INTRACTABILITY RESULTS FOR PROBLEMS IN COMPUTATIONAL LEARNING AND APPROXIMATION

A Thesis
Presented to
The Academic Faculty

by

Rishi Saket

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Algorithms, Combinatorics and Optimization

College of Computing
Georgia Institute of Technology
August 2009

INTRACTABILITY RESULTS FOR PROBLEMS IN COMPUTATIONAL LEARNING AND APPROXIMATION

Approved by:

Prof. Subhash Khot, Advisor
College of Computing
Georgia Institute of Technology

Prof. Eric Vigoda
College of Computing
Georgia Institute of Technology

Prof. Santosh Vempala
College of Computing
Georgia Institute of Technology

Prof. Prasad Tetali
School of Mathematics
Georgia Institute of Technology

Prof. Robin Thomas
School of Mathematics
Georgia Institute of Technology

Date Approved: 16 June 2009

To my Parents,

for their unconditional support and affection.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of my teachers and colleagues. I wish to express deep gratitude to my advisor Dr Subhash Khot who has been the source of invaluable guidance during the last five years. His advice and insights, gleaned through numerous discussions with him, have helped me enormously in my research. Even beyond academics, interacting with him has always been intellectually stimulating and has ensured a memorable and enjoyable time as his student.

I am thankful to Dr Robin Thomas, Dr Santosh Vempala, Dr Prasad Tetali, and Dr Eric Vigoda for serving on my dissertation committee and their helpful advice. In addition, thanks are due to Dr Vijay Vazirani, Dr Milena Mihail, Dr Dana Randall and Dr Matthew Baker for their help and support. I also wish to express my gratitude to my collaborators Nisheeth Vishnoi, Nikhil Devanur and Pariskhit Gopalan, and to fellow students Gagan Goel, Ashok Ponnuswami, Deeparnab Chakrabarty and Subrahmanyam Kalyanasundaram.

Lastly, I am grateful to my family for their endless support, understanding and affection throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In this thesis we prove intractability results for some well studied problems in computational learning and combinatorial optimization.

Minimizing and Learning DNF Expressions. We study the problem of finding the minimum size DNF formula for a function $f : \{0,1\}^d \mapsto \{0,1\}$ given its truth table. We show that unless NP$\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$), there is no polynomial time algorithm that approximates this problem to within factor $d^{1-\varepsilon}$ where $\varepsilon > 0$ is an arbitrarily small constant. Our result essentially matches the known $O(d)$ approximation for the problem.

We also study the learnability of small size DNF formulas. We show that assuming NP$\not\subseteq$ RP, for arbitrarily small constant $\varepsilon > 0$ and any fixed positive integer t, a two term DNF cannot be PAC-learnt in polynomial time by a t term DNF to within $\frac{1}{2} + \varepsilon$ accuracy. Under the same complexity assumption, we show that for arbitrarily small constants $\mu, \varepsilon > 0$ and any fixed positive integer t, an AND function (i.e. a single term DNF) cannot be PAC-learnt in polynomial time under adversarial μ-noise (also known as agnostic learning) by a t-CNF to within $\frac{1}{2} + \varepsilon$ accuracy. Our results improve upon the previously known hardness for these problems [1, 30, 31].

Learning Intersection of Two Halfspaces. We show that unless NP= RP, it is hard to PAC-learn intersection of two halfspaces in $\mathbb{R}^n$ using a hypothesis which is a function of up to ℓ linear threshold functions for any integer ℓ to within accuracy of $\frac{1}{2} + \varepsilon$ for any constant $\varepsilon > 0$. Specifically, we show that for every integer ℓ and an arbitrarily small constant $\varepsilon > 0$, unless NP= RP, no polynomial time algorithm can distinguish whether there is an intersection of two halfspaces that correctly classifies a given set of labeled points in $\mathbb{R}^n$, or whether any function of ℓ linear threshold functions can correctly classify at most $\frac{1}{2} + \varepsilon$ fraction of the points. Our result is optimal up to an arbitrarily small constant $\varepsilon > 0$, and improves upon the previous NP-hardness for this problem [1].

Reconstructing Multivariate Polynomials over $\mathbb{F}[2]$. We study the polynomial reconstruction problem for low-degree multivariate polynomials over $\mathbb{F}[2]$. In this problem, we are given a set of points $\mathbf{x}_i \in \mathbb{F}[2]^n$ and target values $\zeta_i \in \mathbb{F}[2]$ for each of these points, with the strong promise promise that there is a linear polynomial over $\mathbb{F}[2]$ that agrees with at least $1-\varepsilon$ fraction of the point-value pairs. Our goal is to find a degree d polynomial that has good agreement with the set of point-value pairs. We show that it is NP-hard to find a polynomial that agrees with more than $1-2^{-d}+\delta$ fraction of the pairs for any constants $\varepsilon, \delta > 0$ and positive integer d. This extends the previously known hardness of approximation (or even NP-completeness) for the case when $d = 1$, which follows from a celebrated result of Håstad [38]. In the setting of Computational Learning, our result shows the hardness of agnostic learning of parities, where the learner is allowed a low-degree polynomial over $\mathbb{F}[2]$ as a hypothesis.

