all small sets expand a lot, then the collision probability of the k-step random walk decays very fast with k. We conclude that if all small sets expand a lot, then the

[4]The notion of "induced graph" we use involves "regularizing" the graph via self-loops, see Section 4 for the precise definition.

expression in $\|G^k e_i\|_2^2$ must be small, which yields an upper bound on $m(1-\eta)^{2k}$, and hence on the threshold rank m.

A related bound (in the other direction) connecting Schatten norms to small-set expansion was shown by Naor [Nao04], who used the Schatten norm to certify small-set expansion of Abelian Cayley graphs (or more generally, graphs with ℓ_∞ bounded eigenvectors).

1.3 Organization of the paper

The main ideas of this work appear in the simplest form in the subexpontial algorithm for Small-Set Expansion that is described in Section 2. The main component used is Theorem 2.3, showing that small set expanders must have low threshold rank. This theorem is proven in Section 3.

Section 4 contains our decomposition theorem, which is used in our algorithm for unique games appearing in Section 5. We sketch some partial results for other computational problems in Section 6. In Section 7 we show that *hypercontractive graphs*, that appear in many of the known integrality gap examples, have much smaller threshold rank than the bound guaranteed by Theorem 2.3, and use this to show a quasipolynomial time algorithm for efficiently certifying that a hypercontractive graph is a small-set expander.

1.4 Notation

Throughout this paper we restrict our attention to *regular* undirected graphs only, though we allow self loops and weighted edges (as long as the sum of weights on edges touching every vertex is the same). In our context this is without loss of generality, because the computational problems we are interested reduce easily to the regular case (see Appendix A.1).[5] We consider graphs with vertex set $V = [n] = \{1..n\}$, and use G to denote both the graph and its stochastic walk matrix. We use $(i,j) \sim G$ to denote the distribution obtained by choosing a random edge of G (i.e., obtaining (i,j) with probability $G_{i,j}/n$). We define the measure of a subset $S \subseteq [n]$, denoted $\mu(S)$, to be $|S|/n$. For $S, T \subseteq [n]$ we denote $G(S,T) \stackrel{def}{=} \frac{1}{n}\sum_{i\in S, j\in T} G_{i,j} = \Pr_{(i,j)\sim G}[i \in S, j \in T]$. The *expansion* of a subset S of the vertices of a graph G, denoted $\Phi_G(S)$, is defined as $G(S,\overline{S})/\mu(S) = \Pr_{(i,j)\in G}[j \in \overline{S} | i \in S]$, where $\overline{S} = [n] \setminus S$. We will often drop the subscript G and use only $\Phi(S)$ when the graph is clear from the context.

For $x, y \in \mathbb{R}^n$ we let $\langle x, y\rangle = \mathsf{E}_{i\in[n]}[x_i y_i]$. We define the corresponding 2-norm and 1-norm as $\|x\| = \sqrt{\langle x, x\rangle}$ and $\|x\|_1 = \mathsf{E}_{i\in[n]}[|x_i|]$. Note that $\Phi(S) = 1 - \langle \chi_S, G\chi_S\rangle$ where χ_S is the normalized characteristic vector of S, that is $\chi_S(i) = \sqrt{n/|S|}$ if $i \in S$ and $\chi_S(i) = 0$ otherwise. Indeed, $\langle \chi_S, G\chi_S\rangle = (n/|S|)(1/n)\sum_{i\in S}\sum_{j\in S} G_{i,j}$.

We say that $f(n) = \exp(g(n))$ if there is some constant c such that $f(n) \leqslant 2^{c\cdot g(n)}$ for every sufficiently large n. Throughout this paper, the implicit constants used in $O(\cdot)$ notation are absolute constants, independent of any other parameters.

2 An algorithm for small set expansion

In this section we give a subexponential algorithm for small set expansion. Specifically we prove the following theorem:

Theorem 2.1 (Subexponential algorithm for small-set expansion)**.** *For every $\beta \in (0,1)$, $\varepsilon > 0$, and $\delta > 0$, there is an $\exp\left(n^{O(\varepsilon^{1-\beta})}\right)\mathrm{poly}(n)$-time algorithm that on input a regular graph G with n vertices that has a set S of at most δn vertices satisfying $\Phi(S) \leqslant \varepsilon$, finds a set S' of at most δn vertices satisfying $\Phi(S) \leqslant O(\varepsilon^{\beta/3})$.*

[5] For simplicity, we *define* the small set expansion problem only for regular graphs, although one can easily generalize the definition and our results to non-regular graphs, assuming the measure of each vertex is weighted by its degree.

Note that by setting $\beta = O(1/\log(1/\varepsilon))$ we can get an $\exp(n^{O(\varepsilon)})$-time algorithm that given a graph with a small set of expansion at most ε, finds a small set of expansion at most, say, 0.01.

We prove Theorem 2.1 by combining two methods. First we show that if the input graph has at most m large eigenvalues then one can find the non expanding set (if it exists) in time $\exp(m)$. Then we show that if a graph has many eigenvalues that are fairly large then it *must* contain a small set with poor expansion, and in fact there is an efficient way to find such a set. The algorithm is obtained by applying one of these methods to the input graph depending on the number of eigenvalues larger than $1-\eta$ (for some suitably chosen threshold η).

2.1 Enumerating non-expanding sets in low-rank graphs

We start by showing that the search for a non-expanding set in a graph can be greatly speeded up if it has only few large eigenvalues.

Theorem 2.2 (Eigenspace enumeration)**.** *There is an* $\exp(\mathrm{rank}_{1-\eta}(G))\,\mathrm{poly}(n)$*-time algorithm that given* $\varepsilon > 0$ *and a graph* G *containing a set* S *with* $\Phi(S) \leqslant \varepsilon$*, outputs a sequence of sets, one of which has symmetric difference at most* $8(\varepsilon/\eta)|S|$ *with the set* S*.*

In particular for $\varepsilon < 0.01$ and $\eta = 1/2$ the algorithm will output a list of sets containing a set S' such that $|S'| \leqslant 1.1|S|$ and $\Phi(S') \leqslant 13\varepsilon$.

Proof. Let $\delta = \mu(S) = |S|/n$, and let χ_S be the normalized indicator vector of S, that is $\chi_S(i) = 1/\sqrt{\delta}$ if $i \in S$ and $\chi_S(i) = 0$ otherwise. Let $U \subseteq \mathbb{R}^V$ be the span of the eigenvectors with eigenvalue greater than $1-\eta$. The dimension of U is equal to $m = \mathrm{rank}_{1-\eta}(G)$. Suppose $\chi_S = \sqrt{1-\gamma}\cdot u + \sqrt{\gamma}\cdot u^\perp$, where $u \in U$ and $u^\perp$ is orthogonal to U (and hence $u^\perp$ is in the span of the eigenvectors with eigenvalue at most $1-\eta$). Both u and $u^\perp$ are unit vectors. Since $\Phi(S) \leqslant \varepsilon$, we have

$$\varepsilon \geqslant 1 - \langle \chi_S, G\chi_S\rangle = 1 - (1-\gamma)\langle u, Gu\rangle - \gamma\langle u^\perp, Gu^\perp\rangle \geqslant \gamma\cdot\eta\,,$$

where the last step uses $\langle u, Gu\rangle \leqslant 1$ and $\langle u^\perp, Gu^\perp\rangle \leqslant 1-\eta$. Hence, $\|\chi_S - u\|^2 = \gamma \leqslant \varepsilon/\eta$. If we enumerate over a $\sqrt{\varepsilon/\eta}$-net in the unit ball of the m-dimensional subspace U, then we will find a vector v satisfying $\|v - \chi_S\|_2^2 \leqslant 2\varepsilon/\eta$. The size of a $\sqrt{\varepsilon/\eta}$-net in U is at most $\exp(m\log(1/\varepsilon))$. Hence, at most $8\delta\varepsilon/\eta$ fraction of the coordinates of v differ from χ_S by more than $1/(2\sqrt{\delta})$. Let $S' = S'(v)$ be the set defined by setting $i \in S'$ if $v_i \geqslant 1/(2\sqrt{\delta})$ and $i \notin S'$ otherwise. Every coordinate i in the symmetric difference between S and S' corresponds to a coordinate in which v and χ_S differ by at least $1/(2\sqrt{\delta})$ and so the symmetric difference of S and S' has measure at most $8\varepsilon\delta/\eta$. □

This theorem is inspired by a recent result of Kolla [Kol10] (building on [KT07]) who showed an $\exp(\mathrm{rank}_{1-\varepsilon^{1/3}}(G))$ -time algorithm for unique games on Cayley graphs (or more generally graphs with ℓ_∞-bounded eigenvectors), where G is the constraint graph of the unique games. Interestingly, our analogous result for small set expansion is both much simpler and stronger (in the sense that it does not require additional properties of eigenvectors).

It is also instructive to compare Theorem 2.2 with Cheeger's inequality. The proof is basically a variant of the "easy direction" of Cheeger's inequality (showing that if $\lambda_2 \leqslant 1-\varepsilon$ then every set S of size at most $n/2$ satisfies $\Phi(S) \geqslant \varepsilon/2$). But the algorithm actually beats the rounding algorithm of the "hard direction". While the latter is only able to produce a set S with $\Phi(S) \leqslant O(\sqrt{\varepsilon})$ in a graph where $\lambda_2 \geqslant 1-\varepsilon$, the enumeration algorithm will find a set with $\Phi(S) \leqslant O(\varepsilon)$, and the difference only becomes stronger as the set size is smaller. Of course this is not so surprising as we basically do brute force enumeration. Still, at this point the algorithm beats the SDPs for small set expansion and unique games, as those basically have the integrality gap of Cheeger's inequality. What is perhaps more surprising is that, as we will see next, we can generally assume $\mathrm{rank}_{1-\eta}(G) \ll n$, since graphs violating this condition are "trivial" in some sense.

2.2 Finding small non-expanding sets in high-rank graphs

Our next step is to show that every graph with high threshold-rank contains a small non-expanding vertex set. Moreover such a set can be found efficiently, since it can be assumed to be a level set of a column in a power of the matrix $1/2 I + 1/2 G$. (A level set of a vector $x \in \mathbb{R}^V$ is a set of the form $\{i \in V \mid x_i \geq \tau\}$ for some threshold $\tau \in \mathbb{R}$.)

