THEORY OF COMPUTING, Volume 5 (2009), pp. 83–117
www.theoryofcomputing.org

SDP gaps and UGC-hardness for Max-Cut-Gain*

Subhash Khot Ryan O'Donnell†

Received: July 9, 2008; published: May 29, 2009.

Abstract: Given a graph with maximum cut of (fractional) size c, the semidefinite programming (SDP)-based algorithm of Goemans and Williamson is guaranteed to find a cut of size at least $.878 \cdot c$. However this guarantee becomes trivial when c is near $1/2$, since making random cuts guarantees a cut of size $1/2$ (i. e., half of all edges). A few years ago, Charikar and Wirth (analyzing an algorithm of Feige and Langberg) showed that given a graph with maximum cut $1/2+\varepsilon$, one can find a cut of size $1/2+\Omega(\varepsilon/\log(1/\varepsilon))$. The main contribution of our paper is twofold:

1. We give a natural and explicit $1/2+\varepsilon$ vs. $1/2+O(\varepsilon/\log(1/\varepsilon))$ integrality gap for the Max-Cut SDP based on Euclidean space with the Gaussian probability distribution. This shows that the SDP-rounding algorithm of Charikar-Wirth is essentially best possible.

2. We show how this SDP gap can be translated into a Long Code test with the same parameters. This implies that beating the Charikar-Wirth guarantee with *any* efficient algorithm is NP-hard, assuming the Unique Games Conjecture (UGC). This result essentially settles the asymptotic approximability of Max-Cut, assuming UGC.

Building on the first contribution, we show how "randomness reduction" on related SDP gaps for the Quadratic-Programming problem lets us make the $\Omega(\log(1/\varepsilon))$ gap as

*An extended abstract appeared in STOC 2006 [30].

†Supported in part by NSF CAREER grant CCF-0747250.

ACM Classification: F.2.2, G.2.2

AMS Classification: 68Q17, 68Q25, 68W25, 52A40, 90C20, 90C22

Key words and phrases: Max-Cut, Max-Cut-Gain, semidefinite programming, semidefinite programming gaps, Unique Games Conjecture, dictator testing, Gaussian space, quadratic programming, Grothendieck inequality

© 2009 Subhash Khot and Ryan O'Donnell
© Licensed under a Creative Commons Attribution License
DOI: 10.4086/toc.2009.v005a004

large as $\Omega(\log n)$ for n-vertex graphs. In addition to optimally answering an open question of Alon, Makarychev, Makarychev, and Naor, this technique may prove useful for other SDP gap problems.

Finally, illustrating the generality of our second contribution, we also show how to translate the Davie–Reeds SDP gap for the Grothendieck Inequality into a UGC-hardness result for computing the $\|\cdot\|_{\infty\mapsto 1}$ norm of a matrix.

1 Introduction

1.1 Max-Cut

Constraint satisfaction problems (CSPs) constitute some of the most fundamental algorithmic tasks. For most interesting CSPs, finding an optimum solution is NP-hard; hence it is of interest to study "approximation algorithms," i. e., efficient algorithms guaranteed to find a solution within a certain factor of the optimum. Unfortunately, the computational complexity of approximating CSPs is still not well understood; for example, it is not known if approximating Vertex-Cover to a factor of $3/2$ is in P, nor is the problem known to be NP-hard.

The main topic of this paper is the approximability of the Max-Cut problem—arguably the simplest of all NP-hard constraint satisfaction problems. Recall that Max-Cut is the following algorithmic task: Given an undirected graph G with nonnegative weights on the edges, partition its vertices into two parts so as to maximize the "value" of the "cut"—i. e., the sum of the weights of the edges that straddle the partition. Throughout this paper we will assume that graphs' edge-weights are normalized so that their total sum is 1.

Regarding approximation algorithms for Max-Cut, the trivial solution of picking a random partition guarantees (in expectation) a cut of value at least $1/2$. No essential improvement on this was known until the breakthrough paper of Goemans and Williamson [20]. Let us very briefly review the Goemans–Williamson algorithm. Given an n-vertex input graph G with weight a_{ij} on edge (i,j) (and weights summing to 1), the algorithm writes down the associated Max-Cut problem as an integer program:

$$\max \quad \tfrac{1}{2} + \tfrac{1}{2} \sum_{i,j=1}^{n} (-a_{ij})\,(y_i \cdot y_j) \tag{1.1}$$

$$\text{subject to:} \quad y_i \in \{-1, 1\}.$$

The algorithm then relaxes this to a semidefinite program (SDP) which can be solved efficiently:

$$\max \quad \tfrac{1}{2} + \tfrac{1}{2} \sum_{i,j=1}^{n} (-a_{ij})\,(y_i \cdot y_j) \tag{1.2}$$

$$\text{subject to:} \quad y_i \in S^{n-1}.$$

Here S^{n-1} denotes the unit sphere in n dimensions, $\{y \in \mathbb{R}^n : \|y\|_2 = 1\}$, and $\cdot$ is interpreted as inner product. Finally, given the optimal unit vector solution $\{y_i^*\}$, the algorithm "rounds" the vectors to a ± 1

solution as follows: It picks a random vector r from the n-dimensional Gaussian distribution (i. e., each component r_i is an independent standard Gaussian) and sets

$$y_i = \operatorname{sgn}(y_i^* \cdot r)\,.$$

Goemans and Williamson showed that their algorithm has the following two guarantees (in expectation, but the results can be made to hold with high probability):

- Given a graph with maximum cut $1-\varepsilon$ the algorithm finds a cut of value at least $1-\Theta(\sqrt{\varepsilon})$.
- Given a graph with maximum cut s the algorithm finds a cut of value at least $\alpha_{\mathrm{GW}} \cdot s$, where $\alpha_{\mathrm{GW}} \approx .878$ is a certain trigonometric quantity.

On the hardness-of-approximation side, the best NP-hardness result known for Max-Cut, due to Håstad [24] and Trevisan–Sorkin–Sudan–Williamson [44], shows that given a graph with maximum cut $17/21$ it is NP-hard to find a cut with value $16/21+\varepsilon$.

However, Khot, Kindler, Mossel, and O'Donnell [28] recently showed hardness results that *match* the above two guarantees of Goemans and Williamson, assuming Khot's "Unique Games Conjecture" (UGC) [27]. For a discussion of why assuming UGC seems necessary for sharp results given our current knowledge, see [27, 28].

1.2 Max-Cut-Gain

Despite the results from [28], the Goemans–Williamson algorithm is certainly suboptimal in some cases. For example, given a graph with optimum cut .55, the Goemans–Williamson algorithm is only guaranteed to return a cut with value $.878 \cdot .55 < .49$, which is worse than the trivial random algorithm. Indeed, Alon, Sudakov, and Zwick [4] have shown that there exist graphs on which the Goemans–Williamson algorithm performs just this poorly. The issue was addressed first by Zwick [45] who gave an alternate SDP rounding procedure which on graphs with maximum cut of fractional size c is guaranteed to find a cut of size at least $\beta(c)$, where β is a somewhat explicitly defined function satisfying $\beta(c)/c \to 1$ as $c \to 1/2$. The function β was improved by Feige and Langberg [19] using a different rounding algorithm:

$$\text{Step 1:}\quad \text{Set } y_i' = \operatorname{round}_T(y_i^* \cdot r), \qquad \text{where } \operatorname{round}_T(t) = \begin{cases} t/T & \text{if } t \in [-T,T], \\ \operatorname{sgn}(t) & \text{otherwise.} \end{cases} \tag{1.3}$$

$$\text{Step 2:}\quad \text{Randomly round } y_i' \in [-1,1] \text{ to } y_i \in \{-1,1\}. \tag{1.4}$$

In Step 1 (where again r is a random Gaussian vector), the algorithm may try different values for T. The meaning of Step 2 is that we take y_i to be 1 with probability $\frac{1}{2}+\frac{1}{2}y_i'$ and -1 otherwise (so $\mathbf{E}[y_i] = y_i'$), independently for each i. Note that we may as well view the $[-1,1]$ values $\{y_i'\}$ as the final "solution," since with respect to Step 2,

$$\mathbf{E}\left[\tfrac{1}{2}+\tfrac{1}{2}\textstyle\sum(-a_{ij})(y_i \cdot y_j)\right] = \tfrac{1}{2}+\tfrac{1}{2}\textstyle\sum(-a_{ij})(y_i' \cdot y_j')\,.$$

Feige and Langberg believed the rounding functions used in Step 1, which they called "s-linear functions," were "close to being [optimal]" given their rounding technique.

Since the trivial random algorithm finds cuts of value at least $1/2$, it makes sense to measure the performance of a Max-Cut approximation algorithm in terms of how much more than $1/2$ it can guarantee. Indeed, Håstad and Venkatesh [25] suggest measuring the performance of algorithms for CSPs by looking at how well they approximate the *gain* over a random solution; Håstad and Venkatesh were particularly interested in this question for the Max-Lin (mod 2) problem, of which Max-Cut is a special case. To that end, we define the "Max-Cut-Gain" approximation problem as follows: Given a weighted graph with maximum cut $1/2+\varepsilon$, find a cut of value $1/2+\alpha\varepsilon$ for α as large as possible.

Neither Zwick [45] nor Feige–Langberg [19] provided any analysis of their algorithms in the Max-Cut-Gain framework. However, Charikar and Wirth [12] subsequently gave an analysis of the Feige–Langberg algorithm showing that when $T = \Theta(\sqrt{\log(1/\varepsilon)})$, it has the following guarantee:

- Given a graph where (1.2) is at least $1/2+\varepsilon$ (e. g., if the maximum cut (1.1) is at least $1/2+\varepsilon$), the algorithm finds a cut of value at least $1/2+\Omega(\varepsilon/\log(1/\varepsilon))$.

As for hardness of Max-Cut-Gain, the $16/17$ NP-hardness result can be trivially translated into NP-hardness of finding cuts of value $1/2+(11/13)\varepsilon$ in graphs with maximum cut $1/2+\varepsilon$. The methods of [28] slightly improve this to $1/2+(2/\pi)\varepsilon$, assuming UGC.

1.3 Our main results—informal statement and discussion

The first two main results of this paper can now be stated.

Main Result 1: *(For details, see Theorem 4.1.)* For each $\varepsilon > 0$, the Goemans–Williamson SDP for Max-Cut-Gain has an "integrality gap" of $1/2+\varepsilon$ vs. $1/2+O(\varepsilon/\log(1/\varepsilon))$. In other words, there are nonnegative weights a_{ij} summing to 1 such that (1.2) has value at least $1/2+\varepsilon$ but (1.1) has value at most $1/2+O(\varepsilon/\log(1/\varepsilon))$. Thus the Charikar-Wirth SDP-rounding algorithm is essentially optimal.

Main Result 2: *(For details, see Theorems 4.2, 4.3.)* The above SDP gap can be translated into an equivalent "Long Code test." As a consequence, we get that there exists a constant C such that for each $\varepsilon > 0$, given a graph with maximum cut $1/2+\varepsilon$, it is UGC-hard to find a cut of value at least $1/2+C\varepsilon/\log(1/\varepsilon)$. In other words, beating the Charikar-Wirth Max-Cut-Gain guarantee with *any* efficient algorithm is NP-hard, assuming UGC.

Recall that [28] proved sharp UGC-hardness in the "high end"—maximum cut near 1—and also sharp UGC-hardness of .878-factor approximation. Since our second theorem proves sharp UGC-hardness in the "low end"—maximum cut near $1/2$—we consider the question of Max-Cut's approximability to be qualitatively completely settled, up to UGC.

We view these two results as an interesting continuation of the flurry of work in the last four years on 2-bit constraint satisfaction problems such as Vertex-Cover, Max-Cut and Sparsest-Cut. This recent work has made intriguing connections among the following topics:

- semidefinite programming (SDP) algorithms [6, 12, 26, 1, 3, 2, 11];
- SDP integrality gaps [32, 29, 33];

- Fourier analysis-based hardness results [31, 28, 13, 37, 18] subject to the Unique Games Conjecture (UGC) [27].

In particular, within these papers we see SDP rounding algorithms, SDP gap constructions, and Fourier analysis results all motivating one another. It was not until recently—two years subsequent to the initial conference publication of this work—that the full extent of these relationships became clear. (See Section 1.5.)

The main theme in the present paper is the illustration of how SDP gaps for 2-bit constraint satisfaction problems arise naturally in *Gaussian space*, and how these SDP gaps can be naturally translated into Long Code tests and UGC-hardness results.

1.4 Related problems: Correlation-Clustering and Quadratic-Programming

Max-Cut-Gain is a special case of an algorithmic problem called (weighted) Correlation-Clustering. The unweighted (i. e., all edges having equal weight) version of the problem, introduced by Bansal, Blum and Chawla [7], has the following setting: Given is an unweighted graph G with some edges labeled "similar" and the remaining edges labeled "dissimilar." The goal is to partition the vertices into any number of "clusters" with the idea that edges labeled "similar" should be contained within clusters and edges labeled "dissimilar" should straddle clusters. The authors of [7] considered three different goals for a solution: "MaxAgree," namely maximizing the number of correctly positioned edges, "MinDisagree," minimizing the number of incorrectly positioned edges, and "MaxCorr," maximizing the "correlation"—i. e., the difference of the number of correctly positioned edges and incorrectly positioned edges.

Although the approximability of the MaxAgree and MinDisagree versions became fairly well understood, progress on the approximability of the MaxCorr was not made until the paper of Charikar and Wirth [12]. Charikar-Wirth first showed that up to a constant factor (at most 3), partitions into just two clusters are as good as partitions into arbitrary numbers of clusters. Thus not much is lost by restricting to the two-cluster version of Correlation-Clustering. It is easy to see that the weighted two-cluster version of Correlation-Clustering is essentially the same as the Quadratic-Programming problem: Given a matrix of weights $A = (a_{ij})$,

$$\begin{aligned} \max \quad & \sum_{i,j=1}^{n} (a_{ij})\,(y_i \cdot y_j) \\ \text{subject to:} \quad & y_i \in \{-1, 1\}\,. \end{aligned} \tag{1.5}$$

To see the equivalence to two-cluster Correlation-Clustering, think of the y_i as indicating which cluster i is in and think of the positive weights a_{ij} as measuring similarity and the negative weights a_{ij} measuring dissimilarity. Also, note that Max-Cut-Gain is the special case of Quadratic-Programming in which all the weights are nonpositive.