SDP Integrality Gaps with Local ℓ_1-Embeddability. We construct integrality gap instances for SDP relaxation of the Maximum Cut and the Sparsest Cut problems. If the triangle inequality constraints are added to the SDP, then the SDP vectors naturally define an n-point negative type metric where n is the number of vertices in the problem instance. Our gap-instances satisfy a stronger constraint that every sub-metric on $t = O((\log\log\log n)^{\frac{1}{6}})$ points is isometrically embeddable into ℓ_1. The local ℓ_1-embeddability constraints are implied when the basic SDP relaxation is augmented with t rounds of the Sherali-Adams LP-relaxation [75].

For the Maximum Cut problem, we obtain an optimal gap of $\alpha_{GW}^{-1} - \varepsilon$, where α_{GW} is the Goemans-Williamson constant [33] and $\varepsilon > 0$ is an arbitrarily small constant. For the Sparsest Cut problem, we obtain a gap of $\Omega((\log\log\log n)^{\frac{1}{13}})$. The latter result can be rephrased as a construction of an n-point negative type metric such that every t-point sub-metric is isometrically ℓ_1-embeddable, but embedding the whole metric into ℓ_1 incurs distortion $\Omega((\log\log\log n)^{\frac{1}{13}})$.

Integrality Gap for Uniform Sparsest Cut. Arora, Rao and Vazirani [7] showed that the standard semi-definite programming relaxation of the Uniform Sparsest Cut

problem with the *triangle inequality* constraints has an integrality gap of $O(\sqrt{\log n})$. They conjectured that the gap is bounded from above by a constant. In this study, we disprove this conjecture (referred to as the ARV-Conjecture) by constructing an $\Omega(\log \log n)$ integrality gap instance. Khot and Vishnoi [54] had earlier disproved the *non-uniform* version of the ARV-Conjecture.

CHAPTER I

INTRODUCTION

In this introductory chapter we first describe the different topics of interest to us – focusing on the specific problems studied in this thesis – their relevance, the related previous research and the progress made in this thesis. The first few sections of this chapter give the background for our study while the last section is devoted to describing the contributions of this thesis.

1.1 Minimizing and Learning DNF Expressions

A *literal* is a boolean variable or its negation, a *term* is an AND (conjunction) of literals and a *clause* is an OR (disjunction) of literals. A boolean formula in the *disjunctive normal form* (DNF) is given by an OR of terms. Similarly, a formula in *conjunctive normal form* (CNF) is given by an AND of clauses. The *size* of a DNF formula is the number of terms it contains. It is a fundamental fact that any given boolean function can be represented as an equivalent DNF formula (not necessarily unique). The disjunctive normal form is one of the most widely studied representations of boolean functions. In this section we shall describe two classes of problems related to DNF formulas that we study in this thesis.

1.1.1 Minimizing DNF Expressions

We study the following problem which we denote by TT-MinDNF: given the truth table of a function $f : \{0,1\}^d \mapsto \{0,1\}$, to find an equivalent DNF formula for f of minimum size. This problem has been well studied in computer science, and in the next few paragraphs we recall the known results on it.

TT-MinDNF was first studied by Quine [66, 67] in the context of mathematical logic and later by McCluskey[62] in relation to circuit design and both discovered a heuristic to solve the problem. Since then, a large number of heuristics and software tools have been

developed; we refer the interested reader to [22] for a survey.

TT-MinDNF is a special case of the SET-COVER problem. The greedy set cover algorithm gives an $O(\log N) = O(d)$ approximation and runs in time polynomial in N where $N = 2^d$ is the size of the truth table. One the hardness side, the problem was proved to be NP-complete by Masek [61]. Czort [23] showed that unless P = NP, TT-MinDNF cannot be approximated efficiently to within any additive constant. Recently, Feldman [29] showed that TT-MinDNF cannot be approximated to within factor d^γ in polynomial time for some constant $\gamma > 0$ unless P = NP. Allender, Hellerstein, McCabe, Pitassi and Saks [2] independently obtained the same inapproximability result under a stronger assumption that NP $\not\subseteq$ DTIME($n^{\text{poly}(\log n)}$). The constant γ in both results is unspecified; it depends on the parameters of Raz's parallel repetition theorem [71] and is presumably very small.

1.1.2 Learning DNF Expressions

Learning DNFs is a central problem in learning theory. Valiant [79] defined a widely studied model of learning, namely the Probably Approximately Correct (PAC) model. In this thesis we study the problem of (PAC) learning DNFs of small size using a somewhat richer representation classes. More specifically, we study the learnability of (i) a 2-term DNF by a constant term DNF, and (ii) an AND formula (i.e. a 1-term DNF) by a CNF with constant clause size under adversarial noise[1]. In the next few paragraphs we recall the known results for learning DNFs.