Theorem 2.3 (Rank bound for small-set expanders)**.** *Let G be a regular graph on n vertices such that $\mathrm{rank}_{1-\eta}(G) \geq n^{100\eta/\gamma}$. Then there exists a vertex set S of size at most $n^{1-\eta/\gamma}$ that satisfies $\Phi(S) \leq \sqrt{\gamma}$. Moreover, S is a level set of a column of $(\frac{1}{2} \cdot I + \frac{1}{2} \cdot G)^j$ for some $j \leq O(\log n)$.*

One can think of Theorem 2.3 as a generalization of the "difficult direction" of Cheeger's Inequality. The latter says that if $\mathrm{rank}_{1-\eta}(G) > 1$ then there exists a set S with $\mu(S) \leq 1/2$ and $\Phi(S) \leq O(\sqrt{\eta})$. Theorem 2.3 gives the same guarantee, but in addition the measure of the set S is inversely proportional to the threshold rank (i.e., number of large eigenvalues), assuming this rank is larger than $n^{\Omega(\eta)}$. We note that we have made no attempt to optimize the constants of this theorem, and in fact do not know if the constant 100 above cannot be replaced with $1 + o(1)$ (though such a strong bound, if true, will require a different proof).

We now combine Theorem 2.2 and Theorem 2.3 to obtain our subexponential algorithm for small set expansion, namely Theorem 2.1.

Theorem 2.1 (Subexponential algorithm for small-set expansion, restated)**.** *For every $\beta \in (0,1)$, $\varepsilon > 0$, and $\delta > 0$, there is an $\exp(n^{O(\varepsilon^{1-\beta})})\mathrm{poly}(n)$-time algorithm that on input a regular graph G with n vertices that has a set S of at most δn vertices satisfying $\Phi(S) \leq \varepsilon$, finds a set S' of at most δn vertices satisfying $\Phi(S) \leq O(\varepsilon^{\beta/3})$.*

Proof of Theorem 2.1. Set $\eta = \varepsilon^{1-\beta/3}$ and $\gamma = \varepsilon^{2\beta/3}$. If $\mathrm{rank}_{1-\eta}(G) \geq n^{100\eta/\gamma}$, then we can compute in polynomial time a set of size at most $n^{1-\eta/\gamma} \ll \delta n$ and expansion at most $10\sqrt{\gamma} = O(\varepsilon^{\beta/3})$ by Theorem 2.3. Otherwise, if $\mathrm{rank}_{1-\eta}(G) < n^{100\eta/\gamma}$, we can compute in time $\exp(n^{O(\eta/\gamma)}\log(1/\varepsilon)) = \exp(n^{O(\varepsilon^{1-\beta})})$ a set with measure at most $(1 + O(\varepsilon/\eta))\delta \leq 2\delta$ and expansion at most $O(\varepsilon/\eta) = O(\varepsilon^{\beta/3})$. □

3 Threshold-rank bounds for small-set expanders

In this section we prove Theorem 2.3. As mentioned above, Theorem 2.3 is a very natural analog of Cheeger's inequality, roughly saying that if a graph has m eigenvalues larger than $1 - \eta$, then not only can we find a set S of expansion at most $O(\sqrt{\eta})$ (as guaranteed by Cheeger's inequality), but in fact we can guarantee that S's measure is roughly $1/m$. This suggests the following proof strategy— use linear combinations of the m eigenvectors to arrive at a $1/m$-sparse vector v satisfying $\langle v, Gv\rangle \geq 1 - O(\eta)$, and then use the rounding procedure of Cheeger's inequality to obtain from v a set with the desired property. Unfortunately, this strategy completely fails— since in our context $m = o(n)$ (in fact $m = n^{O(\eta)}$) there is no reason to expect that there will exist a sparse vector in the subspace spanned by the first m eigenvectors.[6] So our proof works in an indirect way, via the trace formula $\mathrm{Tr}(G) = \sum_{i=1}^n \lambda_i$. In particular we know that for every k, we'll have $\mathrm{Tr}(G^k) = \sum_{i=1}^n \lambda_i^k \geq m(1-\eta)^k$, so an upper bound on $\mathrm{Tr}(G^k)$ will translate into an upper bound on m. If G is a stochastic matrix, then $\mathrm{Tr}(G^k)$ is equal to the sum over all $i \in [n]$, of the *collision probability* of the distribution $G_i^{k/2}$ defined as taking a $k/2$ step random walk from vertex i. Suppose for the sake of simplicity that this distribution was uniform over the $k/2$-radius ball in which case the collision probability will equal one over the size of the ball. If the graph had expansion at least ε for sets of size at

[6] We do expect however that a "typical" vector will have roughly m/n of its mass in this subspace, and our proof can be viewed as using the observation that this will hold for one of the standard basis vectors, which are of course the sparsest vectors possible.

most δ, we might expect this ball to have size $\min\{\delta n, (1+\varepsilon)^{k/2}\}$.[7] Thus, we'd expect $\mathrm{Tr}(G^k)$ to be at most n times $\max\{(1+\varepsilon)^{-k/2}, 1/(\delta n)\}$, which is roughly the bound we get. (See Theorem 3.1 below, we note that we do get an ε^2 term instead of ε in our bound, though this loss will not make a huge difference for our applications.)

We now state formally and prove our bound on $\mathrm{Tr}(G^k)$. Define the *k-Schatten norm* $S_k(M)$ of a symmetric matrix M with eigenvalues $\lambda_1, \ldots, \lambda_n$ to be $S_k(M)^k = \lambda_1^k + \ldots + \lambda_n^k$. (Note that by the trace formula $S_k(M)^k = \mathrm{Tr}(M^k)$.) We say a graph G is *lazy* if $G = 1/2 \cdot G' + 1/2 \cdot I$ for some regular graph G'. (In other words, G is lazy if $G \geqslant 1/2 \cdot I$ entry-wise.) For technical reasons, we will prove a Schatten norm bound only for lazy graphs. (This bound will suffice to prove Theorem 2.3 also for non-lazy regular graphs.)

Theorem 3.1 (Schatten norm bound)**.** *Let G be a lazy regular graph on n vertices. Suppose every vertex set S with $\mu(S) \leqslant \delta$ satisfies $\Phi(S) \geqslant \varepsilon$. Then, for all even $k \geqslant 2$, the k-Schatten norm of G satisfies*

$$S_k(G)^k \leqslant \max\left\{ n \cdot \left(1 - \varepsilon^2/32\right)^k, \tfrac{4}{\delta} \right\}.$$

Moreover, for any graph that does not satisfy the above bound, we can compute in polynomial time a vertex subset S with $\mu(S) \leqslant \delta$ and $\Phi(S) \leqslant \varepsilon$, where S is a level set of a column of G^j for some $j \leqslant k$.

Before proving Theorem 3.1, lets see how it implies Theorem 2.3

Theorem 2.3 (Rank bound for small-set expanders, restated)**.** *Let G be a regular graph on n vertices such that $\mathrm{rank}_{1-\eta}(G) \geqslant n^{100\eta/\gamma}$. Then there exists a vertex set S of size at most $n^{1-\eta/\gamma}$ that satisfies $\Phi(S) \leqslant \sqrt{\gamma}$. Moreover, S is a level set of a column of $(\frac{1}{2} \cdot I + \frac{1}{2} \cdot G)^j$ for some $j \leqslant O(\log n)$.*

Proof of Theorem 2.3 from Theorem 3.1. Let G, η, γ be as in the theorem, and let $m = \mathrm{rank}_{1-\eta}(G)$. (Note that we can assume $\eta < 100\gamma$ as otherwise the statement is trivial.) Set $G' = 1/2 I + 1/2 G$ to be the "lazy version" of G and note that (1) for every set S, $\Phi_{G'}(S) = \Phi(S)/2$ and (2) since every eigenvalue λ in G translates to an eigenvalue $1/2 + 1/2\lambda$ in G', $m = \mathrm{rank}_{1-\eta/2}(G')$. Now set k to be such that $(1-\gamma/64)^k = 1/n$ and $\delta = n^{-\eta/\gamma}$ and apply Theorem 3.1 to G', k with $\varepsilon = \sqrt{\gamma}/2$. We get that if $\Phi_{G'}(S) \leqslant \sqrt{\gamma}/2$ for every S of measure at most δ, then

$$m(1-\eta/2)^k \leqslant S_k(G')^k \leqslant 4/\delta = 4n^{\eta/\gamma}.$$

(Since the first term in the max expression is 1.) Now use $(1-\eta/2) \sim (1-\gamma/64)^{64\eta/(2\gamma)} \geqslant (1-\gamma/64)^{65\eta/\gamma}$ (in the range we care about) to argue that

$$m(1-\gamma/64)^{k65\eta/\gamma} \leqslant 4n^{\eta/\gamma},$$

but by our choice of k, we get

$$mn^{-65\eta/\gamma} \leqslant 4n^{\eta/\gamma},$$

or (assuming $n^{\eta/\gamma} \gg 4$)

$$m \leqslant n^{100\eta/\gamma}. \tag{3.1}$$

Moreover, if G' violates (3.1), we can find efficiently a level set S of a column G'^j that will satisfy $\mu(S) \leqslant \delta$ and $\Phi_{G'}(S) = \Phi_G(S)/2 \leqslant \sqrt{\gamma}/2$. □

[7] Note in this discussion we ignore the degree of the graph, and you might think we'd expect the ball to have size $(\varepsilon d)^{k/2}$ for small k, but it turns out our bounds are independent of the degree, and in fact one can think of the degree of an absolute constant much smaller than $1/\varepsilon$, in which case the growth does behave more similarly to $(1+\varepsilon)^{k/2}$.