Quadratic-Programming was shown to admit an $\Omega(1/\log n)$-approximation algorithm in works of Nesterov [39], Nemirovski, Roos, and Terlaky [38], and Megretski [36]; as described in Charikar and Wirth [12], running the Feige–Langberg rounding procedure with $T = \Theta(\sqrt{\log n})$ on the natural SDP relaxation (1.5) yields an $\Omega(1/\log n)$-approximation algorithm. Thus Correlation-Clustering also has an $\Omega(1/\log n)$-approximation algorithm.

On the hardness side, Arora, Berger, Hazan, Kindler and Safra [5] showed that giving a $(1/\log^{\gamma_0} n)$-approximation is hard for some universal $\gamma_0 > 0$ unless NP is contained in quasipolynomial time. The

instances constructed in that paper use heavily skewed positive and negative weights and are thus not relevant for the Max-Cut-Gain problem. Alon, Makarychev, Makarychev, and Naor [2] showed a tight SDP gap for Quadratic-Programming of $\Omega(\log n)$. Their proof, however, was completely nonconstructive; they used duality to argue that such a gap existed without giving any explicit instance. They gave as an open problem the question of finding an explicit instance demonstrating the $\Omega(\log n)$ gap. The work [5] gives a fairly complicated construction showing an $\Omega(\log n/\log\log n)$ gap.

The third main result in this paper is the following:

Main Result 3: *(For details, see Theorem 4.4.)* There is a relatively simple and essentially explicit $\Omega(\log n)$ SDP gap for Quadratic-Programming, based on "randomness reduction" of the $1/2+\varepsilon$ vs. $1/2+O(\varepsilon/\log(1/\varepsilon))$ SDP gap we prove for Max-Cut-Gain.

Finally, the two-cluster Correlation-Clustering problem is of special interest when the underlying graph is bipartite. Following Alon, Makarychev, Makarychev, and Naor [2], we call the weighted version of this problem $K_{N,N}$-Quadratic-Programming:

$$\max \quad \sum_{i,j=1}^{n} (a_{ij})\,(y_i \cdot z_j) \tag{1.6}$$

$$\text{subject to:} \quad y_i \in \{-1,1\}, z_j \in \{-1,1\}\,.$$

This problem is equivalent to that of computing the $\|\cdot\|_{\infty\mapsto 1}$ norm of the matrix A, which is closely related to its "cut norm." See Alon and Naor [3] for a discussion of the algorithmic significance of this problem. Alon and Naor studied approximation algorithms for this problem and noted that the Grothendieck inequality [22] from the theory of Banach spaces is nothing more than a constant upper bound on the SDP gap of (1.6). The supremum of the SDP gaps of all instances of (1.6) is called the *Grothendieck constant*. Alon and Naor took the best known upper bound on Grothendieck's constant—$K_{\text{Krivine}} = \pi/(2\ln(1+\sqrt{2})) \approx 1.78$, due to Krivine [34]—and translated it into a $(1/K_{\text{Krivine}})$-approximation algorithm for the problem of approximating $\|A\|_{\infty\mapsto 1}$.

The last main theorem of our paper is the analogue of this for lower bounds/hardness:

Main Result 4: *(For details, see Theorems 4.5, 4.6.)* The best known lower bound on Grothendieck's constant—namely, $K_{\text{Davie–Reeds}} \approx 1.67$, due independently to Davie [17] (see [23]) and Reeds [43]—can be translated into a UGC-hardness result for $(1/K_{\text{Davie–Reeds}})$-approximating the problem $K_{N,N}$-Quadratic-Programming.

1.5 Subsequent work

Subsequent to the 2006 publication of the extended abstract of this work [30], research has progressed on the connections between SDP algorithms and gaps, Fourier analysis, and Long Code test described in Section 1.3. Two papers appearing recently are particularly relevant. In [40], O'Donnell and Wu continue the research in the present paper. Using a combination of the von Neumann Minimax theorem

and Borell's theorem on Gaussian rearrangement [10], the authors show that the Gaussian graphs used to prove our two main theorems are indeed the "worst" graphs. More specifically, they show that the optimal SDP gaps and UGC-hardness for Max-Cut-Gain come from negative probability operators of the form $-(pT_1 + (1-p)T_{\rho_0})$, where $p \in [0,1]$, $\rho_0 \in [-1,0]$—just as in the present paper's Theorem 5.5. This allows O'Donnell and Wu to determine, for every c, the *precise* number s such that c vs. s is the optimal SDP gap and UGC-hardness for Max-Cut-Gain. In [41], Raghavendra significantly extends the methods in these papers, showing (roughly speaking) that for *all* constraint satisfaction problems, the optimal SDP gaps and UGC-hardness match. The only downside of this result is that it is somewhat inexplicit; determining *what* the matching values are is not so easy. In particular, to determine the best s for a given c up to an additive δ, Raghavendra gives only an algorithm taking doubly-exponential time in δ.

1.6 Outline of this paper

The remainder of the paper is organized as follows. In Section 2 we formally define the algorithmic problems we are interested in for this paper: Max-Cut-Gain, Quadratic-Programming, and $K_{N,N}$-Quadratic-Programming. We also describe the SDP relaxations for these problems, as well as a probabilistic rephrasing of them that allows us to consider instances on Gaussian space. In Section 3 we introduce the Fourier and Hermite expansion tools we will need, define "Long Code tests," and introduce the "Davie–Reeds operator" that provides the basis for our SDP gaps and UGC-hardness results. Section 4 is devoted to formal statements of our main results. Section 5 contains the proofs of our first two main results, for Max-Cut-Gain. Section 6 gives the proof of our "randomness reduction" for the Quadratic-Programming SDP. Finally, Section 7 proves our UGC-hardness result for $K_{N,N}$-Quadratic-Programming.

2 Problem definitions

In this section we give formal definitions of the problems studied in this paper.

2.1 Algorithmic problems

The most general algorithmic problem we are concerned with is Quadratic-Programming:

Definition 2.1. The Quadratic-Programming *problem* is the following: Given a symmetric real matrix $A = (a_{ij})$, compute

$$\begin{aligned} \max \quad & \sum_{i,j} (a_{ij})\,(y_i \cdot y_j) \\ \textit{subject to:} \quad & y_i \in [-1,1]\,. \end{aligned}$$

Convention: We will always assume that $\sum_{i,j} |a_{ij}| = 1$. This is without any loss of generality because we can scale all entries a_{ij} by the same positive constant.

The assumption that A is symmetric is without loss of generality. Also, our definition of Quadratic-Programming has a slight difference from the definition given in [12, 5, 2]: namely, we allow the

variables to take values in $[-1,1]$, not just $\{-1,1\}$, and we also allow nonzero diagonal entries in A, corresponding to "self-loops" in the graph in the Correlation-Clustering problem. We believe our definition is mathematically more natural, and in any case, the two definitions are essentially equivalent for all intents and purposes—see Section 2.3.

We are interested in two special cases of Quadratic-Programming. The first is our main problem of interest, Max-Cut-Gain:

Definition 2.2. The Max-Cut-Gain *problem* is the special case of Quadratic-Programming in which all the a_{ij} are nonpositive. This corresponds to the graphical instance of Max-Cut-Gain in which the weight on edge (i,j) is $-a_{ij}/2$.

We remark that Crescenzi, Silvestri and Trevisan [16] showed that the weighted and unweighted versions of Max-Cut-Gain have the same polynomial-time approximability up to an additive $1/\mathrm{poly}(n)$.

We also define the bipartite special case of the Quadratic-Programming problem, investigated by Alon and Naor [3] using the Grothendieck Inequality. This is equivalent to the problem of computing $\|A\|_{\infty\mapsto 1}$.

Definition 2.3. The $K_{N,N}$-QUADRATIC-PROGRAMMING *problem* is the following: Given a real matrix $A=(a_{ij})$, compute

$$\max \quad \sum_{i,j}(a_{ij})\,(y_i\cdot z_j) \tag{2.1}$$

$$\textit{subject to:} \quad y_i, z_i \in [-1,1]\,.$$

It is easy to check that the optimizers in this problem are always in $\{-1,1\}$ and thus the issue of "zeros on the diagonal" is irrelevant.

2.2 SDP relaxations, and the probabilistic viewpoint

We begin this section by defining the semidefinite programming (SDP) relaxation of the Quadratic-Programming problem.

Definition 2.4. For $d\in\mathbb{N}$, we define B_d to be the d-dimensional unit ball, $B_d=\{x\in\mathbb{R}^d : \|x\|_2\le 1\}$. In particular, $B_1=[-1,1]$.

Note that the Quadratic-Programming problem can be written as:

$$\max \quad \sum_{i,j=1}^{n}(a_{ij})\,(y_i\cdot y_j) \tag{2.2}$$

$$\textit{subject to:} \quad y_i \in B_1\,.$$

Definition 2.5. The *d-dimensional SDP relaxation* of (2.2) is defined to be:

$$\max \quad \sum_{i,j=1}^{n} (a_{ij})\,(y_i \cdot y_j) \tag{2.3}$$

$$\textit{subject to:} \quad y_i \in B_d.$$

If unspecified, the value d is assumed to be n.

The interest in the SDP relaxation is that if $d = n$, it can be solved in polynomial time (strictly speaking, it can be solved to within additive error $2^{-\mathrm{poly}(n)}$; see [20]). Thus we have the following strategy for approximating (2.2): solve the SDP relaxation and then try to "round" the optimal vectors y_i^* into reals y_i in $[-1,1]$ such that the value of the quadratic form does not go down too much. This strategy motivates the definition of "SDP integrality gaps":

Definition 2.6. Given the matrix A, we say that the SDP has an *SDP (integrality) gap* of κ vs. κ' if the value of (2.3) is at least κ and the value of (2.2) is at most κ'. The associated *SDP gap ratio* is defined to be κ'/κ. The *SDP gap of the problem* Quadratic-Programming is defined to be the worst SDP gap ratio over all possible inputs A. We make the same definitions for Max-Cut-Gain and $K_{N,N}$-Quadratic-Programming.

The input instances A we construct in exhibiting SDP gaps for Max-Cut-Gain and Quadratic-Programming are most naturally set in *Gaussian space*. One can think of this setting as the one in which the matrices A are *infinite*, their "coordinates" are indexed by points in $\mathbb{R}^n$, and their coordinates are also "weighted" according to the Gaussian probability distribution. The most natural way to allow for this is to rephrase the Quadratic-Programming problem and its SDP relaxation in a probabilistic manner. In the remainder of this section we therefore give definitions for the *probabilistic versions of* Quadratic-Programming *and* Max-Cut-Gain .

Making these definitions requires some preparation. Let (X,μ) be any probability space. The three cases of interest to us are $X = \{1,\ldots,N\}$ with μ the uniform distribution, $X = \{-1,1\}^n$ with μ the uniform distribution, and $X = \mathbb{R}^n$ with $\mu = \gamma$, the n-dimensional Gaussian probability distribution. We consider the set of functions $f : X \to \mathbb{R}$ as an inner product space, with

$$\langle f,g\rangle = \mathop{\mathbf{E}}_{x \xleftarrow{\mu} X}[f(x)g(x)],$$

We will sometimes write $\int_X F(x)\,d\mu(x)$ in place of $\mathbf{E}_{x \xleftarrow{\mu} X}[F(x)]$. We also write $\|f\|_2 = \sqrt{\langle f,f\rangle}$.

Definition 2.7. An instance of the *probabilistic version of* Quadratic-Programming is a (bounded) linear operator A on the above inner product space, and the associated problem is to determine

$$\sup_{f:X\to[-1,1]} \langle f, Af\rangle. \tag{2.4}$$

Note that in the case of $X = \{1,\ldots,N\}$ this is the same as the Quadratic-Programming originally defined, after scaling (the matrix) A by a factor of N.

Definition 2.8. The *probabilistic version of* $K_{N,N}$-Quadratic-Programming is to determine

$$\sup_{f,g:X\to[-1,1]} \langle f, Ag\rangle .$$

Finally, to define the probabilistic version of Max-Cut-Gain, we require the following:

Definition 2.9. A *probability operator* X is any linear operator B satisfying the following two conditions: $Bf \geq 0$ for all $f \geq 0$, and $\langle \mathbf{1}, B\mathbf{1}\rangle = 1$.

In the simple case of $X = \{1,\dots,N\}$, a probability operator is precisely a nonnegative matrix with entries summing to 1. We therefore make the following definition:

Definition 2.10. The *probabilistic version of* Max-Cut-Gain is defined to be the special case of Quadratic-Programming in which A is the negative of a probability operator.

To define the probabilistic version of the SDP for Quadratic-Programming, we need to introduce a few more notions. First, we extend the inner product space defined above to functions $f : X \to \mathbb{R}^d$ by defining

$$\langle f, g\rangle = \mathop{\mathbf{E}}_{x\overset{\mu}{\leftarrow}X}[\langle f(x), g(x)\rangle],$$

where the $\langle\cdot,\cdot\rangle$ inside the expectation is just the usual inner product in $\mathbb{R}^d$. Next:

Definition 2.11. Given a linear operator A on functions $X \to \mathbb{R}$, we say its *component-wise* version is the linear operator $A^{(d)}$ on functions $f : X \to \mathbb{R}^d$ given by applying A component-wise on the d coordinate functions of f.

We now can define the probabilistic version of the SDP:

Definition 2.12. The *probabilistic d-dimensional SDP relaxation of* (2.4) is defined to be

$$\sup_{f:X\to B_d} \langle f, A^{(d)} f\rangle . \tag{2.5}$$

We make the analogous definitions also for the probabilistic SDP relaxations of Max-Cut-Gain and $K_{N,N}$-Quadratic-Programming. When $X = \mathbb{R}^n$ with the Gaussian measure and d is not mentioned, we assume $d = n$.

Regarding the possibility of an infinite domain X, the reader can be assured of the following facts: First, for any constants κ and κ', a probabilistic SDP gap of κ vs. κ' in the Gaussian space setting for any of our problems implies the same gap in the usual $\{1,\dots,N\}$ setup for sufficiently large N, by a limiting argument. Furthermore: a) our Long Code test and UGC-hardness result for Max-Cut-Gain take place entirely in the finite setting of $X = \{-1,1\}^n$, using the Gaussian space setting only for motivation; b) our third theorem on "randomness reduction" shows how to effectively convert an ε vs. $O(\varepsilon/\log(1/\varepsilon))$ SDP gap in Gaussian space into one with the same parameters over $\{1,\dots,N\}$ with $N = \text{poly}(1/\varepsilon)$. We stress that the extension to the Gaussian space framework is made because this is the setting in which our SDP gaps naturally occur.