Valiant [79] showed that for every constant $k \geq 1$, k term DNF can be PAC learnt in polynomial time by a k-CNF, i.e. a CNF with at most k literals in each clause. For unrestricted DNFs (that is when the number of terms could be polynomially large in the number of variables n), the best learning algorithm runs in time $2^{O(n^{1/3}\log n)}$ due to Klivans and Servedio [56]. For learning under uniform distribution, Jackson [41] showed that unrestricted DNFs can be learnt with membership queries, i.e. the algorithm can query for the value of the function at a point. Alekhnovich, Braverman, Feldman, Klivans and Pitassi [1] gave an $n^{O(\sqrt{n \log n})}$ time algorithm to properly learn unrestricted DNFs, i.e. when the

[1]Learning under adversarial noise is also referred to as agnostic learning

hypothesis is also a DNF.

On the hardness side, Pitt and Valiant [65] showed that unless NP = RP there is no efficient algorithm to PAC learn s-term DNF by an s-term DNF where s is unrestricted, i.e. $2 \leq s \leq n^c$, for any constant $c > 0$. In particular, Alekhnovich *et al.* [1] showed that unless NP = RP, there is no efficient algorithm to learn a 2 term DNF by a k term DNF for any constant k. Nock, Jappy and Sallantin [64] showed that unless NP $\subseteq$ ZPP, given constants $0 \leq \alpha \leq 1 + \frac{1}{145}$ and $\beta \geq 0$, there is no efficient algorithm to PAC-learn n^c term DNF with $n^{\alpha c+\beta}$ term DNF. This was improved by Alekhnovich *et al.* [1] who showed that unless NP = RP, for any given constant $\alpha \geq 0$, n^c term DNF cannot be efficiently learnt by a $n^{\alpha c}$ term DNF. Their result rules out polynomial time proper PAC learning of DNFs, unless NP = RP. This was further strengthened by Feldman [29] to the case when the algorithm even has access to membership queries.

We note that the above mentioned intractability results rule out (under appropriate complexity assumptions) a learning algorithm that learns within error $\frac{1}{\text{poly}(n)}$, but do not rule out a learning algorithm that learns within constant error (say within 1%). In other words, for the underlying optimization problem of finding a DNF formula consistent with the maximum number of given set of labeled examples, these are NP-hardness results and do not give APX-hardness. Another reason to study stronger inapproximability is that given an algorithm to PAC-learn a $(\frac{1}{2} + \varepsilon)$-consistent hypothesis (commonly referred to as *weak learning*), using boosting techniques [74] it can used to efficiently find a $(1 - \delta)$ -consistent hypothesis[2]. A hardness result for weak learning provides evidence against such boosting based approaches. For the problem of learning an AND by an AND under adversarial noise, a $\frac{1}{2} + \varepsilon$ inapproximability is known [30, 31]. Recently, [32] improved upon this to show that under adversarial noise an AND is hard to learn by a halfspace within accuracy $\frac{1}{2} + \varepsilon$.

The above described hardness of learning results only rule out, under complexity assumptions, efficient learning using a restricted hypothesis class. One would ideally prefer

[2]After applying the boosting algorithm, the hypothesis class is now a majority over a set of hypotheses used in the weak learning algorithm.

an intractability result extending to all possible efficiently computable hypotheses. However, all known hardness of learning results based on standard complexity assumptions such as NP$\neq$ RPpertain only to restricted hypotheses. Moreover, recently Applebaum, Barak and Xiao [4] showed that under standard complexity assumptions hardness of learning results for unrestricted hypotheses are unlikely to be proved. In light of the above we believe intractability results for restricted hypotheses to be interesting, and as part of this thesis we study this line of research for several problems including learning DNF expressions.

1.2 Learning Intersection of Two Halfspaces

A halfspace in $\mathbb{R}^n$ is given be the set $\{\mathbf{x} \mid \langle \mathbf{r}, \mathbf{x} \rangle \leq c\}$ for some non-zero vector $\mathbf{r}$ and a number c. In this thesis we study the intractability of PAC-learning an intersection of two halfspaces by (say) an intersection of constantly many halfspaces. The problem of learning a halfspace or an intersection of a small number of halfspaces is an extremely well-studied problem in machine learning, with several applications to computer vision [59], artificial intelligence [63] and data mining [72]. We recall the known results for learning halfspaces.

It is well-known that a single halfspace can be PAC-learnt efficiently by sampling a polynomial number of data points and finding a separating hyperplane via linear programming [15]. Blum, Frieze, Kannan, and Vempala [11] showed how to learn a single halfspace even in presence of random classification noise, whereas Kalai, Klivans, Mansour and Servedio [43] gave polynomial time algorithm for learning a single halfspace in presence of adversarial noise under certain distributional assumptions.

For learning intersection of halfspaces, algorithms are known for various special cases. When the data points are drawn from the uniform distribution over the unit ball, Blum and Kannan [13] and Vempala [81] gave algorithms to PAC-learn intersection of a constant number of halfspaces. For the uniform distribution over the boolean hypercube, Klivans, O'Donnell and Servedio [55] obtained an algorithm for learning intersection of constant number of halfspaces. Arriaga and Vempala [9] and Klivans and Servedio [56] gave algorithms for learning intersection of halfspaces when no data point is too close to any separating hyperplane (i.e. the problem instance has a good *margin*). However the general problem of

learning intersection of halfspaces remains an open problem.