A trace bound. The proof of Theorem 2.3 actually achieves a somewhat stronger statement. Define the $1-\eta$ *trace threshold rank* of G, denoted $\mathrm{rank}^*_{1-\eta}(G)$, to be the infimum over $k \in \mathbb{N}$ of $\mathrm{Tr}(G^{2k})/(1-\eta)^{2k}$. Clearly $\mathrm{rank}_{1-\eta}(G) \leqslant \mathrm{rank}^*_{1-\eta}(G)$, since $\mathrm{Tr}(G^{2k}) \geqslant \mathrm{rank}_{1-\eta}(G)(1-\eta)^{2k}$. Because our proof bounds the rank of G via the trace of the lazy graph $1/2I + 1/2G$, it actually achieves the following statement:

Theorem 3.2 (Trace rank bound for small-set expanders)**.** *Let G be a regular lazy graph on n vertices such that $\mathrm{rank}^*_{1-\eta}(G) \geqslant n^{100\eta/\gamma}$. Then there exists a vertex set S of size at most $n^{1-\eta/\gamma}$ that satisfies $\Phi(S) \leqslant \sqrt{\gamma}$. Moreover, S is a level set of a column of G^j for some $j \leqslant O(\log n)$.*

One can also show that the trace rank bound is not too far from the threshold rank, in the range of parameters of interest in this work:

Lemma 3.3. *For every $\delta, \eta \in (0,1)$, $\mathrm{rank}^*_{1-\delta\eta}(G) \leqslant \mathrm{rank}_{1-\eta}(G) n^{5\delta}$.*

Proof. For every k, one can see by the definition of rank^* and the formula $\mathrm{Tr}(G^{2k}) = \sum_{i=1}^n |\lambda_i|^{2k}$ that

$$\mathrm{rank}^*_{1-\delta\eta}(G)(1-\eta\delta)^{2k} \leqslant \mathrm{Tr}(G^{2k}) \leqslant \mathrm{rank}_{1-\eta}(G) + n(1-\eta)^{2k}$$

plugging in $k = \log n/\eta$ we get that $\mathrm{rank}^*_{1-\delta\eta}(G) n^{-4\delta} \leqslant \mathrm{rank}_{1-\eta}(G) + 1$. □

3.1 Proof of Theorem 3.1

In the following, we let G be a fixed lazy graph with vertex set $V = [n]$. (We use the assumption that G is lazy only for Lemma A.1). Recall that we identify G with its stochastic adjacency matrix. The proof of Theorem 3.1 is based on the relation of the following parameter to Schatten norms and the expansion of small sets,

$$\Lambda(\delta) \stackrel{\mathrm{def}}{=} \max_{x \in \Omega_\delta} \frac{\|Gx\|}{\|x\|}\,.$$

Here, the set $\Omega_\delta \subseteq \mathbb{R}^V$ is defined as

$$\Omega_\delta \stackrel{\mathrm{def}}{=} \left\{ x \in \mathbb{R}^V \mid 0 < \|x\|_1^2 \leqslant \delta \cdot \|x\|^2 \right\}.$$

By Cauchy–Schwarz, every vector with support of measure at most δ is contained in Ω_δ.

Since the spectral radius of G is at most 1, the parameter $\Lambda(\delta)$ is upper bounded by 1 for all $\delta > 0$. The following lemma shows that if G is an expander for sets of measure at most δ, then $\Lambda(\delta)$ is bounded away from 1. (In fact, small-set expansion is equivalent to $\Lambda(\delta)$ being bounded away from 1. However, we only need one direction of this equivalence for the proof.)

Lemma 3.4. *Suppose $\Phi(S) \geqslant \varepsilon$ for all sets S of measure at most δ. Then,*

$$\Lambda(\delta/4) \leqslant 1 - \varepsilon^2/32\,.$$

Moreover, if $x \in \Omega_{\delta/4}$ is a unit vector such that $\|Gx\| > 1 - \varepsilon^2/32$, then there exists a level set S of x such that $\mu(S) \leqslant \delta$ and $\Phi(S) < \varepsilon$.

The proof of Lemma 3.4 combines a few standard techniques (Cheeger's inequality with Dirichlet boundary conditions and a truncation argument, e.g. see [GMT06, Chu07, AK09, RST10a]). Also, a variant of this lemma that is strong enough for our purposes was given by Dimitriou and Impagliazzo [DI98]. For completeness, we present a self-contained proof in Appendix A.2.

Next, we obtain a bound on Schatten norms in terms of the parameter $\Lambda(\delta)$. We need the following simple technical lemma, which almost follows immediately from the definition of $\Lambda(\delta)$.

Lemma 3.5. *For every $j \in \mathbb{N}$, $x \in \mathbb{R}^V$, and $\delta > 0$,*

$$\|G^j x\| \leqslant \max\left\{\Lambda(\delta)^j \cdot \|x\|\,,\ \tfrac{1}{\sqrt{\delta}} \cdot \|x\|_1\right\}. \tag{3.2}$$

Proof. Indeed, suppose that $\|G^j x\| \geqslant \|x\|_1/\sqrt{\delta}$. Then, since G is stochastic and hence $\|Gy\|_2 \leqslant \|y\|_2$ and $\|Gy\|_1 \leqslant \|y\|_1$ for all y,

$$\|x\|_2 \geqslant \|Gx\|_2 \geqslant \cdots \geqslant \|G^{j-1}x\|_2 \geqslant \|G^j x\|_2 \geqslant \|x\|_1/\sqrt{\delta} \geqslant \|Gx\|_1/\sqrt{\delta} \geqslant \cdots \geqslant \|G^{j-1}x\|_1/\sqrt{\delta} \geqslant \|G^j x\|_1/\sqrt{\delta}\,.$$

Therefore, we see that $G^i x \in \Omega_\delta$ for all $i \in \{0, \ldots, j\}$, which implies that

$$\|G^j x\| \leqslant \Lambda(\delta)\|G^{j-1}x\| \leqslant \Lambda(\delta)^2\|G^{j-1}x\| \leqslant \ldots \leqslant \Lambda(\delta)^j\|x\|\,. \qquad \square$$

With this lemma, we can prove the following bound on Schatten norms in terms of the parameter $\Lambda(\delta)$.

Lemma 3.6. *For every even integer $k \geqslant 2$,*

$$S_k(G)^k \leqslant \max\left\{n \cdot \Lambda(\delta)^k, \tfrac{1}{\delta}\right\}.$$

Proof. Let $e_1, \ldots, e_n$ be the normalized standard basis vectors, that is, the j-th coordinate of e_i is equal to $\sqrt{n}$ if $i = j$ and equal to 0 otherwise. Note that $\|e_i\| = 1$ and $\|e_i\|_1 = 1/\sqrt{n}$. Using the identity $S_k(G)^k = \mathrm{Tr}(G^k)$, we obtain

$$S_k(G)^k = \mathrm{Tr}(G^k) = \sum_{i=1}^n \langle e_i, G^k e_i\rangle = \sum_{i=1}^n \langle G^{k/2}e_i, G^{k/2}e_i\rangle = \sum_{i=1}^n \|G^{k/2}e_i\|^2 \leqslant n \cdot \max\left\{\Lambda(\delta)^k, \tfrac{1}{n\delta}\right\},$$

where the last inequality uses Lemma 3.5. $\square$

Lemma 3.4 and Lemma 3.6 immediately imply Theorem 3.1 by noting that under the condition of the theorem, $\Lambda(\delta/4) \geqslant 1 - \varepsilon^2/32$, hence implying that $S_k(G)^k \leqslant \max\{n(1-\varepsilon^2/32)^k, 4/\delta\}$. Moreover, following the proof we see that if the condition is violated then we can get a set S with $|S| \leqslant \delta n$ and $\Phi(S) \leqslant \varepsilon$ by looking at a level set of the vector $G^j e_i$ for some $j \leqslant k$ and standard basis vector e_i.

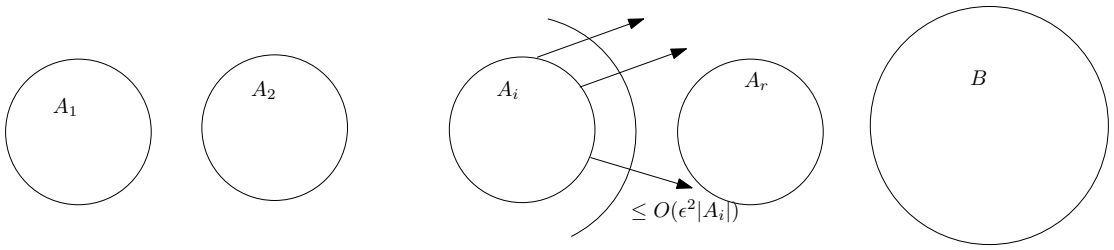
4 Low threshold-rank decomposition of graphs

In this section we obtain our main technical tool for extending our small set expansion algorithm to unique games. This is an algorithm to decompose a graph into parts that have low threshold rank.

We will use the following notation. For a graph G, a *partition* of the vertices $V = V(G)$ is a function $\chi : V \to \mathbb{N}$. We will not care about the numerical values of χ and so identify χ with the family of disjoint sets $\{\chi^{-1}(i)\}_{i \in \mathrm{Image}(\chi)}$. The size of the partition χ, denoted by $\mathrm{size}(\chi)$ is the number of sets/colors it contains. We define the *expansion* or *cost* of the partition, denoted $\Phi(\chi)$, to be the fraction of edges i, j for which $\chi(i) \neq \chi(j)$. If G is a d-regular graph and $U \subseteq V(G)$, we let $G[U]$ be the induced graph on U that is "regularized" by adding to every vertex sufficient number of weight half self loops to achieve degree d. Our decomposition result is the following:

Theorem 4.1 (Low threshold rank decomposition theorem)**.** *There is a polynomial time algorithm that on input a graph G and $\varepsilon > 0$, outputs a partition $\chi = (A_1, \ldots, A_q)$ of $V(G)$ such that $\Phi(\chi) \leqslant O(\varepsilon \log(1/\varepsilon))$ and for every $i \in [q]$,*

$$\mathrm{rank}_{1-\varepsilon^5}(G[A_i]) \leqslant n^{100\varepsilon}\,.$$



We prove Lemma 4.3 by repeatedly using Lemma 4.2 to remove from the graph sets $A_1, \ldots, A_r$ that are "somewhat small" (size $\leqslant n^{1-\varepsilon}$) until the remaining part B has $1-\varepsilon^5$ threshold rank at most $n^{100\varepsilon}$. To count the expansion of the partition we orient all edges to the right in the figure and charge each edge $\overrightarrow{u\,v}$ crossing the partition to the set A_i containing u. Lemma 4.2 guarantees that the weight of directed edges from A_i to $A_{i+1} \cup \cdots \cup B$ is at most $O(\varepsilon^2|A_i|)$. Theorem 4.1 is an immediate implication of Lemma 4.4, and the latter is proven by recursively applying Lemma 4.3 to each of the sets A_i up to $\varepsilon^{-1}\log(1/\varepsilon)$ times until the only non-expander parts remaining are "very small" (size $\leqslant n^{\varepsilon}$). The overall cost of the partition is $O(\varepsilon^2\varepsilon^{-1}\log(1/\varepsilon)) = O(\varepsilon\log(1/\varepsilon))$.