2.3 A slight annoyance: zeros on the diagonal

In this section we consider a minor technical point—the issue of "self-loops," or "zeros on the diagonal."

As noted, our definition of Quadratic-Programming, Definition 2.1, is slightly different from the definition of the problem used by [12, 5, 2]. In their definition, the variables are maximized only over $\{-1,1\}$, and the diagonal entries of the matrix A must be zero. Let us call this version of the problem "Restricted-Quadratic-Programming."

Let us compare the two definitions. First, note that if A has zero diagonal entries then the definitions are completely equivalent: given optimal (y_i^*) for our problem, rounding them randomly as in the Feige–Langberg algorithm yields a solution in $\{-1,1\}$ that is equally good in expectation, and thus there exists maximizers in $\{-1,1\}$ with the same value.

On the other hand, if nonzero diagonal entries are allowed then the maximum over $y_i \in \{-1,1\}$ can be negative, leading to trouble in defining approximation guarantees. However with our definition the solution $y_i \equiv 0$ is always allowed, showing that the maximum is always nonnegative. We believe this makes our definition more mathematically natural.

In any case, the definitions are essentially equivalent for all intents and purposes, as the following proposition (whose trick appears implicitly in [5]) shows:

Proposition 2.13. *There is an efficient reduction mapping* Quadratic-Programming *instances to* Restricted-Quadratic-Programming *instances on M^2 times as many variables under which the SDP value and the actual value change by at most an additive $O(1/M)$.*

Proof. Suppose we have an instance of Quadratic-Programming, with our convention that $\sum |a_{ij}| = 1$ (which is without loss of generality). Replace each variable i with M copies of itself. Split the old weights a_{ij} into M^2 new weights in the natural way: put weight a_{ij}/M^2 between each new i-variable and new j-variable when $i \neq j$; and, put weight $2a_{ii}/M^2$ between each pair of distinct i-variables and weight a_{ii}/M^2 on each i-variable's self-loop. This gives a new instance of Quadratic-Programming and it is easy to see that it has the same value and SDP value as the old one. Indeed, any solution (y_i) in the old instance can be duplicated into an equal-value solution in the new instance, and any solution (y_i) in the new instance can be averaged into an equal-value solution in the old instance (SDP value or real value). But note that if the total weight on the diagonal, $\sum |a_{ii}|$ was α in the old instance, the new instance has only total weight $\alpha/M \leq 1/M$ on its diagonal. If we simply delete this diagonal weight (and rescale slightly) we get an instance of Restricted-Quadratic-Programming, and this deletion changes the value of any solution by at most an additive $O(1/M)$. □

From this, we conclude that the Restricted-Quadratic-Programming and Quadratic-Programming problems have identical SDP gaps and also the same polynomial-time approximability up to arbitrarily small inverse-polynomial additive factors.

3 Long Code tests and Fourier analysis

3.1 Fourier and Hermite expansions

We begin by defining the notions we need for Fourier analysis of Boolean functions and Hermite analysis of functions on Gaussian space.

Definition 3.1. Recall the inner product space on functions $f : \{-1,1\}^n \to \mathbb{R}$ defined in Section 2.2. Under this definition, the set of functions $(\chi_S)_{S\subseteq[n]}$ defined by $\chi_S(x_1,\ldots,x_n) = \prod_{i\in S} x_i$ forms an orthonormal basis, and thus we have the *Fourier expansion* of any $f : \{-1,1\}^n \to \mathbb{R}$,

$$f = \sum_{S\subseteq[n]} \hat{f}(S)\chi_S\,,$$

where $\hat{f}(S) = \mathbf{E}[f\chi_S]$. The functions $\chi_i = \pm\chi_{\{i\}}$ are called the *dictator functions* or *Long Codes*. We also have the Plancherel identity,

$$\langle f,g\rangle = \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\,.$$

We define the *projection to level k operator* P_k (for $0 \le k \le n$) and the *Bonami-Beckner operator* T_ρ (for $0 \le \rho \le 1$) by

$$P_k f = \sum_{|S|=k} \hat{f}(S)\chi_S, \quad T_\rho f = \sum_{k=0}^{n} \rho^k P_k\,.$$

The Bonami-Beckner operator also acts as $(T_\rho f)(x) = \mathbf{E}_y[f(y)]$, where the string y is formed by letting $y_i = x_i$ with probability ρ and letting y_i be uniformly random otherwise, independently for each i. Thus the Bonami-Beckner operator is a probability operator.

Definition 3.2. Recall the inner product space on functions $f : (X,\gamma) \to \mathbb{R}^d$ defined in Section 2.2, where $X = \mathbb{R}^n$ and γ is the n-dimensional Gaussian probability distribution. Under this definition there is a complete set of orthonormal polynomials $\mathbb{R}^n \to \mathbb{R}$ called the *Hermite polynomials*, $(H_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}^n}$ (see, e. g., the book of Ledoux and Talagrand [35]). We write $|\mathbf{k}| = \mathbf{k}_1 + \cdots + \mathbf{k}_n$. Any function $f : \mathbb{R}^n \to B_d$ has a *Hermite expansion*,

$$f = \sum_{\mathbf{k}\in\mathbb{N}^n} \hat{f}(\mathbf{k})H_{\mathbf{k}}\,,$$

where the Hermite coefficients $\hat{f}(\mathbf{k}) = \int_{\mathbb{R}^n} f(x)H_{\mathbf{k}}(x)\,d\gamma(x)$ are in B_d. We have the Plancherel identity

$$\langle f,g\rangle = \sum_{\mathbf{k}\in\mathbb{N}^n} \langle \hat{f}(\mathbf{k}),\hat{g}(\mathbf{k})\rangle\,.$$

We define the *projection to level k operator* P_k (for $k \in \mathbb{N}$) and the *Ornstein-Uhlenbeck operator* T_ρ (for $0 \le \rho \le 1$) by

$$P_k f = \sum_{|\mathbf{k}|=k} \hat{f}(S)H_{\mathbf{k}}, \quad T_\rho f = \sum_{k\ge 0} \rho^k P_k\,.$$

As $H_{\mathbf{0}}(x) = 1$ and $H_{\mathbf{e_i}}(x) = x_i$, we have

$$P_0 f = \mathbf{E}[f], \quad (P_1 f)(x) = \int_{\mathbb{R}^n} \langle x,y\rangle f(y)\,d\gamma(y)\,.$$

We also have that

$$(T_\rho f)(x) = \int_{\mathbb{R}^n} f(\rho x + \sqrt{1-\rho^2}y)\,d\gamma(y)\,,$$

so the Ornstein-Uhlenbeck operator is a probability operator. From these facts we also see that P_0, P_1, and T_ρ are component-wise operators.

3.2 Long Code tests

Our second main theorem in this paper is a hardness result for Max-Cut-Gain. Being a "2-variable constraint satisfaction problem," it seems very difficult to prove a sharp inapproximability NP-hardness result for it using current techniques (see Khot [27]). Thus we prove a UGC-hardness result; i. e., we reduce the "Unique Label Cover" problem to that of approximating Max-Cut-Gain. The paper [28] provides a relatively clean template for such reductions in its UGC-hardness result for Max-Cut; the reduction is through the use of gadgets known as "Long Code tests." (Such reductions originate in the work of Bellare, Goldreich, and Sudan [8] and were developed significantly by the work of Håstad [24].)

Long Code tests are usually defined as property testing algorithms for testing whether a given function $f : \{-1,1\}^n \to \{-1,1\}$ is correlated to a "Long Code" or "dictator"—i. e., $\pm\chi_i$ for some i. For our purposes, however, we can think of a Long Code test as something like an instance A demonstrating a probabilistic SDP integrality gap in the setting $X = \{-1,1\}^n$. The difference is that we don't compare $\langle f, Af\rangle$ for functions $f : \{-1,1\}^n \to B_d$ and $f : \{-1,1\}^n \to [-1,1]$. Instead we compare *dictator functions* $f = \pm\chi_i$ versus functions $f : \{-1,1\}^n \to [-1,1]$ that are *far from being dictators*. Note that a function f is far from being a dictator if $|\hat{f}(i)|$ is small for all i. We make the following definition:

Definition 3.3. In the context of Quadratic-Programming and its subproblems, a linear operator A on functions $f : \{-1,1\}^n \to [-1,1]$ is said to be a *Long Code test with completeness c and soundness s* if:

$$\langle \chi_i, A\chi_i\rangle \geq c \qquad \text{for all } i,$$

$$\text{and} \sup_{\substack{f:\{-1,1\}^n\to[-1,1]\\ \forall i\, |\hat{f}(i)|\leq\varepsilon}} \langle f, Af\rangle \leq s + \delta(\varepsilon)\,, \tag{3.1}$$

where $\delta(\varepsilon)$ is a function such that $\delta(\varepsilon) \to 0$ when $\varepsilon \to 0$.

As a rule of thumb, given a c vs. s Long Code test for a 2-query constraint satisfaction problem, the techniques in [28] tend to yield UGC-hardness of finding solutions of value at least $s+\delta$ on inputs with optimum value at least $c-\delta$, for every $\delta > 0$. We will see that this applies in our case of Max-Cut-Gain, and thus to get our sharp ε vs. $O(\varepsilon/\log(1/\varepsilon))$ UGC-hardness for approximating Max-Cut-Gain it will suffice for us to construct a Long Code test with these parameters.

Indeed, the methodology of [28] can even work in a more relaxed setting, where instead of showing (3.1), one only needs to show

$$\sup_{\substack{f:\{-1,1\}^n\to[-1,1]\\ \mathrm{Inf}_i^{\leq k(\delta)}(f)\leq\varepsilon}} \langle f, Af\rangle \leq s + \delta(\varepsilon)\,,$$

where

$$\mathrm{Inf}_i^{\leq k}(f) := \sum_{|S|\leq k, k\ni i} \hat{f}(S)^2$$

and $k(\delta) < \infty$ is a function independent of n. We will use this relaxation in our UGC-hardness result for $K_{N,N}$-Quadratic-Programming.

3.3 The Davie–Reeds operator

We now define a linear operator that plays a crucial role in all of our results:

Definition 3.4. For each $0 < \lambda < 1$, define the *Davie–Reeds operator* DR_λ on functions $f : (X, \mu) \to \mathbb{R}^d$ by $DR_\lambda = P_1 - \lambda \cdot id$. This definition makes sense both for $X = \{-1,1\}^n$ and μ the uniform distribution, and for $X = \mathbb{R}^n$ and $\mu = \gamma$, the n-dimensional Gaussian distribution. Indeed, as we will see in Section 5.1, this definition makes sense in an even more general context.

Davie [17] and Reeds [43] used this operator (with a suitable value of λ) to give the best known lower bound on the integrality gap of the $K_{N,N}$-Quadratic-Programming SDP (2.1) (i. e., the Grothendieck constant). See Section 7 for further details.

Although a Davie–Reeds operator is not a negative probability operator, it can be closely related to one; we will use it to give our SDP gaps for Max-Cut-Gain and also our Long Code test for this problem. Our "randomness reduction" will also be for Quadratic-Programming instances with Davie–Reeds operators, and our UGC-hardness result for $K_{N,N}$-Quadratic-Programming relies on the Davie–Reeds SDP.

Note that in general, $id = \sum_{k \geq 0} P_k$. Thus we can also write the Davie–Reeds operator DR_λ as

$$DR_\lambda f = -\lambda P_0 + (1-\lambda)P_1 - \lambda P_2 - \lambda P_3 - \cdots .$$

Intuitively, the Davie–Reeds operator mostly keeps the "linear" part of a function and also "negatively highlights" the nonlinear parts of a function. Thus it makes sense that it plays a useful role in Long Code tests, where the functions to be distinguished, dictator functions $\pm\chi$, are precisely the linear $\{-1,1\}$-valued functions.

4 Formal statements of our results

In this section we formally state our main results. They illustrate the main theme of the paper—natural SDP gaps on Gaussian space yielding Long Code tests yielding UGC-hardness results.

First, an SDP gap on Gaussian space for a Max-Cut-Gain operator:

Theorem 4.1. *Let $\varepsilon > 0$ and set $d = \mathrm{poly}(1/\varepsilon)$. Let A denote either the Davie–Reeds operator $DR_{1-\varepsilon}$ or the negative probability operator A_ε defined in Section 5.4. Then we have the following SDP gap on d-dimensional Gaussian space:*

$$\sup_{f:(\mathbb{R}^d,\gamma)\to B_d} \langle f, Af \rangle \geq \Omega(\varepsilon),$$

$$\text{and} \quad \sup_{f:(\mathbb{R}^d,\gamma)\to[-1,1]} \langle f, Af \rangle \leq O(\varepsilon/\log(1/\varepsilon)).$$

For negative probability operators—i. e., those operators used in the Max-Cut-Gain problem—this tradeoff is optimal, by the SDP rounding algorithm of Charikar and Wirth [12]. The key part of this theorem is the study of $\langle f, DR_{1-\varepsilon} f \rangle$ for functions $f : (\mathbb{R}^d, \gamma) \to [-1,1]$. Interestingly, our analysis suggests that the optimal functions f should be close to the form $f(x) = \mathrm{round}_T(\langle x, r \rangle)$ for some vector r, where

round$_T$ is as in the Feige–Langberg rounding algorithm (1.3). This would corroborate a suggestion made in [19], that such functions seem to be close to optimal for rounding the SDP.

We next show that this gap can be translated into an equivalent Long Code test.

Theorem 4.2. *There exist universal constants $C, c > 0$ such that the following holds. Let $\varepsilon > 0$ and assume $n \geq C\log(1/\varepsilon)$. Let A be as in Theorem 4.1. Then we have the following Long Code test on $\{-1,1\}^n$:*

$$\langle \chi_i, A\chi_i \rangle \geq \Omega(\varepsilon) \quad \text{for all } i \in [n],$$

$$\text{and} \quad \sup_{\substack{f:\{-1,1\}^n \to [-1,1] \\ \forall i\, |\hat{f}(i)| \leq c/\log(1/\varepsilon)}} \langle f, Af \rangle \leq O(\varepsilon/\log(1/\varepsilon)).$$

Using the Unique Label Cover reduction from [28], we easily obtain the following:

Theorem 4.3. *Assume UGC holds. Then for all (sufficiently small) constants $\varepsilon > 0$, it is NP-hard to distinguish* Max-Cut-Gain *instances with gain at least ε from instances with gain at most $O(\varepsilon/\log(1/\varepsilon))$.*

Again, because of [12] this result is optimal up to constant factors. It also closes all qualitative aspects of the hardness of Max-Cut subject to UGC: from [28] we have hardness of $1-\varepsilon$ versus $1-\Theta(\sqrt{\varepsilon})$ at the "high end" and also .878 factor hardness (both sharp due to [20]); Theorem 4.3 gives the sharp "low end" tradeoff, $1/2+\varepsilon$ versus $1/2+O(\varepsilon/\log(1/\varepsilon))$.