On the intractability side, Feldman, Gopalan, Khot and Ponnuswami [31] and Guruswami and Raghavendra [37] independently proved a $\frac{1}{2} + \varepsilon$ inapproximability for PAC-learning a single halfspace by a halfspace under adversarial noise. [3]. This has been recently strengthened by [32] who prove a similar hardness for learning an AND (which is a special case of a halfspace) by halfspace under adversarial noise. Note that the adversarial noise is necessary in these results, since via linear programming, one can always efficiently find a halfspace that correctly classifies *all* the points, if one exists. These results are optimal, since one can easily classify $\frac{1}{2}$ fraction of the data points correctly, by taking an arbitrary halfspace or its complement as a hypothesis.

However, a similar optimal result was not known for learning intersection of (two) halfspaces. Note that we can hope to obtain hardness for learning intersection of (two) halfspaces *without* any adversarial noise. Blum and Rivest [14] showed that it is NP-hard to learn the intersection of two halfspaces with intersection of two halfspaces, and Alekhnovich, Braverman, Feldman, Klivans and Pitassi [1] proved a similar result even when the hypothesis is an intersection of ℓ halfspaces for any constant ℓ. Both the results are only NP-hardness results and do not prove APX-hardness for the underlying optimization problem.

In a different line of work, under cryptographic assumptions, Klivans and Sherstov [50] showed that there is no polynomial time algorithm to PAC-learn intersection of n^{ε} halfspaces, and the result holds without any restriction on the hypothesis class.

1.3 Reconstructing Multivariate Polynomials over $\mathbb{F}[2]$

We study the intractability of the Polynomial Reconstruction problem PolyRec(d) for multivariate polynomials in n variables over $\mathbb{F}[2]$ of degree at most d, for d constant. The input to this problem is a set of point-value pairs $\{(\mathbf{x}^i, \zeta^i)\}_{i=1}^m$ where $\mathbf{x}^i \in \mathbb{F}[2]^n$ and $\zeta^i \in \mathbb{F}[2]$ and a degree bound d. The goal is to find the multivariate polynomial $P(X_1, \ldots, X_n)$ of degree at most d that satisfies $P(\mathbf{x}^i) = \zeta^i$ for most points $\mathbf{x}^i$. We will allow the possibility that the same vector $\mathbf{x}$ is repeated multiple times (with possibly different labels ζ). In

[3] The result of Guruswami and Raghavendra [37] holds even when the points are from a boolean hypercube.

addition to being a very natural problem, polynomial reconstruction has found applications in several areas of theoretical computer science including computational complexity, coding theory, derandomization and computational learning. In the next few paragraphs we shall recall the previous related work on this problem, especially in the context of computational learning.

The problem of learning parity (linear) functions over $\{0,1\}^n$ in the presence of classification noise is a central problem in computational learning. This is equivalent to learning a linear form over $\mathbb{F}[2]$ in the presence of classification noise and is, therefore, another instance of multivariate polynomial reconstruction - where the point-value pairs are drawn from a noisy linear form. Two kinds of noise models have been studied: in the random classification noise model, the label of each example is flipped independently with probability $\eta < \frac{1}{2}$ before it is given to the learner. In the agnostic learning model which allows worst-case noise, an adversary changes the labels of some η fraction of the points in $\{0,1\}^n$ before the points are presented to the learner. This problem is equivalent to the well-studied problem of decoding random linear codes in coding theory.

While both these problems are widely believed to be hard, there is a considerable gap in our understanding of their complexity. For random classification noise, the best known algorithm due to Blum, Kalai and Wasserman runs in time $2^{O(n/\log n)}$ for any distribution [12]. A $2^{O(n/\log n)}$ algorithm for learning parity with adversarial noise under the uniform distribution was given recently by Feldman *et al.* [31]. Their algorithm is a proper learning algorithm which produces a parity as hypothesis. The question of whether sub-exponential agnostic learning of parity is possible under other distributions is wide open. The problem of proper learning of monomials with adversarial noise, within accuracy of $\frac{1}{2}+\varepsilon$ is known to be NP-hard [30, 31, 51], whereas Kalai *et al.* [43] give a $2^{O(\sqrt{n})}$ non-proper learning algorithm for all distributions which produces the sign of a real polynomial as its hypothesis. Feldman *et al.* asked whether parity with adversarial noise is hard to learn even using low-degree $\mathbb{F}[2]$ polynomials as hypothesis [31].

In contrast with the progress on the algorithmic side, relatively few negative results are known for polynomial reconstruction. For linear polynomials in n variables, a tight hardness

result follows from the celebrated 3-bit PCP of Håstad [38], which implies a inapproximability factor of $\frac{1}{2} + \varepsilon$ for the accuracy of learning a parity (over $\mathbb{F}[2]$) with a parity under adversarial noise. For $d = 2$ and higher, we are unaware of (even) any previous NP-hardness result for $\mathbb{F}[2]$ or even polynomial-sized fields. Goldreich *et al.* show that the polynomial reconstruction problem is NP-hard for univariate polynomials over exponentially large fields [34].