Figure 1: Overview of proof of Theorem 4.1.

4.1 Proof of the decomposition theorem

We now turn to proving Theorem 4.1. The proof is obtained by repeated applications of Theorem 2.2 with appropriately chosen parameters, as sketched in Figure 1, and so some readers may prefer to skip ahead to Section 5, showing our subexponential algorithms for unique games.

We start with some notation. Throughout the proof we'll fix the graph G (which we also think of as a stochastic matrix) on the set $V = [n]$ of vertices. If $U \subseteq V$, and $S \subseteq U$, then the *relative* expansion of S with respect to U, denoted by $\Phi_U(S)$, is defined as $\Pr_{(i,j)\sim G}[j \in U \setminus S | i \in S]$. Note that $\Phi_U(S)$ is equal to the expansion of S in the graph $G[U]$. (Recall that $G[U]$ is regularized by adding self-loops.) If $\chi = (S_1, \ldots, S_q)$ is a partition of V, then we define the relative expansion of χ with respect to U, denoted as $\Phi_U(\chi)$, to be $\sum_{i=1}^{q} \Phi_U(S_i \cap U)$. We say that χ *refines* a partition τ if $\chi(x) = \chi(y) \Rightarrow \tau(x) = \tau(y)$. We define the relative cost of χ with respect to τ, denoted $\Phi_\tau(\chi)$ to be the fraction of edges that are cut in χ but not in τ. That is,

$$\Phi_\tau(\chi) = \frac{1}{n} \sum_{\substack{i,j\in[n] \\ \chi(i)\neq\chi(j) \\ \tau(i)=\tau(j)}} G_{i,j}\,.$$

Note that if $\tau = (S_1, \ldots, S_q)$ then $\Phi_\tau(\chi) = \sum_{i=1}^{q} \mu(S_i)\Phi_{S_i}(\chi)$. It's not hard to verify that $\Phi(\chi) \leqslant \Phi(\tau) + \Phi_\chi(\tau)$ with equality if χ is a refinement of τ.

The proof will be obtained by a sequence of three lemmas, see Figure 1 for an overview. We start with the following instantiation of our small set expander algorithm:

Lemma 4.2. *Let G be an n vertex graph, and $\varepsilon > 0$. If $\mathrm{rank}_{1-\varepsilon^5}(G) \leqslant n^{100\varepsilon}$ then we can find in polynomial time a set $S \subseteq V(G)$ with $|S| \leqslant n^{1-\varepsilon}$ and $\Phi(S) \leqslant O(\varepsilon^2)$.*

Proof. Instantiate Theorem 2.3 with $\gamma = \varepsilon^4$ and $\eta = \varepsilon^5$. □

The precise powers of ε in Lemma 4.2 are not important. The main point is that the set S in the conclusion of the lemma satisfies $\log_n(1/\mu(S)) \gg \Phi(S)$ (which is the case because $\log_n(1/\mu(S)) = \varepsilon$ and $\phi(S) = O(\varepsilon^2)$).

Next, we apply Lemma 4.2 repeatedly to obtain a *partition* of the graph into sets that are either somewhat small, or are small set expanders.

Lemma 4.3. *There is a polynomial-time algorithm that given an n vertex graph G and $\varepsilon > 0$, outputs a partition $\chi = (A_1, \ldots, A_r, B)$ of $V(G)$, such that $\Phi(\chi) \leqslant O(\varepsilon^2)$, $|A_i| \leqslant n^{1-\varepsilon}$ for all $i \in [r]$, and $\mathrm{rank}_{1-\varepsilon^5}(G[B]) \leqslant n^{100\varepsilon}$.*

Proof. We start with an empty partition χ and will repeatedly add sets to χ until we cover $V = V(G)$. Suppose we have already obtained the sets $A_1, \ldots, A_{i-1}$. Let $U_i = V \setminus (A_1 \cup \cdots \cup A_{i-1})$. We run the algorithm of Lemma 4.2 on $G[U_i]$. If it fails to return anything we add the set $B = U_i$ to the partition and halt. (In this case, Lemma 4.2 guarantees that $\mathrm{rank}_{1-\varepsilon^5}(G[B]) \leqslant n^{100\varepsilon}$.) Otherwise, the algorithm returns a set $A_i \subseteq U_i$ with $|A_i| \leqslant |U_i|^{1-\varepsilon} \leqslant n^{1-\varepsilon}$ and $\Phi_{U_i}(A) \leqslant O(\varepsilon^2)$. We continue in this way until we have exhausted all of V. Let $\chi = (A_1, \ldots, A_r, B)$ be the partition that we obtained in this way. Note that

$$\Phi(\chi) = 2\sum_{i=1}^{r} G(A_i, A_{i+1} \cup \cdots \cup A_r \cup B) = 2\sum_{i=1}^{r} G(A_i, U_i \setminus A_i)\,.$$

But since $G(A_i, U_i \setminus A_i) = \Phi'_{U}(A_i)\mu(A_i) \leqslant O(\varepsilon^2\mu(A_i))$, we can upper bound the cost of χ as desired

$$\Phi(\chi) \leqslant O(\varepsilon^4 \sum_{i=1}^{r} \mu(A_i)) \leqslant O(\varepsilon^2).$$

(Since $\sum_{i=1}^{r} \mu(A_i) = 1$.) □

The idea for the next lemma is to apply Lemma 4.3 recursively until we obtain a partition of the vertices into sets A_i are very small ($|A_i| \ll n^{\varepsilon}$) and sets B_i that are small-set expanders. To achieve the bound on the size of the sets, it is enough to recurse up to depth $O(\log(1/\varepsilon)/\varepsilon)$. In each level of the recursion, we cut at most an $O(\varepsilon^2)$ fraction of edges. Hence, the total fraction of edges that we cut across all levels is at most $O(\varepsilon \log(1/\varepsilon))$. (For this argument, it was important that the algorithm of Lemma 4.2 outputs set with $\log_n(1/\mu(S)) \gg \Phi(S)$.)

Lemma 4.4. *There is an algorithm that given an n vertex graph G, and $\varepsilon > 0$, outputs a partition $\chi = (A_1, \ldots, A_r, B_1, \cdots, B_{r'})$ of $[n]$, such that $\Phi(\chi) \leqslant O(\varepsilon \log(1/\varepsilon))$, $|A_i| \leqslant n^{\varepsilon}$ for all $i \in [r]$, and $\mathrm{rank}_{1-\varepsilon^5}(G[B_j]) \leqslant n^{100\varepsilon}$ all $j \in [r']$.*

Proof. We let χ_0 be the trivial partition of one set (with $\Phi(\chi_0) = 0$) and will continually refine the partition using Lemma 4.3 until we reach the desired form. Now for $i = 0, 1, \ldots, 10\log(1/\varepsilon)/\varepsilon$ we repeat the following steps. As long as χ_i does not satisfy the above form, then for every set A of χ_i that satisfies $\mathrm{rank}_{1-\varepsilon^5}(G[A]) \geqslant n^{100\varepsilon} \geqslant |A|^{100\varepsilon}$ we run Lemma 4.3 to obtain a partition χ_A of A with $\Phi_A(\chi_A) \leqslant O(\varepsilon^2)$. We then let χ_{i+1} be the partition obtained by refining every such set A in χ_i according to χ_A. Note that we maintain the invariant that in χ_i, every set A such that $\mathrm{rank}_{1-\varepsilon^5}(G[A]) \geqslant n^{100\varepsilon}$ has size at most $n^{(1-\varepsilon)^i}$. Thus, after $10\log(1/\varepsilon)/\varepsilon$ iterations every such set will have size at most n^{ε}. At the end we output the final partition $\chi = \chi_{10\log(1/\varepsilon)/\varepsilon}$. It just remains to bound $\Phi(\chi)$. To do that it suffices to prove that $\Phi(\chi_{i+1}) \leqslant \Phi(\chi_i) + O(\varepsilon^2)$, since this implis $\Phi(\chi) \leqslant O(\varepsilon^2 \cdot \log(1/\varepsilon)/\varepsilon) = O(\varepsilon \log(1/\varepsilon))$. So we need to prove $\Phi_{\chi_i}(\chi_{i+1}) \leqslant O(\varepsilon^2)$. But indeed, if we let $A_1, \ldots, A_r$ be the sets in χ_i that χ_{i+1} refines, then one can see that

$$\Phi_{\chi_i}(\chi_{i+1}) = \sum_{j=1}^{b} \mu(A_j)\Phi_{A_j}(\chi_{A_j}) \leqslant O(\varepsilon^2)\,,$$

where the last inequality follows from $\sum \mu(A_j) \leqslant 1$ and the guarantee $\Phi_A(\chi_{A_j}) \leqslant O(\varepsilon^2)$ provided by Lemma 4.3. □

Lemma 4.4 immediately implies Theorem 4.1 □

Trace rank bound. Note that by using Theorem 3.2 instead of Theorem 2.3, if we assume the original graph is lazy, then we can get a partition of small trace threshold-rank instead of threshold rank. (One just needs to note that if G is lazy then $G[A]$ is lazy as well for every subset A of G's vertices.) Thus our proof actually yields the following theorem as well:

Theorem 4.5 (Low trace threshold rank decomposition theorem)**.** *There is a polynomial time algorithm that on input a graph G and $\varepsilon > 0$, outputs a partition $\chi = (A_1, \ldots, A_q)$ of $V(G)$ such that $\Phi(\chi) \leqslant O(\varepsilon \log(1/\varepsilon))$ and for every $i \in [q]$,*

$$\mathrm{rank}^*{}_{1-\varepsilon^5}(G[A_i]) \leqslant n^{100\varepsilon}\,.$$

5 A subexponential algorithm for Unique Games

In this section we give a subexponential algorithm for unique games. A *unique game* of n variables and alphabet k is an n vertex graph G whose edges are labeled with permutations on the set $[k]$, where the edge (i, j) labeled with π iff the edge (j, i) is labeled with π^{-1}. An *assignment* to the game is a string $y = (y_1, \ldots, y_n) \in [k]^n$, and the *value* of y is the fraction of edges (i, j) for which $y_j = \pi(y_i)$, where π is the label of (i, j). The *value* of the game G is the maximum value of y over all $y \in [k]^n$. Khot's Unique Games Conjecture [Kho02] states that for every $\varepsilon > 0$, there is a k, such that it is **NP**-hard to distinguish between a game on alphabet k that has value at least $1 - \varepsilon$, and such a game of value at most ε. We now show that this problem can be solved in subexponential time:

Theorem 5.1 (Subexponential algorithm for unique games)**.** *There is an $\exp(kn^{O(\varepsilon)})\,\mathrm{poly}(n)$-time algorithm that on input a unique game G on n vertices and alphabet size k that has an assignment satisfying $1 - \varepsilon^6$ of its constraints outputs an assignment satisfying $1 - O(\varepsilon \log(1/\varepsilon))$ of the constraints.*

5.1 Proof of Theorem 5.1.

We now turn to the proof. We assume the unique game constraint graph is d-regular for some d— this is without loss of generality (see Appendix A.1). For a unique game G, the *label extended graph* of G, denoted $\hat{G}$, is a graph on nk vertices, where for $i, j \in [n]$ and $a, b \in [k]$ we place an edge between (i, a) and (j, b) iff there is an edge (i, j) in G labeled with a permutation π such that $\pi(a) = b$. That is, every vertex $i \in V(G)$ corresponds to the "cloud" $C_i := \{(i, 1), \ldots, (i, k)\}$ in $V(\hat{G})$. We say that $S \subseteq V(\hat{G})$ is *conflict free* if S intersects each cloud in at most one vertex. Note that a conflict free set S in $\hat{G}$ corresponds to a partial assignment $f = f_S$ for the game G (i.e., a partial function from $V(G)$ to $[k]$). We define the *value* of a partial assignment f, denoted $\mathrm{val}(f)$, to be $2/(nd)$ times the number of labeled edges (i, j, π) such that both $f(i)$ and $f(j)$ are defined, and $\pi(f(i)) = f(j)$.

We say that a unique game is *lazy* if each vertex has half of its constraints as self loops with the identity permutation. The following simple lemma will be quite useful for us:

Lemma 5.2. *Suppose that G is lazy. Then* $\mathrm{rank}^*{}_{1-\eta}(\hat{G}) \leqslant k \cdot \mathrm{rank}^*{}_{1-\eta}(G)$.

Proof. Since $\mathrm{rank}^*{}_{1-\eta}(G) = \inf_t\{\mathrm{Tr}(G^{2t})/(1-\eta)^{2t}\}$, this follows from the fact that $\mathrm{Tr}(\hat{G}^t) \leqslant k\,\mathrm{Tr}(G^t)$ for all t. The latter just follows because of the fact that if there is a length t walk in $\hat{G}$ from the vertex (i, a) back to itself then there must be a corresponding length t walk from i back to itself in G (and in fact one where composing the corresponding permutation yields a permutation that has a as a fixed point). Thus every length t walk in G corresponds to at most k such walks in $\hat{G}$. □

Combining this with Lemma 3.3 we get the following corollary:

Corollary 5.3. *For every δ, η and n vertex constraint graph G on alphabet k,* $\mathrm{rank}_{1-\delta\eta}(\hat{G}) \leqslant kn^{\eta}\mathrm{rank}_{1-\eta}(G)$.

Because of Lemma 5.2, we will find it convenient to use the *trace* threshold rank partitioning algorithm of Theorem 4.5. We note that we could have instead used Corollary 5.3 instead, at some quantitative loss to the parameters. Our algorithm is as follows:

Input: Unique game G on n variables of alphabet k that has value at least $1 - \varepsilon^6$.

Operation:

1. Make G lazy by adding to every vertex self loops accounting to half the weight labeled with the identity permutation.
2. Run the partition algorithm of Theorem 4.5 to obtain a partition $\chi = \{A_1, \ldots, A_q\}$ of the graph G with $\Phi(\chi) \leqslant O(\varepsilon \log(1/\varepsilon))$ such that for every i, $\mathrm{rank}_{1-\varepsilon^5}\mathrm{rank}^*(A_i) \leqslant n^{100\varepsilon}$.
3. Let $\hat{A}_1, \ldots, \hat{A}_q$ be the corresponding partition of the label-extended graph $\hat{G}$. Note that for all $t \in [q]$, $\hat{G}[A_t] = G[\hat{A}_t]$ and hence by Lemma 5.2 $\mathrm{rank}_{1-\varepsilon^5}(\hat{G}[A_i]) \leqslant \mathrm{rank}^*{}_{1-\varepsilon^5}(\hat{G}[A_i]) \leqslant kn^{100\varepsilon}$.
4. For every $t = 1 \ldots q$ do the following:
   (a) Run the $\exp(\mathrm{rank}_{1-\varepsilon^5}(\hat{G}[A_t])$-time enumeration algorithm of Theorem 2.2 on the graph $\hat{G}[\hat{A}_t]$ to obtain a sequence of sets $\mathcal{S}_t$.
   (b) For every set $S \in \mathcal{S}_t$, we compute an assignment f_S to the vertices in A_t as follows: For every $i \in A_t$, if $C_i \cap S = \emptyset$, then f_S assigns an arbitrary label to the vertex i, if $|C_i \cap S| > 0$, then f_S assigns one of the labels in $C \cap S$ to the vertex i. Let f_t be the assignment of maximum value, and assign the variables corresponding to vertices in A_t according to f_t. (Note that since the sets $A_1, \ldots, A_q$ are disjoint, every variable will be assigned at exactly one label.)

We now turn to analyze the algorithm. We assume the game has an assignment f_{opt} satisfying $1 - \varepsilon^6$ of the constraints. Note that f_{opt} still has the same value, and in fact even somewhat better— $1 - \varepsilon^6/2$— after we make the graph lazy. Let $\chi = (A_1, \ldots, A_t)$ be the partition obtained by the algorithm in Step 2. Since $\Phi(\chi) \leqslant 1/2$, the assignment f_{opt} satisfies at least $1 - 2(\varepsilon^6/2) = 1 - \varepsilon^6$ of the constraints that are not cut by χ. Let μ_t be the measure of A_t (also equalling the measure of $\hat{A}_t$), and let ε_t be the fraction of constraints in A_t that are violated by f_{opt}. We know that $\sum_{t=1}^{q} \mu_t \varepsilon_t \leqslant 2\varepsilon^6$.

The following lemma implies that the algorithm will output an assignment satisfying at least $1 - O(\varepsilon)$ fraction of the constraints:

Lemma 5.4. *Every partial assignment f_t satisfies all but a $20\varepsilon_t/\eta$ fraction of the constraints in A_t.*

Proof. Let S_{opt} be the subset of $\hat{A}_t$ corresponding to the assignment f_{opt}. Note that $|S_{\mathsf{opt}}| = |A_t|$ and $\Phi_{\hat{A}_t}(S_{\mathsf{opt}}) \leqslant \varepsilon_t$. Thus, the sequence $\mathcal{S}_t$ contains a set S that has symmetric difference with S_{opt} at most $8(\varepsilon_t/\eta)|A_t|$ (Theorem 2.2). Let S' be the subset of $\hat{A}_t$ corresponding to the assignment f_S. The construction of f_S (and thus S') ensures that the symmetric difference between S' and S is at most the symmetric difference between S and S_{opt}. (In fact, the symmetric difference of S and S' is equal to $\sum_{i \in A_t} ||S \cap C_i| - 1|$.) Hence, S' has symmetric difference with S_{opt} at most $16(\varepsilon_t/\eta)|A_t|$. In other words, f_S agrees with f_{opt} on all but a $16\varepsilon_t/\eta$ fraction of the vertices in A_t. Thus f_S violates at most $\varepsilon_t + 16\varepsilon_t/\eta \leqslant 20\varepsilon_t/\eta$ of the constraints in A_t. The lemma follows because we choose f_t as the best assignment among all assignments f_S for $S \in \mathcal{S}_t$. □

Lemma 5.4 implies that among the constraints not cut by χ, the assignment we output satisfies all but a

$$\sum_t \mu_t \cdot 20\varepsilon_t/\eta = (20/\eta) \sum_u \mu_i \varepsilon_t O(\varepsilon^6/\varepsilon^5) = O(\varepsilon) \qquad \text{fraction of constraints.}$$

Since χ cuts at most $O(\varepsilon \log(1/\varepsilon))$ fraction of the constraints, the correctness of the algorithm follows. One just has to note that any solution satisfying $1 - \gamma$ fraction of the lazy game's constraints satisfies at least $1 - 2\gamma$ fraction of the original game's constraints. □

6 Algorithmic results for some UG-hard problems.

Since the Unique Games problem is reducible to a host of other problems, one would hope that our ideas might apply to these problems. We show that this does work out for the Multi Cut problem, shown unique-games hard by [CKK+05]. For other problems such as Max Cut, Sparsest Cut and Vertex Cover some obstacles remain. However we are able to use our algorithm to solve interesting instances of these problems in subexponential time, where by "interesting" we mean the kind of instances that arose in existing integrality gaps, or PCP-based hardness results. The reason is that these instances are highly expanding in subsets of size $o(n)$, which implies they have low threshold rank.

Very recently, [RST10b] showed that a hypothesis about the approximability of small-set expansion (studied in [RS10]) implies that several UG-hard problems, e.g. Max Cut and Sparsest Cut are hard to approximate even on small-set expanders. In particular, this hypothesis implies that for every small enough $\varepsilon > 0$, given a graph G, it is NP-hard to distinguish between the case that the Max Cut value of G is at least $1 - \varepsilon$ and the case that G's Max Cut value is at most $1 - \Omega(\sqrt{\varepsilon})$ and $\Phi_G(2^{-100/\varepsilon}) > 1/2$.