Regarding Quadratic-Programming, we show that we can perform "randomness reduction" on the SDP gap of Theorem 4.1 for $DR_{1-\varepsilon}$ on Gaussian space; i. e., this SDP gap can be highly efficiently "discretized." In particular, we can discretize to $\text{poly}(d) = \text{poly}(1/\varepsilon)$ many points and maintain the gap.

Theorem 4.4. *Let $\varepsilon > 0$ be sufficiently small, let $d = 1/\varepsilon^3$ and $N = \Theta(d^7)$. Let G be a set of N points drawn at random from d-dimensional Gaussian space and think of G as having the uniform probability distribution. Then for almost all choices of G (specifically, with probability at least $1 - 1/N$), we obtain an SDP gap on G with factor $\Omega(\log(1/\varepsilon)) = \Omega(\log N)$:*

$$\sup_{f:G \to B_d} \langle f, DR_{1-\varepsilon} f \rangle \geq \Omega(\varepsilon),$$

$$\text{and} \quad \sup_{f:G \to [-1,1]} \langle f, DR_{1-\varepsilon} f \rangle \leq O(\varepsilon/\log(1/\varepsilon)).$$

(Here we are using our general definition of the Davie–Reeds operator on arbitrary $L_2^{(d)}(\mathbb{R}^n, \mu)$ spaces; see Section 5.1 for more details.)

This SDP gap for Quadratic-Programming is optimal up to constant factors by the $O(\log N)$-approximation algorithm of [12]. The fact that Quadratic-Programming has an SDP gap of $\Omega(\log N)$ was first established by Alon, Makarychev, Makarychev, and Naor in [2]. However the argument used there was completely nonconstructive; they used duality to argue that an $\Omega(\log N)$ gap exists without explicitly giving any instance. They left the problem of describing an instance achieving the gap as an open problem. Arora, Berger, Hazan, Kindler, and Safra [5] gave a somewhat complicated construction achieving gap $\Omega((\log N)/(\log\log N))$. We believe our construction essentially resolves the problem by providing a natural and more or less explicit instance with gap $\Omega(\log N)$.

Finally, we describe our hardness result for the $K_{N,N}$-Quadratic-Programming problem. Here we show that the Davie–Reeds lower bound for Grothendieck's constant can be translated into a "two-function Long Code test" on $\{-1,1\}^n$:

Theorem 4.5. *For all $0 < \lambda < .4$, $d \geq 1$, and $\delta > 0$,*

$$\sup_{f,g:\{-1,1\}^n \to B_d} \langle f, DR_\lambda g\rangle = \langle \chi_i, DR_\lambda \chi_i\rangle = 1-\lambda \quad \text{for all } i \in [n],$$

$$\text{and} \quad \sup_{\substack{f,g:\{-1,1\}^n \to \{-1,1\} \\ \forall i\, |\hat{g}(i)| \leq \delta}} \langle f, DR_\lambda g\rangle \leq s(\lambda) + O(\delta).$$

Here $s(\lambda)$ is SDP "soundness" arising in the Davie–Reeds proof; see Section 7 for more details. This Long Code test is not yet sufficient for a UGC-hardness result because it only shows that if f and g pass the test with probability significantly more than $s(\lambda)$ then each has a coordinate with large low-degree influence. However we need f and g to *share* such a coordinate. This technical difficulty can be overcome, however, with a novel application of Green's "Szemerédi Regularity Lemma for $(\mathbb{Z}/2\mathbb{Z})^n$" [21]. Having done this, we get a UGC-hardness result for $K_{N,N}$-Quadratic-Programming:

Theorem 4.6. *Assume UGC holds. Then for every constant $\varepsilon > 0$ it is NP-hard to approximate the $K_{N,N}$-Quadratic-Programming problem to within factor $1/K_{\text{Davie–Reeds}} + \varepsilon \approx .597$.*

This result complements the SDP rounding algorithm of Alon and Naor [3] which uses Krivine's *upper* bound on Grothendieck's constant to give a $1/K_{\text{Krivine}} \approx .561$ approximation for $K_{N,N}$-Quadratic-Programming. The result also improves significantly on the $12/13 \approx .923$ NP-hardness result for $K_{N,N}$-Quadratic-Programming given in [3], albeit only under UGC.

5 Quadratic-Programming **and** Max-Cut-Gain

5.1 Elementary identities

In this section we give some elementary identities regarding the operators P_1 and $DR_{1-\varepsilon}$. Most of the identities hold completely formally from the definitions and work in the setting of functions $\mathbb{R}^n \to \mathbb{R}^d$. Let us give some general definitions:

Definition 5.1. Let μ be any probability distribution on $\mathbb{R}^n$, discrete or continuous. We write $\mathbf{E}[\cdot]$ for integration with respect to μ. For functions $f : \mathbb{R}^n \to \mathbb{R}^d$ in the inner product space $L_2^{(d)}(\mathbb{R}^n, \mu)$ we define the following:

- The ith "orthogonal coefficient" $\hat{f}(i) \in \mathbb{R}^d$, for $1 \leq i \leq n$, via
$$\hat{f}(i) = \mathop{\mathbf{E}}_{y}[y_i f(y)].$$
- The operator P_1 via
$$(P_1 f)(x) = \mathop{\mathbf{E}}_{y}[\langle x, y\rangle f(y)] = \sum_{i=1}^{n} \hat{f}(i) x_i.$$

- The "projection deficiency" operator Δ_1 via

$$\Delta_1 = P_1^2 - P_1 .$$

- The Davie–Reeds operator $DR_{1-\varepsilon}$ via

$$DR_{1-\varepsilon} f = P_1 f - (1-\varepsilon)\cdot id .$$

The following identities hold immediately from expanding the definitions:

Lemma 5.2. *For any f in $L_2^{(d)}(\mathbb{R}^n,\mu)$,*

$$\langle f, P_1 f\rangle = \sum_{i=1}^n \|\hat{f}(i)\|_2^2 \tag{5.1}$$

$$= \|f\|_2^2 - \|P_1 f - f\|_2^2 + \langle \Delta_1 f, f\rangle \tag{5.2}$$

$$\langle f, DR_{1-\varepsilon} f\rangle = \sum_{i=1}^n \|\hat{f}(i)\|_2^2 - (1-\varepsilon)\|f\|_2^2 \tag{5.3}$$

$$= \varepsilon \sum_{i=1}^n \|\hat{f}(i)\|_2^2 - (1-\varepsilon)\|P_1 f - f\|_2^2 + (1-\varepsilon)\langle \Delta_1 f, f\rangle \tag{5.4}$$

$$\langle \Delta_1 f, f\rangle = \sum_{i,j=1}^n (\mathbf{E}[x_i x_j] - \delta_{ij})\cdot \langle \hat{f}(i), \hat{f}(j)\rangle . \tag{5.5}$$

(Here δ_{ij} is the Kronecker delta.)

Our main interest is in the case that μ is either the n-dimensional Gaussian distribution or the uniform distribution on $\{-1,1\}^n$. (The general case is only necessary for the Gaussian SDP discretization argument in Section 6.) In these two cases we have $\mathbf{E}[x_i x_j] = \delta_{ij}$; hence $\langle \Delta_1 f, f\rangle = 0$ for every f and P_1 is indeed a projection operator ($P_1^2 = P_1$). This implies all the familiar identities from Fourier and Hermite analysis; e. g., $\langle f, P_1 f\rangle = \|P_1 f\|_2^2$.

5.2 SDP gap for $DR_{1-\varepsilon}$ on Gaussian space

In this section we show an ε versus $O(\varepsilon/\log(1/\varepsilon))$ SDP gap for the Davie–Reeds operator $DR_{1-\varepsilon}$ on Gaussian space. This yields the $A = DR_{1-\varepsilon}$ part of Theorem 4.1.

Theorem 5.3. *There is a universal constant C such that for every $\varepsilon > 0$ and $d \geq 1$ we have*

$$\sup_{f:(\mathbb{R}^d,\gamma)\to B_d} \langle f, DR_{1-\varepsilon} f\rangle \geq \varepsilon - C/d , \tag{5.6}$$

and

$$\sup_{f:(\mathbb{R}^d,\gamma)\to[-1,1]} \langle f, DR_{1-\varepsilon} f\rangle \leq C\varepsilon/\log(1/\varepsilon) . \tag{5.7}$$

Proof. To prove (5.6), take $f : (\mathbb{R}^d, \gamma) \to B_d$ to be the function $f(z) = z/\|z\|_2$. In this case,

$$\hat{f}(i) = \int y_i \frac{y}{\|y\|} \, d\gamma(y), \qquad \text{and hence} \quad \hat{f}(i)_j = \int \frac{y_i y_j}{\|y\|_2} \, d\gamma(y).$$

It is easy to see that for $j \neq i$ we have $\hat{f}(i)_j = 0$ (since the density function of y_j is even, say). Thus

$$\|\hat{f}(i)\|_2 = \int \frac{y_i^2}{\|y\|_2} \, d\gamma(y), \qquad \text{and hence} \quad \sum_{i=1}^{d} \|\hat{f}(i)\|_2 = \int \|y\|_2 \, d\gamma(y).$$

It is well known that the expected length of a random d-dimensional Gaussian is at least $\sqrt{d} - C/\sqrt{d}$, for some universal constant C (this also follows from Lemma 6.2). Thus we have

$$\sum_{i=1}^{d} \|\hat{f}(i)\|_2 \geq \sqrt{d} - C/\sqrt{d}.$$

But all summands here are in fact the same, by symmetry; thus

$$\langle f, P_1 f \rangle = \sum_{i=1}^{d} \|\hat{f}(i)\|_2^2 \geq (\sqrt{d} - C/\sqrt{d})^2/d \geq 1 - 2C/d,$$

where we used (5.1). Substituting this into (5.3) and using $\|f\|_2 \leq 1$ we get

$$\langle f, DR_1 f \rangle \geq \varepsilon - 2C/d$$

for our particular f, thus proving the "completeness" (5.6).

We now prove the "soundness" (5.7) using identity (5.4); note that the $\langle \Delta_1 f, f \rangle$ term is 0 in our Gaussian setting. Given any $f : (\mathbb{R}^d, \gamma) \to [-1, 1]$, the orthogonal coefficients $\hat{f}(i)$ are just d real numbers and $(P_1 f)(x)$ is just the linear form $\sum_{i=1}^{d} \hat{f}(i) x_i$ with the x_i's being independent standard one-dimensional Gaussians. Thus $P_1 f$ is distributed as a mean-zero one-dimensional Gaussian with variance equal to $\sigma^2 = \sum_{i=1}^{d} \hat{f}(i)^2$. Note that $\sigma^2 \leq \|f\|_2^2 \leq 1$, by (5.2). Using (5.4), we have

$$\langle f, DR_{1-\varepsilon} f \rangle = \varepsilon\sigma^2 - (1-\varepsilon)\|P_1 f - f\|_2^2. \tag{5.8}$$

Now whenever $|P_1 f| > 2$ there is a contribution of at least 1 to $\|P_1 f - f\|_2^2$, simply because f's values are in $[-1, 1]$. Hence by the "heavy tail" property of mean-zero one-dimensional Gaussians, i. e.,

$$\mathbf{Pr}[|P_1 f| > 2] \geq \exp(-C \cdot 2^2/\sigma^2)$$

for some universal constant C, we have

$$\langle f, DR_{1-\varepsilon} f \rangle \leq \varepsilon\sigma^2 - (1-\varepsilon)\exp(-4C/\sigma^2).$$

If $\sigma^2 \geq C'/\log(1/\varepsilon)$, where C' is a sufficiently large constant, then $(1-\varepsilon)\exp(-C/\sigma^2) \geq 2\varepsilon$; since $\sigma^2 \leq 1$ this would make $\langle f, DR_{1-\varepsilon} f \rangle$ negative. Thus we can assume $\sigma^2 \leq C'/\log(1/\varepsilon)$. But this means $\langle f, DR_{1-\varepsilon} f \rangle \leq C'\varepsilon/\log(1/\varepsilon)$ and so (5.7) is proved. □

Let us make a remark about the analysis of the soundness case. Our upper bound worked by first considering the linear part of f to be fixed; say, $P_1 f(x) = \langle x, r\rangle$ for some vector r with $\|r\|_2^2 = \sigma^2$. It then considered the "worst possible" values f could have in (5.8), assuming that these values could be chosen arbitrarily in $[-1,1]$. Of course they can't be, since $P_1 f$ was fixed. But if they could, it's clear from (5.8) that it's worst if $f(x)$ equals $P_1 f(x)$ whenever this value is in $[-1,1]$, and equals $\operatorname{sgn}(P_1 f(x))$ otherwise. Then optimizing σ^2 to $\Theta(1/\log(1/\varepsilon))$, we conclude that the worst functions "for our analysis" are of the form

$$f(z) = \operatorname{round}_T(\langle z, r\rangle),$$

where $T = \Theta(\sqrt{\log(1/\varepsilon)})$. Interestingly, this precisely mimics the SDP rounding algorithm for Quadratic-Programming/Max-Cut-Gain suggested by Feige and Langberg [19] and analyzed by Charikar and Wirth [12].

5.3 Long Code test with $DR_{1-\varepsilon}$ on the discrete cube

When $DR_{1-\varepsilon}$ operates on functions on the discrete cube $\{-1,1\}^n$ rather than functions on Gaussian space, it turns out that the "dictator" functions $\pm\chi_i$ become optimal for $\langle f, DR_{1-\varepsilon} f\rangle$. Thus the completeness bound (5.6) becomes simply ε even for $d = 1$, and therefore the soundness bound (5.7) no longer holds. However the soundness bound holds if we restrict to functions which have all the quantities $\hat{f}(i)^2$ small enough. (Note that $\hat{f}(i)^2 = \operatorname{Inf}_i^{\leq 1}(f)$.) The following gives the $A = DR_{1-\varepsilon}$ part of Theorem 4.2.