1.4 SDP Integrality Gaps with Local ℓ_1 Embeddability

For several well-studied problems such as Maximum Cut and Sparsest Cut, the best known approximation algorithms are based on a Semi-definite Programming relaxation. For Maximum Cut, the basic SDP relaxation suffices to achieve the best-known approximation guarantee whereas for the Sparsest Cut problem, adding additional constraints called the *triangle-inequality constraints* provably improves the approximation guarantee. Once these constraints are added, the SDP vectors naturally define a so-called negative type (or squared-ℓ_2) metric, and such metrics can be embedded *well* into the class of ℓ_1 metrics. After the ℓ_1-embedding is carried out, it is straightforward to output a good cut since n-point ℓ_1 metrics are precisely the convex combinations of *cut-metrics*. In general, it is a worthwhile (and of great current interest) goal to investigate whether stronger LP/SDP relaxations help, by adding (say polynomially many) natural constraints that an integral solution must satisfy. One natural family of constraints is to require that the negative type metric defined by the SDP vectors has an additional property that every sub-metric on t points embeds isometrically into ℓ_1. This certainly makes sense for the Sparsest Cut problem since we would like the metric to be as close to ℓ_1 as possible. The local ℓ_1-embeddability condition can be enforced by adding $n^{O(t)}$ LP-constraints and requiring that the LP solution is *consistent* with the SDP vectors. Concretely, it suffices to add all LP constraints generated by t rounds of the Sherali-Adams LP hierarchy (see Chapter 5 for a detailed description).

In this thesis, we investigate whether this approach is likely to yield *good* approximations. The two problems that we focus on are Maximum Cut and Sparsest Cut and in the

next few paragraphs we recall the previous work on these problems.

1.4.1 Maximum Cut

Given a graph $G(V, E)$ with a non-negative weight function **wt** on the edges, the problem of Maximum Cut is find a cut that maximizes the weight of the crossing edges. For this problem, a break-through result of Goemans and Williamson [33] showed that the integrality gap of the basic SDP relaxation is at most α_{GW}^{-1} where $\alpha_{GW} \approx 0.878$ is the optimum of a certain trigonometric function. Feige and Schechtman [28] gave a matching integrality gap instance with gap $\alpha_{GW}^{-1} - \varepsilon$. Khot and Vishnoi [54] showed that even after adding the triangle inequality constraints, the integrality gap is still lower bounded by $\alpha_{GW}^{-1} - \varepsilon$. This result is quite involved and especially, the proof that the triangle inequality constraints hold, is by brute-force with little intuitive explanation. In an incomparable result, Charikar, Makarychev, and Makarychev [17] gave $(1 - \varepsilon, \frac{1}{2} + \varepsilon)$-integrality gap for the Sherali-Adams hierarchy even with $n^{c(\varepsilon)}$ rounds. However, Goemans and Williamson showed that for the basic SDP relaxation, the gap cannot be stronger than $(1 - \varepsilon, 1 - \Omega(\sqrt{\varepsilon}))$ and thus the Sherali-Adams relaxation is qualitatively different from the SDP relaxation. We investigate the following natural question: what is the integrality gap if we combine the SDP with t rounds of the Sherali-Adams LP hierarchy?

1.4.2 Sparsest Cut

Given a graph $G(V, E)$ with non-negative weights **wt** and demands **dem** on edges, the problem of Sparsest Cut is to find a cut $(S, \overline{S})$, $S \subseteq V$, to minimize the following quantity,

$$\frac{\sum_{e \in E(S,\overline{S})} \mathbf{wt}(e)}{\sum_{e \in E(S,\overline{S})} \mathbf{dem}(e)}.$$

In the special case of all the demands **dem** being equal the problem is referred to as Uniform Sparsest Cut.

For the Uniform Sparsest Cut problem on n-vertex graphs, the basic SDP relaxation is very poor and has an integrality gap of $\Omega(n)$. In a recent break-through, Arora, Rao, and Vazirani [7] showed that the gap improves to $O(\sqrt{\log n})$ after adding the triangle inequality

constraints (i.e. the distance $d(\cdot,\cdot)$ is required to be a metric). Arora, Lee, and Naor [5] proved essentially the same upper bound even for the more general non-uniform Sparsest Cut problem. In fact, it had been conjectured earlier by Goemans and Linial that the integrality gap for non-uniform Sparsest Cut problem is at most a universal constant. This is equivalent to a conjecture that n-point negative type metrics embed into ℓ_1 with constant distortion. Khot and Vishnoi [54] disproved the conjecture by constructing an n-point negative type metric with ℓ_1-distortion at least $(\log\log n)^{\Omega(1)}$. The lower bound was subsequently improved to $\Omega(\log\log n)$ by Krauthgamer and Rabani [57], and to $\Omega(\log\log n)$ for the Uniform Sparsest Cut by Devanur *et al.* [25]. Lee and Naor [58] proposed a different counter-example to the Goemans-Linial conjecture, and the works of Cheeger, Kleiner, and Naor [19, 20, 21] showed that this counter-example gives a further improved lower bound of $(\log n)^{\Omega(1)}$ (the upper bound is $\tilde{O}(\sqrt{\log n})$ as mentioned before).