We show that this problem can be solved (even with Max Cut value $1 - O(\varepsilon)$ in the NO case) in time $\exp(n^{\varepsilon^c})$ for some absolute constant $c > 0$.

6.1 Subexponential algorithm for Multi Cut

In an instance of Multi Cut, we are given a graph G with vertex set V and a set D of pairs of vertices (*demand pairs*). A (D-)multicut is a partition $\chi = \{S_1, \ldots, S_r\}$ of V such that none of the sets S_i contains a demand pair in D. The goal is to find a multicut χ that minimize $\Phi(\chi)$, the fraction of edges cut by the partition.

Using the ideas in [SV09], a straight-forward (but somewhat tedious) adaptation of our algorithm for Unique Games gives the following algorithm for Multi Cut.

Theorem 6.1. *There exists a constant $d \geqslant 1$ and an $\exp(n^{\varepsilon})\,\mathrm{poly}(n)$-time algorithm that given a multicut instance with optimal value ε^d finds a solution with value ε. Here, d is an absolute constant.*

The approximation guarantee of the Multi Cut algorithm above is incomparable with the $O(\log n)$-approximation of Garg, Vazirani, and Yannakakis [GVY04]. For $\varepsilon \ll 1/\log n$, the $O(\log n)$-approximation gives better guarantees. If $\varepsilon \geqslant 1/\log n$, our algorithm gives the best known guarantees. (If the degrees of the demand pairs are small, better approximations are possible [SV09].)

An important special case of Unique Games, called Γ-Max 2-Lin, reduces to Multi Cut [SV09]. The reduction in [SV09] has the feature that the blow-up of the instance is linear in the alphabet size of the Γ-Max 2-Lin instance (in contrast to the usual UG-reductions using longcodes).

Combining this reduction with the above theorem, we get an algorithm for Γ-Max 2-Lin. In contrast to Theorem 5.1, the algorithm for Γ-Max 2-Lin has a non-trivial running time even for $k = n$.

Theorem 6.2 (Improved algorithm for Γ-Max 2-Lin)**.** *There is an $\exp((kn)^{O(\varepsilon)})\,\mathrm{poly}(n)$-time algorithm that on input a Γ-Max 2-Lin instance G on n vertices and alphabet size k that has an assignment satisfying $1 - \varepsilon^d$ of its constraints outputs an assignment satisfying $1 - O(\varepsilon)$ of the constraints. Here, $d \geqslant 1$ is an absolute constant.*

We give a rough sketch of Theorem 6.1. The details are deferred to the full version.

Proof Sketch for Theorem 6.1. Let $\eta = \varepsilon^5$. We consider two cases. If $\mathrm{rank}_{1-\eta}(G) \geqslant n^{100\varepsilon}$, then there exists a set S with $|S| \leqslant n^{1-\varepsilon}$ and $\Phi(S) \leqslant O(\varepsilon^2)$ (Lemma 4.2). In this case, we recurse on the subgraphs $G[S]$ and $G[V \setminus S]$. Otherwise, if $\mathrm{rank}_{1-\eta}(G) \leqslant n^{100\varepsilon}$, we can enumerate all non-expanding subsets in time $\exp(n^{100\varepsilon})\,\mathrm{poly}(n)$. For every such set S, we prune it to a S' that doesn't contain any demand pairs (using the

obvious greedy algorithm). Out of all such pruned sets S', we select the set S^* that has the small expansion in G. We can use S^* as one of the components of our multicut and recurse on the subgraph $G[V \setminus S^*]$. This recursive algorithm and its analysis is very similar to Theorem 4.1. An important difference is that the costs of the sets S^* are charged to the cost of the optimal Multi Cut solution. □

6.2 Max Cut, Sparsest Cut and Vertex Cover on low threshold rank graphs

We now sketch how one can use the eigenspace enumeration method to give improved approximation guarantees Max Cut, Sparsest Cut and Vertex Cover in time exponential in the threshold rank. One can then optimize the relation between expansion and threshold rank, as in Theorem 2.2, to obtain subexponential algorithms on small set expanders, although we defer these calculations to the final version of this work. We first record the following variant of the enumeration result (Theorem 2.2):

Theorem 6.3 (Eigenspace enumeration revisited)**.** *There is an* $\exp(\mathrm{rank}_{1-\eta}(G))\,\mathrm{poly}(n)$*-time algorithm that given* $\varepsilon > 0$ *and an* n *vertex* d*-regular graph* G *containing a set* S *with* $G(S, \overline{S}) \geqslant 1 - \varepsilon$*, outputs a sequence of sets, one of which has symmetric difference at most* $16(\varepsilon/\eta)|S|$ *with the set* S*.*

In particular for $\varepsilon < 0.01$ and $\eta = 1/2$ the algorithm will output a list of sets containing a set S' such that $|E(S', \overline{S'})| \geqslant (1 - 17\varepsilon)|E|$.

Proof sketch. The proof follows in the same way as Theorem 2.2, except that we use a vector with positive and negative entries instead of the zero/non-zero characteristic vector used there. That is, we define $\chi_S(i) = +1$ if $i \in S$ and $\chi_S(i) = -1$ otherwise. Note that $\langle \chi_S, G\chi_S \rangle \leqslant -1 + 2\varepsilon$. We then use the same argument as before to show that χ_S must have large projection to the top eigenspace. □

We now sketch our algorithms for Max Cut, Sparsest Cut and Vertex Cover on graphs with small rank:

Theorem 6.4. *There is an absolute constant* c *and an algorithm that on input an* n *vertex regular graph* G *such that* Max Cut$(G) \geqslant 1 - \varepsilon$*, runs in time* $\exp(\mathrm{rank}_{1-\eta}(G))\,\mathrm{poly}(n)$ *and produces a cut of size at most* $1 - c\varepsilon/\eta$*.*

Proof sketch. The maximum cut is a set S satisfying $G(S, \overline{S}) \geqslant 1 - \varepsilon$, and we will use Theorem 6.3 to obtain a set S' of $O(\varepsilon/\eta)$ symmetric difference from S. □

Theorem 6.5. *There is an absolute constant* c *and an algorithm that on input an* n *vertex regular graph* G *such that with a set* S *of size in* $(n/3, 2n/3)$ *such that* $\Phi_G(S) \leqslant \varepsilon$*, runs in time* $\exp(\mathrm{rank}_{1-\eta}(G))\,\mathrm{poly}(n)$ *and produces a set* S' *of size in* $(n/3, 2n/3)$ *satisfying* $\Phi_G(S) \leqslant c\varepsilon/\eta$*.*

Proof sketch. This is obtained by just using the enumeration of Theorem 2.2. □

Theorem 6.6. *There is an absolute constant* c *and an algorithm that on input an* n *vertex regular graph* G *with a vertex cover of size* k*, outputs a vertex cover of size* $(2 - c\eta)k$ *in time* $\exp(\mathrm{rank}_{1-\eta}(G))\,\mathrm{poly}(n)$*.*

Proof sketch. Since the graph is regular, we know that $k \leqslant n/2$, and so denote $k = n(1/2 + \varepsilon)$ for some ε. Letting S be the vertex cover, this means that $G(S, \overline{S}) \geqslant 1 - 2\varepsilon$, and so we can find in the above time a set S' with symmetric difference at most $(32\varepsilon/\eta)|S|$ from S. This means that after removing S' from the graph, we have a vertex cover of the remaining edges consisting of at most $(32\varepsilon/\eta)|S|$ vertices, and by the simple greedy algorithm we can find a vertex cover of at most $(64\varepsilon/\eta)|S|$ vertices. This means we can always find a vertex cover of size $\min\{n, k(1+64\varepsilon/\eta)\}$ or equivalently a factor $\min\{n/k, 1+64\varepsilon/\eta\}$ larger than the optimum k. Since $k = 1/2 + \varepsilon$ one can see that the ε that will maximize this expression is roughly $\eta/64$, in which case we'll get $2 - O(\eta)$ approximation factor. □

7 Threshold rank of hypercontractive graphs

Integrality gap examples for several problems use graphs defined using the *noise operator* on the unit sphere. We define the class of *hypercontractive* graphs that include such graphs, and show that they have low (i.e., polylogarithmic) threshold rank. As a corollary we get an algorithm that certifies in quasipolynomial time that a hypercontractive graph is indeed a small set expander.

A graph G with vertex set V is *θ-hypercontractive* if

$$\forall x \in \mathbb{R}^v.\ \|Gx\| \leqslant \|x\|_{2-\theta}\,,$$

where $\|x\|_{2-\theta}^{2-\theta} = \mathsf{E}_{i\in V}\, x_i^{2-\theta}$.

Lemma 7.1. *If G is θ-hypercontractive, then $\Lambda(\delta) < \delta^\gamma$ for some $\gamma = \Omega(\theta)$, where Λ is the parameter defined in Section 3.1.*

Proof. For every vector $x \in \mathbb{R}^V$,

$$\begin{aligned}
\|Gx\|^{2-\theta} &\leqslant \mathop{\mathsf{E}}_{i\in V} x_i^{2-\theta} \\
&= \mathop{\mathsf{E}}_{i\in V} x_i^{2(1-\theta)} \cdot x_i^{\theta} \\
&\leqslant (\mathsf{E}_{i\in V}\, x_i^2)^{1-\theta} \cdot (\mathsf{E}_{i\in V}\, x_i)^{1/\theta} \qquad \text{(using Hölder's Inequality)} \\
&= \|x\|^{2(1-\theta)}\|x\|_1^{\theta}
\end{aligned}$$

Thus, for $\gamma = \theta/(2-\theta)$, we have $\|Gx\| \leqslant \|x\|^{1-\gamma}\|x\|_1^{\gamma}$. Every vector $x \in \Omega_\delta$ satisfies $\|x\|_1 \leqslant \sqrt{\delta}\|x\|$. Hence, for $x \in \Omega_\delta$, we have $\|Gx\| \leqslant \delta^{\gamma/2}\|x\|$, which proves that $\Lambda(\delta) \leqslant \delta^{\gamma/2}$. □

Lemma 7.2 (Schatten norm bound for hypercontractive graphs)**.** *If G is θ-hypercontractive, then for all even k,*

$$\operatorname{Tr} G^k \leqslant n^{(1-\gamma)^k}\,,$$

where $\gamma = \theta/(2-\theta)$.