Theorem 5.4. *There is a universal constant C such that for every $0 < \varepsilon < 1/2$ and $d \geq 1$ we have*

$$\sup_{f:\{-1,1\}^n \to B_d} \langle f, DR_{1-\varepsilon} f\rangle = \langle \chi_i, DR_{1-\varepsilon}\chi_i\rangle = \varepsilon \quad \text{for all } i, \tag{5.9}$$

and

$$\sup_{\substack{f:\{-1,1\}^n \to [-1,1] \\ \forall i\, |\hat{f}(i)| \leq 1/C\log(1/\varepsilon)}} \langle f, DR_{1-\varepsilon} f\rangle \leq C\varepsilon/\log(1/\varepsilon). \tag{5.10}$$

Proof. The proof of (5.9) is trivial: By identity (5.4) (with the $\langle \Delta_1 f, f\rangle$ term dropping out) we see that $\langle f, DR_{1-\varepsilon} f\rangle$ is always at most ε. Since dictators achieve ε, they are indeed optimal. As for (5.10), the proof is similar to the Gaussian case, except now $P_1 f$ is a Rademacher average, $P_1 f = \sum_{i=1}^n \hat{f}(i) x_i$ with the x_i independent uniform ± 1. Nevertheless the heavy tail property still holds assuming all $|\hat{f}(i)|$ are small enough: the following result appears in, e. g., the book of Ledoux–Talagrand [35, p. 92]: *There is a universal constant C such that if $\sum_{i=1}^n \alpha_i^2 = \sigma^2$, $t \geq \sigma$, and $|\alpha_i| \leq \sigma^2/Ct$ for all i, then*

$$\mathbf{Pr}_{x\in\{-1,1\}^n}[\textstyle\sum_{i=1}^n \alpha_i x_i \geq t] \geq \exp(-Ct^2/\sigma^2).$$

The result (5.10) now follows by the same argument as in the Gaussian case. □

We now have a "Long Code test" for Quadratic-Programming using $DR_{1-\varepsilon}$ very much like the one for Max-Cut used in [28]. However $DR_{1-\varepsilon}$ is not a Max-Cut-Gain operator, which we would prefer. However, as we will see in the next section, it can be replaced by one.

5.4 The relation of $DR_{1-\varepsilon}$ to Max-Cut-Gain

In this section we show that $DR_{1-\varepsilon}$ can be replaced by a negative probability operator with essentially the same properties. This gives us an SDP gap and Long Code test for Max-Cut-Gain with essentially the same soundness and completeness as in Theorems 5.3 and 5.4, completing the proof of Theorems 4.1 and 4.2.

Theorem 5.5. *Let X denote either $L_2(\mathbb{R}^n,\gamma)$ or $L_2(\{-1,1\}^n)$. Then for each $0<\varepsilon<1/2$ there is a negative probability operator A_ε on X such that for all $d\geq 1$,*

$$\sup_{f:X\to B_d}\langle f, DR_{1-\varepsilon}f\rangle \;\leq\; \sup_{f:X\to B_d}\langle f, A_\varepsilon f\rangle \;\leq\; \frac{1}{4}\sup_{f:X\to B_d}\langle f, DR_{1-4\varepsilon}f\rangle. \tag{5.11}$$

Proof. We let $A_\varepsilon = -(pT_1 + (1-p)T_{-1/2})$, where T_ρ denotes the Bonami-Beckner operator in the discrete cube case and the Ornstein-Uhlenbeck operator in the Gaussian case; and, where p is chosen so that the projection function $g(y)=y_1$ satisfies $\langle g, A_\varepsilon g\rangle = \varepsilon$; specifically, $p = \frac{1}{3}-\frac{2}{3}\varepsilon$. Since the Bonami-Beckner and Ornstein-Uhlenbeck operators are probability operators, A_ε is clearly a negative probability operator.

Let us now calculate $\langle f, A_\varepsilon f\rangle$. We write f in terms of its Fourier or Hermite expansion, $f=\sum_{k\geq 0}P_k f$, and recall that $T_\rho P_k f = \rho^k P_k f$. Hence

$$\langle f, A_\varepsilon f\rangle \;=\; \sum_{d\geq 0} c_d \|P_d f\|_2^2,$$

where $c_d = -(p\cdot 1^d + (1-p)\cdot(-1/2)^d)$. It's not very difficult to check that $c_1=\varepsilon$ and that

$$-1 \leq c_d \leq c_3 = -(\tfrac{1}{4}-\tfrac{3}{4}\varepsilon)$$

for all $d\neq 1$. Hence

$$\varepsilon\|P_1 f\|_2^2 - \|f-P_1 f\|_2^2 \;\leq\; \langle f, A_\varepsilon f\rangle \;\leq\; \varepsilon\|P_1 f\|_2^2 - (\tfrac{1}{4}-\varepsilon)\|f-P_1 f\|_2^2.$$

Comparing this with (5.4) (which reads $\langle f, DR_{1-\varepsilon}f\rangle = \varepsilon\|P_1 f\|_2^2 - (1-\varepsilon)\|P_1 f - f\|_2^2$ in the Gaussian setting since $\langle \Delta_1 f, f\rangle = 0$) yields the theorem, (5.11). □

5.5 UGC-hardness of Max-Cut-Gain

Using the Long Code test just developed, we can get Theorem 4.3, an ε versus $O(\varepsilon/\log(1/\varepsilon))$ UGC-hardness result for Max-Cut-Gain which matches the algorithm of Charikar and Wirth [12]. The proof follows by repeating almost verbatim the reduction in [28] that proves $.878+\varepsilon$ UGC-hardness for Max-Cut.

More specifically, we first modify [28]'s reduction in Section 8.1. That reduction picks $x\in\{-1,1\}^M$ uniformly, forms $y = x\mu \in \{-1,1\}^M$ by picking μ from the ρ-biased distribution, and "accepts" if and only if $f_w(x\circ\sigma)\neq f_w(y\circ\sigma')$. Instead, given a constant $\varepsilon>0$, our reduction chooses the pair (x,y) with probability $-A_{xy}$, where $A=A_\varepsilon$ is the negative probability operator from Theorem 4.1. This makes

sense: since A is a negative probability operator, its entries are all negative and their absolute values sum to 1. Under this distribution, for $f : \{-1,1\}^M \to \{-1,1\}$ we have, as in (1.1),

$$\mathbf{Pr}[f(x) \neq f(y)] = \sum_{x,y} (-A_{xy}) \left(\tfrac{1}{2} - \tfrac{1}{2} f(x) f(y)\right) = \tfrac{1}{2} + \tfrac{1}{2} \langle f, Af \rangle ;$$

and hence the "gain" over $1/2$ is $\langle f, Af \rangle$.

The reduction's completeness is as in [28]: If f is a dictator (Long Code) then the first part of Theorem 4.2 implies that the Max-Cut-Gain instance will have value at least $\Omega(\varepsilon) - O(\eta)$. (The label size M can be increased to $O(\log(1/\varepsilon))$ if necessary.) This value is still $\Omega(\varepsilon)$ if $\eta > 0$ is taken to be a small enough constant compared to ε. As for soundness, the same analysis as in [28] gives that the value of the Max-Cut-Gain instance is $\langle g_v, Ag_v \rangle$, where g_v is as defined in [28]. We define a vertex v to be "good" if $\langle g_v, Ag_v \rangle$ is, say, at least twice the $O(\varepsilon/\log(1/\varepsilon))$ quantity in Theorem 4.2. Then at least an $\Omega(\varepsilon/\log(1/\varepsilon))$ fraction of v's are good, and for each, our Theorem 4.2 implies that g_v has at least one coordinate i with $|\hat{g_v}(i)| \geq \Omega(1/\log(1/\varepsilon))$. We can interpret this as $\mathrm{Inf}_i^{\leq 1}(g_v) \geq \Omega(1/\log^2(1/\varepsilon))$ and proceed with the proof from [28], taking its "k" to be 1. Overall, we get a Unique Label Cover labeling satisfying at least a $\gamma' = \Omega(\varepsilon/\log^5(1/\varepsilon))$ fraction of edges, a positive constant as necessary.

6 Discretizing the Gaussian SDP gap for $DR_{1-\varepsilon}$

6.1 Overview

As an interlude before we move on to $K_{N,N}$-Quadratic-Programming, we describe how to perform a discretization or "randomness-reduction" on our Gaussian SDP gap for $DR_{1-\varepsilon}$. The ideas in this section may be useful for shortening the Long Code and possibly eliminating the loss of a logarithm in certain SDP gap constructions, such as those in [32, 33].

By taking the constant $\varepsilon \to 0$, Theorem 5.3 shows that there is no constant integrality gap for the Quadratic-Programming problem. We would like to give a construction where ε is "subconstant"; i. e., a function of the number of points in the domain. This doesn't make sense in Gaussian space since this is an infinite domain. However we will show that we can discretize Gaussian space into N points in such a way that the SDP gap still holds. In particular we will show how to do this with dimension $d = \mathrm{poly}(1/\varepsilon)$ and number of points $N = \mathrm{poly}(d) = \mathrm{poly}(1/\varepsilon)$. Since the SDP gap is $\Omega(\log(1/\varepsilon))$, this yields a Quadratic-Programming integrality gap of $\Omega(\log N)$ on N-vertex graphs. As discussed in Section 4, this is tight up to constant factors.

6.2 How to discretize

The discretization we perform is as simple as possible: We simply pick a set $G \subset \mathbb{R}^d$ of N points randomly according to the d-dimensional Gaussian distribution. We view G as having the uniform distribution μ. We can define orthogonal coefficients, P_1, Δ_1, and $DR_{1-\varepsilon}$ on $L_2^{(d)}(G,\mu)$ as in Section 5.1. We then show that with high probability over the choice of G, the ε versus $O(\varepsilon/\log(1/\varepsilon))$ SDP gap from Theorem 5.3 still holds. The basic idea is that the proof of Theorem 5.3 uses only a small number of facts about Gaussian random variables; we will isolate the facts used and ensure that G has each of them, at least approximately.

Specifically, given $d \geq 1$, we will require following three facts about $G \subset \mathbb{R}^d$:

$$\forall\, x \in G, \sqrt{d} - O(\sqrt{\log d}) \leq \|x\|_2 \leq \sqrt{d} + O(\sqrt{\log d})\,, \tag{6.1}$$

$$\forall\, 1 \leq i, j \leq n, \quad \left| \mathop{\mathbf{E}}_{x \in G}[x_i x_j] - \delta_{ij} \right| \leq 1/d^3\,, \tag{6.2}$$

$$\forall\, v \in \mathbb{R}^d \text{ with } \|v\|_2 = 1, \quad \mathop{\mathbf{Pr}}_{x \in G}\left[\langle v, x\rangle > \Omega(\sqrt{\log d})\right] \geq \alpha, \text{ where } \alpha = d^{-1/4}\,. \tag{6.3}$$

(Here δ_{ij} denotes the Kronecker delta.) In words: Property (6.1) demands that all vectors in G have length about $\sqrt{d}$; Property (6.2) demands that products of pairs of coordinate have essentially the correct expectation, since $\int x_i x_j \, d\gamma(x) = \delta_{ij}$; and Property (6.3) demands that the random variable $\langle v, x\rangle$, which should act like a linear combination of Gaussians, has an appropriately heavy tail for all unit-length coefficient vectors v.

Standard arguments based on the probabilistic method and the union bound show that G has all three properties with high probability when $N = d^{O(1)}$:

Lemma 6.1. *Let d be at least a sufficiently large constant and let $N = \Theta(d^7)$. Then with probability at least $1 - 1/N$, G satisfies Properties (6.1), (6.2), and (6.3).*

Proof. For Property (6.1), if g is drawn from the d-dimensional Gaussian distribution then $\mathbf{E}[\|g\|_2^2] = d$ and there is sharp concentration around the mean; in particular, Part 1 of Lemma 6.2 below implies

$$\mathbf{Pr}\left[\left|\|g\|_2^2 - d\right| \leq O(\sqrt{d \log N})\right] = \mathbf{Pr}\left[\sqrt{d} - O(\sqrt{d \log d}) \leq \|g\|_2 \leq \sqrt{d} + O(\sqrt{d \log d})\right] \geq 1 - \frac{1}{3N^2}$$

if the constant in the $O(\cdot)$ is large enough and d is at least a large constant. So by a union bound we have that Property (6.1) holds for G except with probability $1/(3N)$.

The proof of Property (6.2) is essentially similar. Fix some particular $i \neq j$; then when G is chosen at random, the distribution of the quantity $\mathbf{E}_{x \in G}[x_i x_j]$ is that of the average of N i.i.d. random variables whose distribution is the product of two independent normals. By Part 2 of Lemma 6.2 below, the quantity is within $O(\sqrt{(\log N)/N}) \leq 1/d^3$ of the desired value $\delta_{ij} = 0$ except with probability at most $1/(3N^2)$. Similarly, when $j = i$ we have that the quantity is within $1/d^3$ of $\delta_{ij} = 1$ except with probability at most $1/(3N^2)$, using Part 1 of Lemma 6.2 below. Taking a union bound over all $d^2 < N$ pairs (i, j), we get that Property (6.2) holds for G except with probability at most $1/(3N)$.

The proof of Property (6.3) is only a little bit trickier. Let us first fix a particular vector $v \in \mathbb{R}^d$ with $\|v\|_2 = 1$ and analyze the probability (over the choice of G) that $Y_v \geq \alpha N$, where Y_v is defined by

$$Y_v = \#\{x \in G : \langle v, x\rangle > \Omega(\sqrt{\log d})\}\,.$$

We can write $Y_v = \sum_{i=1}^N X_i$, where X_i is an indicator random variable for the event that $\langle v, g_i\rangle \geq \Omega(\sqrt{\log d})$. Since $\langle v, g_i\rangle$ is distributed as a standard one-dimensional Gaussian, by the heavy tail property of Gaussians we get that $\mathbf{Pr}[X_i = 1] \geq 2\alpha$, assuming the constant in the $\Omega(\cdot)$ is small enough. Hence by standard large deviation bounds with respect to the sum $Y_v = \sum_{i=1}^n X_i$ we conclude that $\mathbf{Pr}[Y_v < \alpha N] \leq \exp(-\Omega(2\alpha N)) \leq \exp(-d^6)$.