In light of the extensive research on the Sparsest Cut integrality gap, it is natural to investigate whether the integrality gap becomes a constant if we require the negative type metric $d(\cdot,\cdot)$ to have the property that every sub-metric on t points embeds isometrically into ℓ_1.

Connections to inapproximability results via the Unique Games Conjecture

All the results mentioned in this section so far are intimately connected with the Unique Games Conjecture (UGC) of [47]. The conjecture states that approximating the Unique Games problem (see Definition 5.2.3) is NP-hard and is proposed as an avenue towards proving strong inapproximability results for many NP-hard problems. Indeed, assuming the conjecture, Khot *et al.* [48] proved that it is NP-hard to approximate the Maximum Cut problem within any factor strictly less than α_{GW}^{-1}, which betters the hardness of approximation factor (not based on UGC) of $\frac{17}{18}-\varepsilon$ given by Håstad [38]. This means that, assuming the UGC and that P $\neq$ NP, any LP/SDP relaxation for Maximum Cut with polynomially many constraints, must have integrality gap arbitrarily close to α_{GW}. Thus integrality gap instances for potentially more and more powerful LP/SDP relaxations give more and more evidence towards the truth of the UGC.

Khot and Vishnoi [54] used this connection in the reverse direction to actually construct integrality gap instances for cut-problems. They (and independently Chawla *et al.* [18]) gave a *reduction* from the Unique Games problem to Sparsest Cut and showed that if the UGC is true, then Sparsest Cut has no constant approximation which is stronger than the recent result (not based on the UGC) of Ambühl *et al.* [3] which rules out the existence of a PTAS for the Uniform Sparsest Cut problem. Hence, if UGC is true then Goemans-Linial conjecture must be false. This observation led [54] to a construction of an integrality gap instance for the Unique Games problem (see the SDP in Figure 1) and then they *translated* this instance into an integrality gap instance for Sparsest Cut via the reduction alluded to before. A nice feature of the reduction is that it allows a translation of the SDP solution as well, i.e. starting with a vector solution for the Unique Games SDP, one can construct a vector solution for Sparsest Cut SDP in a natural way. However an unsatisfying feature of [54] is that there is no intuitive reason why the Sparsest Cut vector solution obeys triangle inequalities.

We now explain how this work fits with Raghavendra's recent result [68]. Raghavendra shows that for every *constraint satisfaction problem*, there is a certain generic relaxation such that, any integrality gap instance for this relaxation with gap α, can be translated into a UGC based hardness result with the hardness factor same as α. The relaxation he uses is exactly the combination of a basic SDP and a constant number of rounds of the Sherali-Adams LP (the number of rounds is at most $O(k+q)$ for a k-ary CSP over q-ary alphabet)! An implication of his result is that (assuming UGC and that P $\neq$ NP) adding more constraints to the generic relaxation does not help. Therefore, integrality gap examples for SDP relaxations strengthened with Sherali-Adams LP constraints would partially confirm Raghavendra's implication, namely that adding more Sherali-Adams rounds to the generic relaxation does not help.

Other LP and SDP Hierarchies

Finally, a few words about other LP and SDP hierarchies are in order. Recent works have obtained integrality gap results for many different problems (cut problems, vertex cover,

independent set, 3SAT etc.) for LP and/or SDP relaxations in different hierarchies, i.e. Lovász-Schrijver, Sherali-Adams, and Lasserre. A full overview of these results is beyond the scope of this introduction and we do not attempt it here. We would like to mention however that the Lasserre hierarchy is the most powerful one and it remains a challenging open problem to prove Lasserre integrality gaps. A t-round Lasserre includes, for example, the basic SDP as well as t rounds of Sherali-Adams LP. It is conceivable that the techniques in this thesis could be applied towards obtaining strong Lasserre integrality gaps.

1.5 Integrality Gap for Uniform Sparsest Cut

Given an n-vertex graph $G(V, E)$, the *sparsity* of a cut $(S, \overline{S})$ is defined as $\frac{E(S,\overline{S})}{|S||\overline{S}|}$, where $E(S, \overline{S})$ denotes the set of edges crossing the cut. The Uniform Sparsest Cut problem is to find a cut with minimum sparsity. In the related problem of b-Balanced Separator, for some fixed constant $0 < b \leq 1/2$, the objective is to find a cut $(S, \overline{S})$, with $|S|, |\overline{S}| \geq bn$, which minimizes the number of edges cut. It is well-known that a factor $f(n)$ approximation algorithm for Uniform Sparsest Cut can be used iteratively to design a factor $O(f(n))$ (pseudo-) approximation algorithm for Balanced Separator: Given a graph that has a $(\frac{1}{2}, \frac{1}{2})$ partition cutting an α fraction of the edges, the algorithm produces a $(\frac{1}{3}, \frac{2}{3})$ partition that cuts at-most $O(f(n)\alpha)$ fraction of the edges. Such partitioning algorithms are very useful as sub-routines in designing graph theoretic algorithms via the divide-and-conquer paradigm. A comprehensive survey of the applications of these two important problems in computer science can be found in [76].