Proof. In the proof of the previous lemma, we showed that $\|Gx\| \leqslant \|x\|^{1-\gamma}\|x\|_1^{\gamma}$. Let $e_1, \ldots, e_n$ be the canonical basis of $\mathbb{R}^V$. We normalize the vectors so that $\|e_i\|_1 = 1$ and $\|e_i\| = \sqrt{n}$ for all $i \in V$. We can easily verify that for all $j \in \mathbb{N}$ and $i \in V$,

$$\|G^j e_i\| \leqslant \|e_i\|_1^{(1-\gamma)^j} = n^{(1-\gamma)^j/2}\,.$$

(Here, we use that $\|G^j e_i\|_1 = 1$ for all $j \in N$ and $i \in V$.) Then, for all even k

$$\begin{aligned}
\operatorname{Tr} G^k &= \mathop{\mathsf{E}}_{i\in V} \langle e_i, G^k e_i\rangle \\
&= \mathop{\mathsf{E}}_{i\in V} \|G^{k/2} e_i\|^2 \\
&\leqslant n^{(1-\gamma)^{k/2}}\,.
\end{aligned}$$

□

Lemma 7.3 (Threshold rank for hypercontractive graphs)**.** *If G is θ-hypercontractive, then*

$$\operatorname{rank}_{1/2}(G) \leqslant (\log n)^{\gamma}\,.$$

Proof. Let $m = \operatorname{rank}_{1/2}(G)$. By the previous lemma, we have

$$m \leqslant 2^k \operatorname{Tr} G^k \leqslant 2^k n^{(1-\gamma)^{k/2}}\,.$$

If we choose $k \approx 2/\gamma \cdot \log\log n$, we get that $m \leqslant (\log n)^{\gamma}$. □

8 Conclusions

The most important open question is of course whether our methods can be extended to yield an $\exp(n^{o(1)})$-time algorithm for unique games, hence refuting the Unique Games Conjecture, and more generally what is the true complexity of the unique games and small set expansion problems. We note that any quantitative improvement to the bounds of Theorem 2.3 would translate to an improvement in our algorithm for the small set expansion problem, and so it might result in refuting the stronger variant of the UGC proposed in [RS10]. Another open question is whether our techniques can yield subexponential algorithms with better approximation guarantees for unique-games hard problems such as Vertex Cover, Max Cut, Sparsest Cut on every instance.

Acknowledgements

We are very grateful to Assaf Naor, who first suggested in an intractability center meeting that the eigenvalue distribution, and in particular Schatten norms, could be related to small set expansion, and gave us a copy of the manuscript [Nao04]. We thank Alexandra Kolla for giving us an early copy of the manuscript [Kol10]. The authors also had a number of very fruitful conversations on this topic with several people including Moritz Hardt, Thomas Holenstein, Russell Impagliazzo, Guy Kindler, William Matthews, Prasad Raghavendra, and Prasad Tetali.

References

[AIMS10] S. Arora, R. Impagliazzo, W. Matthews, and D. Steurer. Improved algorithms for unique games via divide and conquer. Technical Report ECCC TR10-041, ECCC, 2010.

[AK09] N. Alon and B. Klartag. Economical toric spines via Cheeger's inequality. *J. Topol. Anal.*, 1(2):101–111, 2009.

[AKK+08] S. Arora, S. Khot, A. Kolla, D. Steurer, M. Tulsiani, and N. Vishnoi. Unique games on expanding constraints graphs are easy. In *Proc. 40th STOC*. ACM, 2008.

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Combin. Theory Ser. B*, 38(1):73–88, 1985.

[Aus07] P. Austrin. Towards sharp inapproximability for any 2-CSP. In *FOCS*, pages 307–317, 2007.

[BHHS10] B. Barak, M. Hardt, T. Holenstein, and D. Steurer. Subsampling mathematical relaxations and average-case complexity, 2010. Manuscript.

[BJK00] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput*, 34(3):720–742, 2005. Prelim version in ICALP '00.

[Che70] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in analysis (Papers dedicated to Salomon Bochner, 1969)*, pages 195–199. Princeton Univ. Press, 1970.

[Chu07] F. Chung. Random walks and local cuts in graphs. *Linear Algebra Appl.*, 423(1):22–32, 2007.

[CKK+05] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006. Prelim version in CCC 2005.

[CMM06a] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *STOC*, pages 205–214, 2006.

[CMM06b] E. Chlamtac, K. Makarychev, and Y. Makarychev. How to play unique games using embeddings. In *Proc. 47th FOCS*, pages 687–696. Citeseer, 2006.

[CP05] C. Chekuri and M. Pal. A recursive greedy algorithm for walks in directed graphs. In *Proc. FOCS*, pages 245–253, 2005.

[DI98] T. Dimitriou and R. Impagliazzo. Go with the winners for graph bisection. In *Proc 9th SODA*, pages 510–520, 1998.

[Dod84] J. Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Trans. Amer. Math. Soc.*, 284(2):787–794, 1984.

[FK99] A. M. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[FR04] U. Feige and D. Reichman. On systems of linear equations with two variables per equation. *Proc. of RANDOM-APPROX*, pages 117–127, 2004.

[GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543, 2003.

[GMR08] V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *FOCS*, pages 573–582, 2008.

[GMT06] S. Goel, R. Montenegro, and P. Tetali. Mixing time bounds via the spectral profile. *Electron. J. Probab.*, 11:no. 1, 1–26 (electronic), 2006.

[GT06] A. Gupta and K. Talwar. Approximating unique games. In *Proc. 17th SODA*, page 106. ACM, 2006.

[GVY04] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[Hås97] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. Prelim version STOC '97.

[HK03] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proc. STOC*, page 594. ACM, 2003.

[IPZ01] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of 34th STOC*, pages 767–775, New York, 2002. ACM Press.

[KKMO07] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

[Kol10] A. Kolla. Spectral algorithms for unique games. In *Proc. CCC*, 2010. To appear.

[KR08] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[KS09a] S. Khot and R. Saket. SDP integrality gaps with local ℓ_1-embeddability. In *FOCS*, pages 565–574, 2009.

[KS09b] G. Kun and M. Szegedy. A new line of attack on the dichotomy conjecture. In *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing*, pages 725–734. ACM New York, NY, USA, 2009.

[KT07] A. Kolla and M. Tulsiani. Playing random and expanding unique games. Unpublished manuscript available from A. Kolla's webpage, to appear in journal version of [AKK+08], 2007.

[KV05] S. Khot and N. K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *FOCS*, pages 53–62, 2005.

[LL93] A. K. Lenstra and H. W. J. Lenstra. *The Development of the Number Field Sieve*. Springer-Verlag, 1993.

[LS93] N. Linial and M. E. Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

[Luk82] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

[MNRS08] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *STOC*, pages 11–20, 2008.

[MOO05] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Annals of Mathematics*, 101:101, 2008. Preliminary version in FOCS '05.

[MR08] D. Moshkovitz and R. Raz. Two query PCP with sub-constant error. In *Proc. 49th FOCS*, pages 314–323, 2008.

[Nao04] A. Naor. On the Banach space valued Azuma inequality and small set isoperimetry in Alon-Roichman graphs. Unpublished manuscript, 2004.

[PY88] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *JCSS*, 43(3):425–440, 1991. Prelim version in STOC 88.

[Rag08] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proc.* 40*th STOC*, pages 245–254. ACM, 2008.

[RS09] P. Raghavendra and D. Steurer. Integrality gaps for strong SDP relaxations of Unique Games. In *FOCS*, pages 575–585, 2009.

[RS10] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *STOC*, 2010. To appear.

[RST10a] P. Raghavendra, D. Steurer, and P. Tetali. Approximations for the isoperimetric and spectral profile of graphs and related parameters. In *Proc. STOC*. ACM, 2010.

[RST10b] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. manuscript, 2010.

[SHI90] R. Stearns and H. Hunt III. Power indices and easier hard problems. *Math. Syst. Theory*, 23(4):209–225, 1990.

[Ste10] D. Steurer. Fast SDP algorithms for constraint satisfaction problems. In *SODA*, 2010. To appear.

[SV09] D. Steurer and N. Vishnoi. Connections between unique games and multicut. Technical Report TR09-125, ECCC, 2009.

[Sze76] E. Szemerédi. Regular partitions of graphs. *Problèmes combinatoires et théorie des graphes, Orsay*, 1976.

[Tre05] L. Trevisan. Approximation algorithms for unique games. In *FOCS*, pages 197–205, 2005.

A Appendix

A.1 Reductions to regular instances

We sketch here a reduction from the problem, for $\varepsilon < \gamma$, of distinguishing between a $1-e$ vs $1-\gamma$ satisfiable unique game G to the problem of distinguishing between a $1-\varepsilon/10$ vs $1-\gamma/10$ satisfiable unique game G' whose constraint graph is *regular* (i.e., every vertex is adjacent to the edges with the same total weight). The proof uses the standard expander-based construction originating from [PY88]. We replace each vertex v in G of degree d with a "cloud" of d new vertices, each connected to one of v's original neighbors. We place an expander graph of some constant degree d_0 on the vertices of the clouds, and set the weights so that now every vertex in the graph has one outside edge, and d_0 expander edges into its cloud. We assume that the expander graph has the property that every set S with $\mu(S) \leqslant 1/2$ satisfies $\Phi(S) > 0.2$. We then set the weights so that the total weight of the expander edges is 0.9 of the graph, and place equality constraints on these edges to obtain the game G'. Clearly, a $1-\varepsilon$ satisfying assignment in G corresponds to a $0.9+0.1(1-\varepsilon) = 1-\varepsilon/10$ satisfying assignment in G' that assigns the same value to all vertices in the same cloud. Moreover, given an assignment y' satisfying $1-\beta$ fraction of G' constraints, one can see that if we transform y' to y by assigning every cloud according to its most popular value, then we can only improve the fraction of constraints we satisfy. Indeed, consider a set S of vertices in a cloud that were originally given a value that was not the most popular one. Thus, $\mu(S) \leqslant 1/2$ and hence $\Phi(S) > 0.2$. This implies that in the assignment y, S is involved in at least $0.9 \cdot 0.2|S|$ weight of unsatisfied equality constraints. On the other hand, in y' the set S will satisfy all equality constraints and violate at most $0.1|S|$ of its other constraints. Summing up over all such S's we get that the weight of constraints changed from violated to satisfied when moving from y to y' is at least as large than the weight of constraints changed from satisfied to violated. (We may be counting some constraints twice, but we had a factor of two slackness anyway.) Thus y' still has at least $1-\beta$ value in G', and hence translates into an assignment with $1-10\beta$ value in G.