Now let $\mathcal{N}$ be a $(1/d)$-net on the d-dimensional unit sphere; by a standard bound we may assume $|\mathcal{N}| \le 2^{O(d \log d)}$. Since $|\mathcal{N}| \exp(-d^6)$ is much less than $1/(3N)$, another union bound lets us conclude that when G is chosen at random, except with probability at most $1/(3N)$ we have

$$\forall\, v \in \mathcal{N}, \quad \mathop{\mathbf{Pr}}_{x \in G}\left[\langle v,x\rangle > \Omega(\sqrt{\log d})\right] \ge \alpha\,.$$

Assuming this happens, and also assuming Property (6.1) holds, Property (6.3) follows easily. To see this, let w be *any* unit vector in $\mathbb{R}^d$, and let v be the closest point in $\mathcal{N}$ to w. We know that at least an α fraction of the points x in G satisfy $\langle v,x\rangle > \Omega(\sqrt{\log d})$. But for these x's,

$$\langle w,x\rangle \ge \langle v,x\rangle - \|v-w\|_2 \cdot \|x\|_2 > \Omega(\sqrt{\log d}) - (1/d)(\sqrt{d} + O(\sqrt{\log d})) = \Omega(\sqrt{\log d})\,;$$

thus Property (6.3) holds for w. The proof is complete. □

Following are the large deviation bounds we used in the above proof:

Lemma 6.2. *There is a universal constant $C < \infty$ such that the following hold:*

1. *Let $X_1, \ldots, X_n$ be i.i.d. random variables, where X_i is distributed as the square of a standard Gaussian. Then for all $s < \sqrt{n}$,*

$$\mathbf{Pr}\left[\left|(X_1 + \cdots + X_n) - n\right| \ge s\sqrt{n}\right] \le 2\exp(-s^2/C)\,.$$

2. *If X_i is instead distributed as the product of two independent standard Gaussians, then for all $t < \sqrt{n}$,*

$$\mathbf{Pr}\left[\left|(X_1 + \cdots + X_n)/n\right| \ge s/\sqrt{n}\right] \le 2\exp(-s^2/C)\,.$$

Proof. These are completely standard applications of the theory of large deviations [15, 14] of Cramér and Chernoff; we include the proofs for convenience. For Part 1, direct calculation shows that the moment generating function of $X_1 - 1$ is $g(t) = \exp(-t)/\sqrt{1-2t}$, converging for $t < 1/2$. Writing $\varepsilon = s/\sqrt{n} < 1$ and selecting $t = \varepsilon/(2+2\varepsilon) < 1/2$, the theory tells us that

$$\mathbf{Pr}\left[(X_1 + \cdots + X_n - n)/n \ge s/\sqrt{n}\right] \le \left(\exp(-\varepsilon/2)\sqrt{1+\varepsilon}\right)^n.$$

We have the Taylor expansion $\exp(-\varepsilon/2)\sqrt{1+\varepsilon} = 1 - \varepsilon^2/4 + \varepsilon^3/6 - \cdots$ and so it's easy to check that for $\varepsilon < 1$ the quantity is upper-bounded by $\exp(-\varepsilon^2/C)$ for some $C < \infty$; indeed, $C = 7$ suffices. Thus

$$\mathbf{Pr}\left[X_1 + \cdots + X_n - n \ge s\sqrt{n}\right] \le \left(\exp(-\varepsilon^2/C)\right)^n = \exp(-s^2/C)\,.$$

The proof of $\mathbf{Pr}[X_1 + \cdots + X_n - n \le -s\sqrt{n}] \le \exp(-s^2/C)$ is the same, replacing t with $-t$.

As for Part 2, direct calculation shows that the moment generating function of X_1 is $1/\sqrt{1-t^2}$, converging for $|t| < 1$. Again we write $\varepsilon = s/\sqrt{n} < 1$, and we then select $t = \varepsilon$. Then Chernoff's theory tells us that

$$\mathbf{Pr}\left[(X_1 + \cdots + X_n)/n \ge s/\sqrt{n}\right] \le \left(\exp(-\varepsilon^2)/\sqrt{1-\varepsilon^2}\right)^n.$$

We have the Taylor expansion $\exp(-\varepsilon^2)/\sqrt{1-\varepsilon^2} = 1 - \varepsilon^2/2 + (3/8)\varepsilon^4 + \cdots$, and the remainder of the proof is very similar to that for Part 1. □

6.3 Reproving the SDP gap on G

We can now prove Theorem 4.4:

Proof. Applying Lemma 6.1 we have that G satisfies Properties (6.1), (6.2), and (6.3). We now proceed with the proof of Theorem 5.3, making the corrections necessary given that we are on the domain G rather than $(\mathbb{R}^d, \gamma)$.

For the "completeness" lower bound we use the same function $f(z) = z/\|z\|_2$. Now

$$\hat{f}(i)_i = \mathop{\mathbf{E}}_{y \in G}\Big[\frac{y_i^2}{\|y\|_2}\Big] \geq \frac{1 - 1/d^3}{\sqrt{d} + O(\sqrt{\log d})} \geq \frac{1}{\sqrt{d}} - O\Big(\frac{\sqrt{\log d}}{d}\Big)$$

using Properties (6.1) and (6.2); hence

$$\sum_{i=1}^{d} \|\hat{f}(i)\|_2^2 \geq 1 - O(\sqrt{(\log d)/d}) \geq 1 - \varepsilon/2\,,$$

and therefore $\langle f, DR_{1-\varepsilon} f\rangle \geq \varepsilon/2$, using (5.3) again.

For the "soundness" lower bound, given $f : G \to [-1,1]$ we can no longer use $\langle \Delta_1 f, f\rangle = 0$. However using Properties (6.1) and (6.2) it is not hard to show that $|\langle \Delta_1 f, f\rangle|$ is very small. Combining (5.5) and Property (6.2) we have

$$|\langle \Delta_1 f, f\rangle| \leq (1/d^3) \sum_{i,j=1}^{d} |\hat{f}(i)\hat{f}(j)| \leq (1/d^2) \sum_{i=1}^{d} \hat{f}(i)^2\,.$$

We can crudely upper-bound $\sum_{i=1}^{d} \hat{f}(i)^2$ using the following formal identity:

$$\sum_{i=1}^{d} \hat{f}(i)^2 = \mathop{\mathbf{E}}_{x,y \in G}[\langle x, y\rangle f(x) f(y)]\,.$$

Since f's range is $[-1,1]$ and $|\langle x, y\rangle| \leq O(d)$ for every $x, y \in G$ (using Property (6.1)), we get $\sum_{i=1}^{d} \hat{f}(i)^2 \leq O(d)$. Overall we conclude that

$$|\langle \Delta_1 f, f\rangle| \leq O(1/d) \leq \varepsilon^2\,. \tag{6.4}$$

We now continue as in the proof of Theorem 5.3. Let $w \in \mathbb{R}^d$ be the vector whose coefficients are the $\hat{f}(i)$'s and let $\sigma^2 = \|w\|_2^2 = \sum \hat{f}(i)^2$. Now that we know $|\langle \Delta_1 f, f\rangle| \leq O(1/d) \leq \varepsilon^2$, the identity (5.2) lets us conclude the much sharper upper bound $\sigma^2 \leq 1 + \varepsilon^2$. We will again use (5.4) (combined with (6.4)):

$$\langle f, DR_{1-\varepsilon} f\rangle \leq \varepsilon\sigma^2 - (1-\varepsilon)\|P_1 f - f\|_2^2 + \varepsilon^2\,. \tag{6.5}$$

Now $(P_1 f)(x) = \langle w, x\rangle$ is no longer a Gaussian; however, Property (6.3) still lets us conclude that

$$\mathop{\mathbf{Pr}}_{x \in G}\Big[(P_1 f)(x) > \Omega(\sqrt{\log d}/\sigma)\Big] \geq \alpha = \varepsilon^{3/4}\,.$$

So as in the old proof, if $\sigma^2 \geq C'/\log(1/\varepsilon)$ for C' a sufficiently large constant, then $\Omega(\sqrt{\log d}/\sigma > 2$. In this case, $(P_1 f)(x)$ exceeds 2 with probability at least $\varepsilon^{3/4}$ hence $\|P_1 f - f\|_2^2 \geq \varepsilon^{3/4}$ and so (6.5) is negative (using $\sigma^2 \leq 1 + \varepsilon^2$). Thus we may assume $\sigma^2 \leq C'/\log(1/\varepsilon)$ and we get $\langle f, DR_{1-\varepsilon} f\rangle \leq O(\varepsilon/\log(1/\varepsilon))$ from (6.5), as required. □

7 $K_{N,N}$-Quadratic-Programming

7.1 SDP gap for DR_λ on Gaussian space: The Davie–Reeds lower bound on the Grothendieck constant

In this section we review the best known lower bound on the Grothendieck constant. This bound (i. e., SDP gap) was first proven by Davie [17], although his paper is unpublished and difficult to obtain. We will instead follow the independent proof of Reeds [43], which uses the same operator. Reeds's proof is less slick but it is the one that we will be able to translate into an equivalent Long Code test in the $\{-1,1\}^n$ setting. The main translational difficulty is that both papers begin by simplifying the analysis using the rotational symmetry of the Gaussian distribution. This is of course missing in the $\{-1,1\}^n$ case. We show in this section how to change Reeds's proof so that it does not use this step. For more on SDP gaps and Long Codes tests for the $K_{N,N}$-Quadratic-Programming problem, the reader may consult the very recent paper of Raghavendra and Steurer [42].

Let us begin with some notation from Reeds's paper. For $0 < \lambda < .4$, define $s(\lambda)$ to be $(\lambda/\eta)^2 + \lambda(1-4\Phi(-\eta))$, where $\eta \in (0,1)$ satisfies $\sqrt{2/\pi}\eta \exp(-\eta^2/2) = \lambda$ and Φ is the cdf of a standard normal.[1] The Davie–Reeds result shows an SDP gap of $1-\lambda$ versus $s(\lambda)$; in particular, this gives a lower bound of

$$K_{\text{Davie–Reeds}} = \sup_{0<\lambda<.4} (1-\lambda)/s(\lambda) \approx 1.6769$$

for Grothendieck's constant, taking $\lambda = \lambda^* \approx .19748$.

We now sketch (a slight alteration of) Reeds's proof:

Theorem 7.1. *There is a universal constant C such that for all $0 < \lambda < .4$ and $d \geq 1$,*

$$\sup_{f,g:(\mathbb{R}^d,\gamma)\to B_d} \langle f, DR_\lambda g\rangle \geq 1-\lambda-C/d \tag{7.1}$$

and

$$\sup_{f,g:(\mathbb{R}^d,\gamma)\to[-1,1]} \langle f, DR_\lambda g\rangle = \sup_{g:(\mathbb{R}^d,\gamma)\to[-1,1]} \int |DR_\lambda g| \leq s(\lambda). \tag{7.2}$$

Proof. (Sketch of Reeds's proof.) The bound (7.1) follows immediately from (5.6). To prove (7.2), first note that it suffices to take the supremum over functions $g : (\mathbb{R}^d,\gamma) \to \{-1,1\}$, since $\int |DR_\lambda g|$ is a convex function of g. Hence we must determine

$$\sup_{g:(\mathbb{R}^d,\gamma)\to\{-1,1\}} \int |P_1 g - \lambda g|. \tag{7.3}$$

So suppose $g : (\mathbb{R}^d,\gamma) \to \{-1,1\}$. Let ℓ denote $P_1 g$ and $\sigma = \|\ell\|_2 \leq 1$. If $\sigma = 0$ then clearly (7.3) equals λ, which is smaller than $s(\lambda)$ for $\lambda < .4$. Hence we may assume $\sigma > 0$; in this case, let ϕ denote the pdf of ℓ/σ (so in fact here ϕ is the standard Gaussian pdf). Let $\theta_g : \mathbb{R} \to [-1,1]$ be defined by

[1] Reeds's proof considers all $0 \leq \lambda \leq \sqrt{2/e\pi} \approx .484$, but the proof is slightly simpler if one restricts λ's range slightly so that we always have $\lambda < s(\lambda)$. This doesn't hurt because the optimal λ turns out to be .19748. (Note that Reeds's paper states the optimal λ is .25573 but this is a typo—that's the optimal value of η.)

$\theta_g(z) = \mathbf{E}[g \mid \ell/\sigma = z]$, so $\mathbf{Pr}[g = 1 \mid \ell/\sigma = z] = 1/2 + \theta_g(z)/2$ and $\mathbf{Pr}[g = -1 \mid \ell/\sigma = z] = 1/2 - \theta_g(z)/2$. Note that, by definition,

$$\int_{\mathbb{R}} \phi(z) z \theta_g(z)\, dz = \sigma \tag{7.4}$$

and

$$\int |P_1 g - \lambda g| = \int_{\mathbb{R}} \phi(z) \left(\left(\tfrac{1}{2} + \tfrac{1}{2}\theta_g(z) \right) \left| \sigma z - \lambda \right| + \left(\tfrac{1}{2} - \tfrac{1}{2}\theta_g(z) \right) \left| \sigma z + \lambda \right| \right) dz. \tag{7.5}$$

Now (7.3) can be bounded from above as follows: First fix the value σ. Then the supremum in (7.3), conditioned on g satisfying $\|P_1 g\|_2^2 = \sigma^2$, is upper-bounded by what can thought of as an (infinite) linear program: the unknowns are the values of θ_g, the objective is (7.5) (which is linear in θ_g), and (7.4) and $|\theta_g| \le 1$ serve as the constraints. For each value of σ an optimal θ^* is determined; substituting this into (7.5) gives an upper bound for (7.3) conditioned on $\|P_1 g\|_2^2 = \sigma^2$. Finally, this upper bound is maximized over $0 < \sigma \le 1$.

Let us rewrite the objective (7.5) as

$$\int_{\mathbb{R}} \phi(z) \psi(z) \theta(z)\, dz + C_{\lambda,\sigma} \tag{7.6}$$

where

$$\psi(z) = \tfrac{1}{2}\left(|\sigma z - \lambda| - |\sigma z + \lambda| \right), \qquad C_{\lambda,\sigma} = \int_{\mathbb{R}} \phi(z) \cdot \tfrac{1}{2}\left(|\sigma z - \lambda| + |\sigma z + \lambda| \right) dz.$$

Since ϕ is an even function (and ψ is odd), we see that both the constraint (7.4) and the objective (7.6) involve integrating θ against an odd function. It follows that the optimal θ^* may as well be odd as well, since one can replace an arbitrary optimizer θ^* by $\theta'(z) = (\theta^*(z) - \theta^*(-z))/2$. Thus we equivalently are trying to find $\theta : \mathbb{R}^+ \to [-1,1]$ maximizing

$$\int_0^\infty 2\phi(z)\psi(z)\theta(z)\, dz + C_{\lambda,\sigma} \qquad \text{subject to} \qquad \int_0^\infty 2\phi(z) z \theta(z)\, dz = \sigma.$$

Since $\psi(z)/z$ is increasing in z for all σ, λ, it follows that the optimizing θ^* is of the form

$$\theta^*(z) = \begin{cases} 1 & \text{if } z \in (-h, 0) \cup (h, \infty) \\ -1 & \text{if } z \in (-\infty, -h) \cup (0, h), \end{cases} \tag{7.7}$$

where $h = h(\sigma) \ge 0$ is selected to be as large as possible such that the constraint (7.4) holds.