In this thesis we study the integrality gap for the Uniform Sparsest Cut and Balanced Separator SDP relaxations equipped with triangle inequalities. The background for the Uniform Sparsest Cut and related problems is given in Section 1.4.2. As mentioned, Arora, Rao and Vazirani [7] gave an upper bound of $O(\sqrt{\log n})$ for the integrality gap of the Uniform Sparsest Cut SDP relaxation with triangle inequalities. The authors also conjectured the integrality gap of this relaxation to be $O(1)$, which we shall refer to as ARV-Conjecture. Essentially, our study is concerned with the validity of the ARV-Conjecture.

1.6 Contributions of this Thesis

In the remainder of this chapter we state the contributions of this thesis in the study of the topics described in Sections 1.1- 1.5.

1.6.1 Hardness of Minimizing and Learning DNF Expressions

In this thesis we prove almost essentially optimal hardness results for the problems of minimizing DNF expressions, and learning small DNFs. We state the results in the next few paragraphs. These appear as part of joint work with Subhash Khot [51] and the proofs are given in Chapter 2.

1.6.1.1 Minimizing DNF Expressions. We prove the following theorem regarding the intractability of TT-MinDNF.

Theorem 1.6.1. *For any $\varepsilon > 0$, there is no polynomial time algorithm that, given the truth table of a boolean function $f : \{0,1\}^d \mapsto \{0,1\}$, over d variables, computes an equivalent DNF formula for f of size within $d^{1-\varepsilon}$ of the minimum size equivalent DNF formula for f, unless* NP $\subseteq$ DTIME$(n^{\mathrm{poly}(\log n)})$.

The above theorem gives a $d^{1-\varepsilon}$ hardness factor for TT-MinDNF, and since there is a $O(d)$ approximation algorithm for it, the hardness of approximation factor is essentially optimal. Our reduction actually proves hardness of approximation factor of $d^{1-\varepsilon}$ for a related problem of covering a subset $\mathcal{S}$ of the hypercube $\{0,1\}^d$ using minimum number of terms from a given set $\mathcal{T}$ of terms. Feldman [29] showed that this implies the same hardness of approximation factor for TT-MinDNF.

1.6.1.2 Learning small DNF formulas. We prove the following two hardness results relating to learning DNF formulas in the PAC model.

Theorem 1.6.2. *For any $\varepsilon > 0$ and any given positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is a 2 term DNF formula that is consistent with all the examples of $\mathcal{D}$, unless* NP

$=$ RP *there is no polynomial time PAC learning algorithm to compute a DNF formula of up to t terms that is consistent with the examples with probability $\frac{1}{2}+\varepsilon$ under the distribution $\mathcal{D}$.*

Theorem 1.6.3. *For any constants $\varepsilon, \mu > 0$ and any positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is an AND formula that is consistent with the examples with probability (under $\mathcal{D}$) at least $1-\mu$, unless* NP $=$ RP *there is no polynomial time PAC learning algorithm to compute a t-CNF formula, i.e. a CNF formula with at most t literals in each clause, that is consistent with examples with probability(under $\mathcal{D}$) at least $\frac{1}{2}+\varepsilon$.*

The results are essentially optimal since a trivial formula that is either the constant 1 or the constant 0 satisfies the examples with probability $\frac{1}{2}$. The distributions in both the theorems are supported over polynomially (in n) many points of the hypercube, and therefore they can be given explicitly.

1.6.2 Hardness of Learning Intersection of Two Halfspaces

In this thesis we prove the following hardness result regarding the learnability of intersection of two halfspaces.

Theorem 1.6.4. *Let ℓ be any fixed integer and $\varepsilon > 0$ be an arbitrarily small constant. Then, given a set of labeled points in $\mathbb{R}^n$ with a guarantee that there is an intersection of two halfspaces that classifies all the points correctly, there is no polynomial time algorithm to find a function f of up to ℓ linear threshold functions that classifies $\frac{1}{2}+\varepsilon$ fraction of points correctly, unless* NP $=$ RP.

Note that the above theorem implies that for any constant $\varepsilon > 0$ and $\ell \in \mathbb{Z}^+$, there is no polynomial time PAC learning algorithm to learn an intersection of two halfspaces by a function of up to ℓ linear thresholds, unless NP $=$ RP.

We state our result in terms of functions of ℓ linear threshold functions. This encompasses hypotheses such as intersection of ℓ halfspaces since a linear threshold function $sgn(c - \langle \mathbf{r}, \mathbf{x} \rangle)$ is same as the halfspace $\{\mathbf{x} \mid \langle \mathbf{r}, \mathbf{x} \rangle \leq c\}$ for any (unit) vector $\mathbf{r}$ in $\mathbb{R}^n$ and real

number c. Note that the result holds with perfect completeness, i.e. the problem is hard even when an intersection of two halfspaces is guaranteed to classify *every* point correctly. The result is essentially optimal since an arbitrary halfspace or its complement has a success rate of $\frac{1}{2}$ on any given data set. It provides evidence that the approach of weak learning intersection of two halfspaces with a function of a constant number of halfspaces followed by boosting may not work.