A.2 Restricted eigenvalues and small-set expansion (Proof of Lemma 3.4)

In this section, we prove Lemma 3.4.

Lemma 3.4 (Restated). *Suppose $\Phi(S) \geqslant \varepsilon$ for all sets S of measure at most δ. Then,*

$$\Lambda(\delta/4) \leqslant 1 - \varepsilon^2/32\,.$$

Moreover, if $x \in \Omega_{\delta/4}$ is a unit vector such that $\|Gx\| > 1 - \varepsilon^2/32$, then there exists a level set S of x such that $\mu(S) \leqslant \delta$ and $\Phi(S) < \varepsilon$.

To allow for a more systematic proof, we introduce two parameters, $\Lambda'(\delta)$ and $\Lambda''(\delta)$, which are closely related to $\Lambda(\delta)$,

$$\Lambda'(\delta) \stackrel{\text{def}}{=} \max_{x \in \Omega_\delta} \frac{\langle x, Gx\rangle}{\langle x, x\rangle}\,, \tag{A.1}$$

$$\Lambda''(\delta) \stackrel{\text{def}}{=} \max_{\substack{x \in \mathbb{R}^V \\ \mu(\operatorname{supp} x) \leqslant \delta}} \frac{\langle x, Gx\rangle}{\langle x, x\rangle}\,. \tag{A.2}$$

Here, $\operatorname{supp} x$ denotes the set of non-zero coordinates of $x \in \mathbb{R}^V$. We remark that the parameters are ordered as follows $\Lambda(\delta) \geqslant \Lambda'(\delta) \geqslant \Lambda''(\delta)$. We omit the (straightforward) proofs, since we don't need these relations in the proof of Lemma 3.4. For lazy graphs, all three parameters will turn out to be equivalent up to small constant factors. (In fact, all but the next lemma hold also for general graphs.) The last lemma of this section relates $\Lambda''(\delta)$ to the expansion of small sets in the graph. The parameter $\Lambda''(\delta)$ is the *spectral profile*[8] of a graph (see [GMT06, RST10a]).

Lemma A.1. *For every $\delta > 0$,*

$$\Lambda(\delta)^2 \leqslant \Lambda'(\delta)\,.$$

Proof. Since G is lazy, there exists a graph G' such that $G = \frac{1}{2}G' + \frac{1}{2}I$. Notice that $G - G^2 = \frac{1}{4}I - \frac{1}{4}(G')^2$ is positive semidefinite. Therefore, $\langle x, Gx\rangle \geqslant \langle x, G^2x\rangle = \|Gx\|^2$ for every vector $x \in \mathbb{R}^V$. It follows that $\Lambda'(G) \geqslant \Lambda(G)^2$. □

Recall that Ω_δ contains all δ-sparse vectors. (Here, we say a vector is δ-sparse if its support has measure at most δ.) The following lemma shows that, in fact, every vector in Ω_δ is "close" to a $O(\delta)$-sparse vectors. The proof uses a straight-forward truncation argument.

Lemma A.2. *If $\Lambda'(\delta) \geqslant 1 - \varepsilon$, then*

$$\Lambda''(4\delta) \geqslant 1 - 2\varepsilon\,.$$

Proof. Let $x \in \Omega_\delta$ be such that $\langle x, Gx\rangle \geqslant (1-\varepsilon)\langle x, x\rangle$. We normalize x so that $\|x\|^2 = \delta$ (and therefore $\|x\|_1 \leqslant \delta$, since $x \in \Omega_\delta$). We also may assume that x is non-negative. Consider the vector y with $y_i = \max\{x_i - 1/4, 0\}$. We can verify that $y_i^2 \geqslant x_i^2 - 1/2 \cdot x_i$ and $(y_i - y_j)^2 \leqslant (x_i - x_j)^2$ for all $i, j \in V$. The first property implies

$$\|y\|^2 \geqslant \|x\|^2 - \tfrac{1}{2}\|x\|_1 \geqslant \tfrac{\delta}{2}\,. \tag{A.3}$$

The second property implies

$$\|y\|^2 - \langle y, Gy\rangle = \tfrac{1}{2}\sum_{ij} G_{ij}(y_i - y_j)^2 \leqslant \tfrac{1}{2}\sum_{ij} G_{ij}(x_i - x_j)^2 = \|x\|^2 - \langle x, Gx\rangle \leqslant \varepsilon\delta\,. \tag{A.4}$$

Combining (A.3) and (A.4), we get

$$\frac{\langle y, Gy\rangle}{\langle y, y\rangle} \geqslant \frac{\|y\|^2 - \varepsilon\delta}{\|y\|^2} = 1 - \varepsilon\delta/\|y\|^2 \geqslant 1 - 2\varepsilon\,.$$

On the other hand, the support of y has measure at most 4δ. (The average value of $|x_i|$ is δ. Hence, at most a 4δ fraction of the coordinates can have value more than $1/4$.) We conclude $\Lambda(4\delta) \geqslant 1 - 2\varepsilon$. □

[8] More precisely, the spectral profile usually refers to the quantity $1 - \Lambda''(\delta)$.

The following relation between $\Lambda''(\delta)$ and the expansion of small sets in G is a direct consequence of the proof of Cheeger's inequality [AM85, Alo86] (see for example [GMT06, Chu07, AK09]).

Lemma A.3. *Suppose $\Lambda''(\delta) \geqslant 1-\varepsilon$. Then, there exists a vertex set S with $\mu(S) \leqslant \delta$ and*

$$\Phi(S) \leqslant \sqrt{8\varepsilon}\,.$$

Proof. By the definition of $\Lambda''(\delta)$, there exists a δ-sparse vector such that $\langle x, Gx\rangle \geqslant (1-\varepsilon)\langle x, x\rangle$. By scaling and taking absolute values, we may assume that all coordinates of x satisfy $0 \leqslant x_i \leqslant 1$. We consider the following distribution over vertex sets S: Sample a random threshold $t \in [0,1]$ and output the set $S_t := \{i \in V \mid x_i^2 \geqslant t\}$. This distribution over vertex sets has the following properties:

$$\mathop{\mathsf{E}}_{t\in[0,1]} \mu(S_t) = \langle x, x\rangle\,, \tag{A.5}$$

$$\mathop{\mathsf{E}}_{t\in[0,1]} \langle \mathbf{1}_{S_t}, G\mathbf{1}_{V\setminus S_t}\rangle = \mathop{\mathsf{E}}_{i\in V} \sum_j G_{ij}|x_i^2 - x_j^2|\,. \tag{A.6}$$

By Cauchy–Schwarz, we can upper bound the latter expectation as follows

$$\begin{aligned}
\mathop{\mathsf{E}}_{i\in V} \sum_j G_{ij}|x_i^2 - x_j^2| &\leqslant \Big(\mathop{\mathsf{E}}_{i\in V} \textstyle\sum_j G_{ij}(x_i - x_j)^2\Big)^{1/2} \cdot \Big(\mathop{\mathsf{E}}_{i\in V} \textstyle\sum_j G_{ij}(x_i + x_j)^2\Big)^{1/2} \\
&= \Big(2\langle x, x\rangle - 2\langle x, Gx\rangle\Big)^{1/2} \cdot \Big(2\langle x, x\rangle + 2\langle x, Gx\rangle\Big)^{1/2} \\
&\leqslant 2\sqrt{2}\Big(\langle x, x\rangle - \langle x, Gx\rangle\Big)^{1/2} \cdot \langle x, x\rangle^{1/2} \leqslant 2\sqrt{2}\cdot\sqrt{\varepsilon}\cdot\langle x, x\rangle\,.
\end{aligned}$$

It follows that there exists a threshold t^* such that

$$\Phi(S_{t^*}) = \frac{\langle \mathbf{1}_{S_{t^*}}, G\mathbf{1}_{V\setminus S_{t^*}}\rangle}{\mu(S_t)} \leqslant \frac{\mathsf{E}_{t\in[0,1]}\langle \mathbf{1}_{S_t}, G\mathbf{1}_{V\setminus S_t}\rangle}{\mathsf{E}_{t\in[0,1]}\mu(S_t)} \leqslant 2\sqrt{2}\cdot\sqrt{\varepsilon}\,.$$

On the other hand, every set S_t is a subset of the support of x and thus $\mu(S_{t^*}) \leqslant \delta$ as desired. □

We can prove Lemma 3.4 by combining the previous lemmas: Suppose that $\Phi(S) \geqslant \varepsilon$ for all S with $\mu(S) \leqslant \delta$. Then, we have $\Lambda''(\delta) \leqslant 1-\varepsilon^2/8$ (Lemma A.3), $\Lambda'(\delta/4) \leqslant 1-\varepsilon^2/16$ (Lemma A.2) and so $\Lambda(\delta/3) \leqslant \sqrt{1-\varepsilon^2/16} \leqslant 1-\varepsilon^2/32$ (Lemma A.1). Moreover, following the proof of these lemmas, we see that they allow us to obtain from a vector $x \in \Omega_{\delta/4}$ satisfying $\|Gx\|_2 \geqslant (1-\varepsilon^2/32)\|x\|_2$ a set S of measure at most δ satisfying $\Phi(S) \leqslant \varepsilon$, and moreover the set S is obtained as a level set of the vector y where $y_i = \max\{x_i - 1/4, 0\}$, or equivalently, a level set of x.