Thus we conclude that

$$\sup_{f,g:(\mathbb{R}^d,\gamma)\to[-1,1]} \langle f, DR_\lambda g \rangle \le \sup_{0<\sigma\le 1} \int_{\mathbb{R}} \phi(z) \Psi_{\lambda,h}(z)\, dz, \tag{7.8}$$

where $\Psi_{\lambda,h} : \mathbb{R} \to \mathbb{R}$ is defined by

$$\Psi_{\lambda,h}(z) = \tfrac{1}{2}(|\sigma z - \lambda| - |\sigma z + \lambda|)\theta^*(z) + \tfrac{1}{2}(|\sigma z - \lambda| + |\sigma z + \lambda|),$$

h is defined in terms of σ via

$$\int_{\mathbb{R}} \phi(z) z \theta^*(z)\, dz = \sigma, \tag{7.9}$$

and θ^* is defined in (7.7).

With $\phi(z) = \exp(-z^2/2)/\sqrt{2\pi}$, everything in the expression on the right in (7.8) is completely explicit, and Reeds calculates that it is equal to (7.2). □

7.2 Long Code test with DR_λ on the discrete cube

We now transfer the SDP gap from the previous section to the discrete cube setting, just as we did in the Quadratic-Programming case. This time the SDP gap used more than just the "heavy tail" property of the linear part of a Gaussian function; it uses estimates on certain piecewise linear functionals of the linear part. The following is a restatement of Theorem 4.5.

Theorem 7.2. *There is a universal constant C such that for all $0 < \lambda < .4$, $d \geq 1$, and $\delta > 0$,*

$$\sup_{f,g:\{-1,1\}^n \to B_d} \langle f, DR_\lambda g\rangle = \langle \chi_i, DR_\lambda \chi_i\rangle = 1-\lambda \quad \text{for all } i, \tag{7.10}$$

$$\text{and} \sup_{\substack{f,g:\{-1,1\}^n \to \{-1,1\} \\ \forall i\, |\hat{g}(i)| \leq \delta}} \langle f, DR_\lambda g\rangle = \sup_{\substack{g:\{-1,1\}^n \to \{-1,1\} \\ \forall i\, |\hat{g}(i)| \leq \delta}} \mathbf{E}[|DR_\lambda g|] \leq s(\lambda) + C\delta\,. \tag{7.11}$$

Proof. We begin by noting that for $f, g : \{-1,1\}^n \to B_d$,

$$\langle f, DR_\lambda g\rangle \leq \|DR_\lambda g\|_2 \leq \sqrt{(1-\lambda)^2 \|P_1 g\|_2^2 + \lambda^2 \textstyle\sum_{i\neq 1} \|P_i g\|_2^2}; \tag{7.12}$$

thus certainly $\langle f, DR_\lambda g\rangle \leq 1-\lambda$ and so (7.10) is proved.

To prove (7.11) we proceed as in the proof of (7.2) from Reeds's proof; we bound $\mathbf{E}[|DR_\lambda g|]$ assuming $g : \{-1,1\}^n \to \{-1,1\}$ is some function satisfying $|\hat{g}(i)| \leq \delta$ for all i. Again we let $\ell = P_1 g$, a Rademacher average, and define $\sigma = \|\ell\|_2$. We may assume

$$\sigma^2 \geq .001 \tag{7.13}$$

because otherwise $\mathbf{E}[|DR_\lambda g|] \leq \|DR_\lambda g\|_2 \leq \sqrt{.001 + \lambda^2}$ using (7.12), and it can be checked numerically that $\sqrt{.001+\lambda^2} \leq s(\lambda)$. (In fact, we only need this for the optimal λ^*.)

We now define ϕ' to be the pmf (probability mass function) of ℓ/σ and proceed with the linear programming part of Reeds's proof. We continue to use the notation

$$\int_{\mathbb{R}} \phi'(z) F(z)\, dz$$

for the expected value of the discrete random variable $F(z)$ when z is distributed as ℓ/σ. Note that are now only a finite number of "variables" $\theta(z)$, corresponding to the values that ℓ/σ can attain. The pmf ϕ' is still an even function, so the optimizing θ^* can again be odd. Again, the fact that $\psi(z)/z$ is increasing implies that the optimizing θ^* is of the form (7.7) for some h that ℓ/σ attains; the value of θ^* precisely at $\pm h$ may be in the interior of $[-1,1]$ so as to satisfy the constraint (7.4). Thus we have

$$\sup_{\substack{g:\{-1,1\}^n \to \{-1,1\} \\ \forall i\, |\hat{g}(i)| \leq \delta}} \mathbf{E}[|DR_\lambda g|] \leq \sup_{.001 \leq \sigma \leq 1} \int_{\mathbb{R}} \phi'(z) \Psi_{\lambda,h}(z)\, dz$$

as in Reeds's proof.

Our goal is to show this is the at most the same quantity with ϕ in place of ϕ' (as in (7.8)), up to an additive $O(\delta)$. By the piecewise-linearity of θ^* and $\Psi_{\lambda,h}$, it is easy to check that this would follow immediate from:

Claim 7.3. *If $F(z)$ is any piecewise-linear function on $\mathbb{R}$ with $O(1)$ pieces, each of the form $az+b$ with $|a|,|b| \le O(1)$, then*

$$\left| \int \phi'(z)F(z)\,dz - \int \phi(z)F(z)\,dz \right| \le O(\delta).$$

We complete the overall proof by proving the claim. The key is to use the fact that (ℓ/σ) is a Rademacher average with variance 1 and all coefficients at most $\delta/\sigma \le 1000\delta$ in magnitude (using (7.13)). Thus the Central Limit Theorem implies that ϕ' is "close" to the normal pdf ϕ. More precisely, for $I \subseteq \mathbb{R}$ an interval, define $\Phi'(I) = \mathbf{Pr}[\ell/\sigma \in I]$ and define $\Phi(I) = \mathbf{Pr}[N(0,1) \in I]$. Then the non-uniform Berry-Esseen Theorem due to Bikelis [9] implies that

$$\left|\Phi'(-\infty,t] - \Phi(\infty,t]\right| \le \frac{O(\delta)}{1+|t|^3} \qquad \forall\, t \in \mathbb{R}. \tag{7.14}$$

In particular, $|\Phi'(I) - \Phi(I)| \le O(\delta)$ for all intervals $I \subseteq \mathbb{R}$, and $\mathbf{Pr}[\ell/\sigma = t] \le O(\delta)$ for each $t \in \mathbb{R}$.

To prove the Claim it suffices to prove the following: for any linear function $F(z) = az+b$ on any interval $I \not\ni 0$,

$$\left| \int_I \phi'(z)F(z)\,dz - \int_I \phi(z)F(z)\,dz \right| \le O(\delta)(|a|+|b|)$$

(here the notation $\int_I \phi'(z)F(z)\,dz$ should be interpreted as $\int \phi'(z)\mathbf{1}_{z\in I}F(z)\,dz$). We will show this for $I = [u,v)$ with $u \ge 0$, leaving the other cases to the reader. We have

$$\int_{[u,v)} \phi'(z)F(z)\,dz = b\Phi'[u,v) + a\int_{[u,v)} \phi'(z)z\,dz = b\Phi'[u,v) + a\left(u\Phi'[u,v) + \int_u^v \Phi'[t,v)\,dt\right).$$

Subtracting this from the same expression with Φ gives an error term of

$$b\left(\Phi'[u,v) - \Phi[u,v)\right) + a\left(u\left(\Phi'[u,v) - \Phi[u,v)\right) + \int_u^v \left(\Phi'[t,v) - \Phi[t,v)\right)dt\right),$$

the magnitude of which, by (7.14), is indeed at most

$$|b|O(\delta) + |a|\left(u\cdot O(\delta)/(1+u^3) + O(\delta)\int_u^v 1/(1+t^3)\,dt\right) \le O(\delta)(|a|+|b|),$$

as claimed. □

Using this theorem we get a sort of "two-function $K_{N,N}$-Quadratic-Programming Long Code test": we see that if $f,g : \{-1,1\}^n \to \{-1,1\}$ have $\langle f, DR_\lambda g\rangle > s(\lambda) + C\delta$ then g must have a coordinate i with $\mathrm{Inf}^{\le 1}(f) \ge \delta^2$. Indeed, since DR_λ is self-adjoint, *both* f and g must have at least one influential coordinate. However this is not a proper two-function Long Code test, because to get something useful for UGC-hardness we need to show that f and g *share* an influential coordinate when $\langle f, DR_\lambda g\rangle$ is too large.

It's not clear whether or not this is true. However in the next section we will show that such a statement can be made if we slightly "smooth" the operator DR_λ.

7.3 A smoothed version of DR_λ

To get a useful two-function $K_{N,N}$-Quadratic-Programming Long Code test we introduce the following operator, for small $\varepsilon > 0$:

$$DR_\lambda^{(1-\varepsilon)} = DR_\lambda T_{1-\varepsilon} = T_{1-\varepsilon} DR_\lambda ;$$

here $T_{1-\varepsilon}$ is the Bonami-Beckner operator with parameter $1-\varepsilon$. Introducing the slight smoothing $T_{1-\varepsilon}$ does not hurt completeness very much, and on the soundness side it helps by making general influences and low-degree influences essentially equivalent. With $T_{1-\varepsilon}$ built into our operator it might seem that a straightforward averaging trick—like the one used by Dinur, Mossel, and Regev [18] in the context of noise stability tests—should suffice to give us a two-function Long Code test. However there is a technical difficulty: we only know how to prove soundness (7.11) for functions $g : \{-1,1\}^n \to \{-1,1\}$ with small $|\hat{g}(i)|$'s, not for the functions $g : \{-1,1\}^n \to [-1,1]$ that arise from averaging. Bypassing this difficulty requires some technical machinations.

The main trick involves a twist on the so-called Szemerédi Regularity Lemma for $(\mathbb{Z}/2\mathbb{Z})^n$ due to Green [21]. Given $f : \{-1,1\}^n \to \{-1,1\}$, $0 < \varepsilon < 1/2$, and $J \subseteq [n]$ we say that f *is ε-flat for J* if for all but an ε fraction of the strings $x_J \in \{-1,1\}^J$, it holds that $|\widehat{f_{x_J}}(i)| \le \varepsilon$ for all $i \in [n] \setminus J$. Here f_{x_J} denotes the restricted function on $\{-1,1\}^{[n]\setminus J}$ obtained by fixing the inputs x_J.

Theorem 7.4. *There is a universal function $W : (0,1/2) \to \mathbb{N}$ such that the following holds: For any $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon < 1/2$ one can associate a canonical set of coordinates $J = J_{f,\varepsilon} \subseteq [n]$ satisfying $|J| \le W(\varepsilon)$ such that f is ε-flat for J.*

To prove this theorem one only needs to repeat Green's proof with the notion of ε-flatness replacing his notion of ε-regularity.

We can now state our two-function Long Code test based on $K_{N,N}$-Quadratic-Programming.

Theorem 7.5. *There is a universal constant C' such that the following holds: Given $0 < \varepsilon < 1/2$, set $k = \lceil \log(1/\varepsilon)/\varepsilon \rceil$ and $\delta = \varepsilon^2/W(\varepsilon)$, where W is the function from Theorem 7.4. Then*

$$\sup_{f,g:\{-1,1\}^n \to \{-1,1\}} \langle f, DR_\lambda^{(1-\varepsilon)} g \rangle = \langle \chi_i, DR_\lambda^{(1-\varepsilon)} \chi_i \rangle = (1-\lambda)(1-\varepsilon) \quad \text{for all } i, \tag{7.15}$$

$$\text{and} \qquad \sup'_{f,g:\{-1,1\}^n \to \{-1,1\}} \langle f, DR_\lambda^{(1-\varepsilon)} g \rangle \le s(\lambda) + C'\varepsilon , \tag{7.16}$$

where the sup′ *in (7.16) is taken over all f and g such that $\mathrm{Inf}_i^{\le k}(g) \le \delta$ for all $i \in J = J_{f,\varepsilon}$, the canonical set of coordinates given by Theorem 7.4.*

Proof. The completeness statement (7.15) follows easily from (7.12), so we now focus on the soundness statement. Specifically, assume $f, g : \{-1,1\}^n \to \{-1,1\}$, $J = J_{f,\varepsilon}$ is the set of coordinates guaranteed by Theorem 7.4, and $\mathrm{Inf}_i^{\le k}(g) \le \delta$ for all $i \in J$.

Write $Q = DR_\lambda^{(1-\varepsilon)}$ for short and note that the notation $Qg^{\le k}$ makes sense, in that Q "commutes" with taking the low degree part. We have

$$\langle f, Qg \rangle = \langle f, Qg^{\le k} \rangle + \langle f, Qg^{>k} \rangle \le \langle f, Qg^{\le k} \rangle + \|Qg^{>k}\|_2 .$$

We have $\|Qg^{>k}\|_2^2 = \|DR_\lambda T_{1-\varepsilon} g^{>k}\|_2^2$; also $\|T_{1-\varepsilon} g^{>k}\|_2^2 \le (1-\varepsilon)^{2k} \le \varepsilon^2$ and DR_λ is a contraction on L_2. Thus it suffice to show

$$\langle f, Qg^{\le k}\rangle \le s(\lambda) + C'\varepsilon\,.$$

Given a function $h : \{-1,1\}^n \to \{-1,1\}$, let us use a bar to denote averaging over the coordinates of J. In other words, we write $\bar{h} : \{-1,1\}^n \to [-1,1]$ for the function $\mathbf{E}_{x_J}[h_{x_J}] = \sum_{S\cap J=\emptyset} \hat{h}(S)\chi_S$, and this is a function which doesn't depend on the coordinates of J. Note that this averaging also commutes with taking the low degree part. We have

$$\|g^{\le k} - \bar{g}^{\le k}\|_2^2 = \sum_{|S|\le k, S\cap J\neq 0} \hat{g}(S)^2 \le \sum_{i\in J} \mathrm{Inf}_i^{\le k}(g) \le |J|\delta = \varepsilon^2\,.$$

Using the fact that $DR_\lambda^{(1-\gamma)}$ is a contraction on L_2, an argument similar to the last one implies that it suffices for us to show

$$\langle f, Q\bar{g}^{\le k}\rangle \le s(\lambda) + C'\varepsilon\,.$$

Now since $Q\bar{g}^{\le k}$ does not depend on the coordinates of J we may write

$$\langle f, Q\bar{g}^{\le k}\rangle = \langle \bar{f}, Q\bar{g}^{\le k}\rangle\,,$$

where here $\langle , \rangle$ is the inner product on $\{-1,1\}^{[n]\setminus J}$. We will henceforth work over this probability space. Using the fact that DR_λ is self-adjoint and then using linearity, we have

$$\langle \bar{f}, Q\bar{g}^{\le k}\rangle = \mathop{\mathbf{E}}_{x_J}\langle DR_\lambda f_{x_J}, T_{1-\varepsilon}\bar{g}^{\le k}\rangle\,.$$

Now again, $\|T_{1-\varepsilon}\bar{g}^{>k}\|_2 \le (1-\varepsilon)^k \le \varepsilon$ so it suffices for us to prove

$$\mathop{\mathbf{E}}_{x_J}\langle T_{1-\varepsilon}\bar{g}, DR_\lambda f_{x_J}\rangle \le s(\lambda) + C'\varepsilon\,.$$

Since $T_{1-\varepsilon}\bar{g}$ has range $[-1,1]$, it suffices to show

$$\mathop{\mathbf{E}}_{x_J}\mathbf{E}[|DR_\lambda f_{x_J}|] \le s(\lambda) + C'\varepsilon\,.$$

We now use the fact that f is ε-flat for J. There are at most ε exceptional strings x_J for which we have no control over f_{x_J}. For these x_J we use simply $\mathbf{E}[|DR_\lambda f_{x_J}|] \le 1$. For the remainder of the strings x_J we have that f_{x_J} is a $\{-1,1\}$-valued function with $|\widehat{f_{x_J}}(i)| \le \varepsilon$ for all i. Thus $\mathbf{E}[|f_{x_J}|] \le s(\lambda) + C\varepsilon$ for these strings, by (7.11). This completes the proof. □

7.4 UGC-hardness of $K_{N,N}$-Quadratic-Programming

The two-function Long Code test from Theorem 7.5 yields Theorem 4.6, a $1/K_{\text{Davie–Reeds}} + \varepsilon \approx .597$ UGC-hardness result for $K_{N,N}$-Quadratic-Programming. This complements the approximation algorithm of Alon and Naor [3], which gives an approximation algorithm for $K_{N,N}$-Quadratic-Programming with factor $1/K_{\text{Krivine}} \approx .561$, where K_{Krivine} denotes the best known upper bound on Grothendieck's constant, due to Krivine [34].

The proof of Theorem 4.6 is again very easily derived by substituting the Long Code test from Theorem 7.5 into the UGC-hardness result for Max-Cut given in [28]. Indeed, the proof is even simpler because one directly replaces each constraint in the bipartite Unique Label Cover instance by the $DR_{\lambda^*}^{(1-\Omega(\varepsilon))}$ Long Code test, instead of putting tests between pairs of vertices on the right side and averaging over Long Codes on the left side.

References

[1] AMIT AGARWAL, MOSES CHARIKAR, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: $O(\sqrt{\log n})$ approximation algorithms for Min-Uncut, Min-2CNF-Deletion, and directed cut problems. In *Proc. 37th STOC*, pp. 573–581. ACM Press, 2005. [doi:10.1145/1060590.1060675]. 86

[2] NOGA ALON, KONSTANTIN MAKARYCHEV, YURY MAKARYCHEV, AND ASSAF NAOR: Quadratic forms on graphs. *Invent. Math.*, 163(3), 2006. [doi:10.1007/s00222-005-0465-9]. 86, 88, 89, 93, 97

[3] NOGA ALON AND ASSAF NAOR: Approximating the Cut-Norm via Grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803, 2006. [doi:10.1137/S0097539704441629]. 86, 88, 90, 98, 112

[4] NOGA ALON, BENNY SUDAKOV, AND URI ZWICK: Constructing worst case instances for semidefinite programming based approximation algorithms. *SIAM J. Discrete Math.*, 15(1):58–72, 2002. [doi:10.1137/S0895480100379713]. 85

[5] SANJEEV ARORA, ELI BERGER, ELAD HAZAN, GUY KINDLER, AND MULI SAFRA: On non-approximability for quadratic programs. In *Proc. 46th FOCS*, pp. 206–215. IEEE Comp. Soc. Press, 2005. [doi:10.1109/SFCS.2005.57]. 87, 88, 89, 93, 97

[6] SANJEEV ARORA, SATISH RAO, AND UMESH VAZIRANI: Expander flows, geometric embeddings and graph partitioning. In *Proc. 36th STOC*, pp. 222–231. ACM Press, 2004. [doi:10.1145/1007352.1007355]. 86

[7] NIKHIL BANSAL, AVRIM BLUM, AND SHUCHI CHAWLA: Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. [doi:10.1023/B:MACH.0000033116.57574.95]. 87

[8] MIHIR BELLARE, ODED GOLDREICH, AND MADHU SUDAN: Free bits, PCPs, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998. [doi:10.1137/S0097539796302531]. 95

[9] ALGIMANTAS BIKELIS: Estimates of the remainder in the Central Limit Theorem. *Litovsk. Mat. Sb.*, 6(3):323–346, 1966. In Russian. 110

[10] CHRISTER BORELL: Geometric bounds on the Ornstein-Uhlenbeck velocity process. *Z. Wahrsch. Verw. Gebiete*, 70(1):1–13, 1985. [doi:10.1007/BF00532234]. 89

[11] MOSES CHARIKAR, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: Near-optimal algorithms for unique games. In *Proc. 38th STOC*, pp. 205–214. ACM Press, 2006. [doi:10.1145/1132516.1132547]. 86

[12] MOSES CHARIKAR AND ANTHONY WIRTH: Maximizing quadratic programs: Extending Grothendieck's inequality. In *Proc. 45th FOCS*, pp. 54–60. IEEE Comp. Soc. Press, 2004. [doi:10.1109/FOCS.2004.39]. 86, 87, 89, 93, 96, 97, 101, 102

[13] SHUCHI CHAWLA, ROBERT KRAUTHGAMER, RAVI KUMAR, YUVAL RABANI, AND D. SIVAKUMAR: On the hardness of approximating Multicut and Sparsest-Cut. *Comput. Complexity*, 15(2):94–114, 2006. [doi:10.1007/s00037-006-0210-9]. 87

[14] HERMAN CHERNOFF: A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23(4), 1952. [doi:10.1214/aoms/1177729330]. 105

[15] HARALD CRAMÉR: Sur un nouveau théroème-limite de la théorie des probabilités. *Actualités Scientifiques et Industrielles*, 736, 1938. 105

[16] PIERLUIGI CRESCENZI, RICCARDO SILVESTRI, AND LUCA TREVISAN: On weighted vs. unweighted versions of combinatorial optimization problems. *Inform. and Comput.*, 167(1):10–26, 2001. [doi:10.1006/inco.2000.3011]. 90

[17] ALEXANDER DAVIE: Lower bound for K_G. Unpublished, 1984. 88, 96, 107

[18] IRIT DINUR, ELCHANAN MOSSEL, AND ODED REGEV: Conditional hardness for approximate coloring. In *Proc. 38th STOC*, pp. 344–353. ACM Press, 2006. [doi:10.1145/1132516.1132567]. 87, 111

[19] URIEL FEIGE AND MICHAEL LANGBERG: The RPR^2 rounding technique for semidefinite programs. *J. Algorithms*, 60(1):1–23, 2006. [doi:10.1016/j.jalgor.2004.11.003]. 85, 86, 97, 101

[20] MICHEL GOEMANS AND DAVID WILLIAMSON: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995. [doi:10.1145/227683.227684]. 84, 91, 97

[21] BEN GREEN: A Szemerédi-type regularity lemma in abelian groups, with applications. *Geom. Funct. Anal.*, 15(2):340–376, 2005. [doi:10.1007/s00039-005-0509-8]. 98, 111

[22] ALEXANDRE GROTHENDIECK: Résumé de la théorie métrique des produits tensoriels topologiques. *Bol. Soc. Mat. Sao Paulo*, 8:1–79, 1953. 88

[23] UFFE HAAGERUP: A new upper bound for the complex Grothendieck constant. *Israel J. Math.*, 60(2):199–224, 1987. [doi:10.1007/BF02790792]. 88

[24] JOHAN HÅSTAD: Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. [doi:10.1145/502090.502098]. 85, 95

[25] JOHAN HÅSTAD AND SRINIVASAN VENKATESH: On the advantage over a random assignment. *Random Structures Algorithms*, 25(2):117–149, 2004. [doi:10.1002/rsa.20031]. 86

[26] GEORGE KARAKOSTAS: A better approximation ratio for the Vertex Cover problem. In *Proc. 32nd Internat. Conf. on Automata, Languages, and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pp. 1043–1050. Springer, 2005. [doi:10.1007/11523468_84]. 86

[27] SUBHASH KHOT: On the power of unique 2-prover 1-round games. In *Proc. 34th STOC*, pp. 767–775. ACM Press, 2002. [doi:10.1145/509907.510017]. 85, 87, 95

[28] SUBHASH KHOT, GUY KINDLER, ELCHANAN MOSSEL, AND RYAN O'DONNELL: Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. [doi:10.1137/S0097539705447372]. 85, 86, 87, 95, 97, 101, 102, 103, 113

[29] SUBHASH KHOT AND ASSAF NAOR: Nonembeddability theorems via Fourier analysis. *Math. Ann.*, 334(4):821–852, 2006. [doi:10.1007/s00208-005-0745-0]. 86

[30] SUBHASH KHOT AND RYAN O'DONNELL: SDP gaps and UGC-hardness for MaxCutGain. In *Proc. 47th FOCS*, pp. 217–226. IEEE Comp. Soc. Press, 2006. [doi:10.1109/FOCS.2006.67]. 83, 88

[31] SUBHASH KHOT AND ODED REGEV: Vertex Cover might be hard to approximate to within $2-\varepsilon$. In *Proc. 18th Conf. on Computational Complexity (CCC)*, pp. 379–386. IEEE Comp. Soc. Press, 2003. 87

[32] SUBHASH KHOT AND NISHEETH VISHNOI: The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proc. 46th FOCS*, pp. 53–62. IEEE Comp. Soc. Press, 2005. [doi:10.1109/SFCS.2005.74]. 86, 103

[33] ROBERT KRAUTHGAMER AND YUVAL RABANI: Improved lower bounds for embeddings into L_1. In *Proc. 17th Symp. on Discrete Algorithms (SODA)*, pp. 1010–1017. ACM Press, 2006. [doi:10.1145/1109557.1109669]. 86, 103

[34] JEAN-LOUIS KRIVINE: Sur la constante de Grothendieck. *Comptes Rendus Acad. Sci. Paris Sér. A-B*, 284:445–446, 1977. 88, 112

[35] MICHEL LEDOUX AND MICHEL TALAGRAND: *Probability in Banach Spaces*. Springer, 1991. 94, 101

[36] ALEXANDRE MEGRETSKI: Relaxations of quadratic programs in operator theory and system analysis. In *Systems, Approximation, Singular Integral Operators, and Related Topics: International Workshop on Operator Theory and Applications*. Birkhauser, 2001. 87

[37] ELCHANAN MOSSEL, RYAN O'DONNELL, AND KRZYSZTOF OLESZKIEWICZ: Noise stability of functions with low influences: Invariance and optimality. In *Proc. 46th FOCS*, pp. 21–30. IEEE Comp. Soc. Press, 2005. To appear, *Annals of Mathematics*. [doi:10.1109/SFCS.2005.53]. 87

[38] ARKADI NEMIROVSKI, CORNELIS ROOS, AND TAMÁS TERLAKY: On maximization of quadratic form over intersection of ellipsoids with common center. *Math. Program.*, 86(3):463–473, 1999. [doi:10.1007/s101070050100]. 87

[39] YURII NESTEROV: *Global quadratic optimization via conic relaxation*, pp. 363–384. Kluwer Academic Publishers, 2000. 87

[40] RYAN O'DONNELL AND YI WU: An optimal SDP algorithm for Max-Cut, and equally optimal Long Code tests. In *Proc. 40th STOC*, pp. 335–344. ACM Press, 2008. [doi:10.1145/1374376.1374425]. 88

[41] PRASAD RAGHAVENDRA: Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th STOC*, pp. 245–254. ACM Press, 2008. [doi:10.1145/1374376.1374414]. 89

[42] PRASAD RAGHAVENDRA AND DAVID STEURER: Towards computing the Grothendieck constant. In *Proc. 19th Symposium on Discrete Algorithms (SODA)*, pp. 525–534. ACM Press, 2009. 107

[43] JAMES REEDS: A new lower bound on the real Grothendieck constant. Manuscript (http://www.dtc.umn.edu/~reedsj/bound2.dvi), 1991. 88, 96, 107

[44] LUCA TREVISAN, GREGORY SORKIN, MADHU SUDAN, AND DAVID WILLIAMSON: Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, 2000. [doi:10.1137/S0097539797328847]. 85

[45] URI ZWICK: Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. 31st STOC*, pp. 679–687. ACM Press, 1999. [doi:10.1145/301250.301431]. 85, 86

AUTHORS

Subhash Khot
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
khot@cs.nyu.edu
http://cs.nyu.edu/~khot

Ryan O'Donnell
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
odonnell@cs.cmu.edu
http://www.cs.cmu.edu/~odonnell

ABOUT THE AUTHORS

SUBHASH KHOT is an Associate Professor in the Computer Science Department at New York University, part of the Courant Institute of Mathematical Sciences. He completed his Ph. D. in the summer of 2003 at the Princeton CS Department under the supervision of Sanjeev Arora. He stayed in Princeton for another year as a member of the School of Mathematics at IAS. He has been an Assistant Professor at the College of Computing at Georgia Tech since the fall of 2004, currently on leave.

RYAN O'DONNELL received a B. S. from the University of Toronto in 1999 and a Ph. D. from the MIT Mathematics Department in 2003. His Ph. D. advisor was Madhu Sudan. Following this he was a postdoc at IAS for a year in Avi Wigderson's group, and a postdoc at Microsoft Research for two years in Jennifer Chayes's group. Since 2006 he has been an assistant professor in the Computer Science Department at Carnegie Mellon University. Ryan's research interests include Analysis of Boolean Functions, Hardness of Approximation, Learning Theory, and Probability. He enjoys his spare time.