This result appears as part of joint work with Subhash Khot [52] and the detailed reduction is given in Chapter 3 of this thesis.

1.6.3 Hardness of Reconstructing Multivariate Polynomials

In this thesis we prove the following hardness result for the Polynomial Reconstruction problem PolyRec(d) as described in Section 1.3.

Theorem 1.6.5. *For any constants $\varepsilon, \delta > 0$ and positive integer d, given an instance of* PolyRec(d) *over $\mathbb{F}[2]$, with the guarantee that there is a linear polynomial satisfying $P(\mathbf{x}^i) = \zeta^i$ for $1 - \varepsilon$ fraction of the points, it is* NP*-hard to find a polynomial P of degree at most d that satisfies $P(\mathbf{x}^i) = \zeta^i$ for at most $1 - 2^{-d} + \delta$ fraction of the points.*

In the case $d = 1$, our result matches the tight bound of $\frac{1}{2} + \delta$ for linear equations which follows from Håstad's work [38], but via a very different proof technique. To our knowledge, for $d \geq 2$, this was the first hardness of approximation or even NP-completeness for a fixed field. Theorem 1.6.5 gives a strong guarantee: the polynomial fitting the data is linear. This implies the NP-hardness of agnostic learning of parity even if the learning algorithm is allowed $\mathbb{F}[2]$ polynomials of degree d for any constant d. We note, however that the inapproximability factor of $1 - 2^{-d} + \delta$ we obtain is not optimal, and one could expect a factor close to $\frac{1}{2} + \delta$. Subsequent to our results, Viola [82] showed that an independent sum of d pseudo-random generators which fool linear polynomials, fools degree d polynomials. We believe that one can obtain an optimal $\frac{1}{2} + \varepsilon$ hardness factor for PolyRec(d) using this result of [82].

Our result appears as part of joint work with Parikshit Gopalan and Subhash Khot [35] and a detailed proof of Theorem 1.6.5 is presented in Chapter 4 of this thesis.

1.6.4 SDP Integrality Gaps with Local ℓ_1 Embeddability

In this thesis we prove the following results regarding the integrality gap of SDP relaxations strengthened with Sherali-Adams LP constraints for the MAXIMUM CUT and the SPARSEST CUT problems.

Theorem 1.6.6. *Let $\varepsilon > 0$ be an arbitrarily small constant. For the* MAXIMUM CUT *problem on a graph of n vertices, the SDP relaxation augmented with $O((\log\log\log n)^{\frac{1}{6}})$ rounds of Sherali-Adams LP hierarchy has an integrality gap at least $\alpha_{GW}^{-1} - \varepsilon$, where α_{GW} is the Goemans-Willamson constant [33].*

Theorem 1.6.7. *For the* SPARSEST CUT *problem on a graph of n vertices, the SDP relaxation augmented with $O((\log\log\log n)^{\frac{1}{6}})$ rounds of Sherali-Adams has an integrality gap at least $\Omega((\log\log\log n)^{\frac{1}{13}})$. Also, there is an n-point negative type metric such that every sub-metric on $O((\log\log\log n)^{\frac{1}{6}})$ points is isometrically ℓ_1-embeddable, but embedding the whole metric into ℓ_1 incurs distortion $\Omega((\log\log\log n)^{\frac{1}{13}})$.*

Consider the distance $d(u,v) := \|\mathbf{w}_u - \mathbf{w}_v\|^2$ defined on the set of vertices by the SDP vector solution $\{\mathbf{w}_u\}_{u\in V}$. An easy and well-known observation is that (see the last paragraph in Section 5.2.1) if the vector solution is consistent with t rounds of Sherali-Adams solution, then for any set $S \subseteq V, |S| \leq t$, the space $(S, d(\cdot,\cdot))$ embeds isometrically into ℓ_1. In particular, the distance $d(\cdot,\cdot)$ satisfies triangle inequality. As mentioned earlier, the proof in [54] that the triangle inequalities hold is very technical. On the other hand, the construction in this thesis, though not necessarily simpler, is quite intuitive and there is a reasonable explanation why it works. Our construction does use techniques from [54].

We note that in an incomparable result, Charikar, Makarychev, and Makarychev [17] gave integrality gap of $\Omega\left(\sqrt{\frac{\log n}{\log t + \log\log n}}\right)$ for t rounds of the Sherali-Adams hierarchy (without the SDP). This amounts to an ℓ_1 lower bound for (general, not negative-type) metrics such that any sub-metric on t points is isometrically ℓ_1-embeddable. Also, in a very recent work, Raghavendra and Steurer [69] have shown similar SDP integrality gaps for MAXIMUM CUT and SPARSEST CUT augmented with $\Omega((\log\log n)^{\frac{1}{4}})$ rounds of Sherali-Adams constraints, which is stronger than our results in the number of rounds of Sherali-Adams

constraints. The techniques employed in [69] are quite similar to those in this thesis.

Our results appear as part of joint work with Subhash Khot [53]. A detailed description of the proof of Theorem 1.6.6 is given in Chapter 5. The proof of Theorem 1.6.7 is very similar and we give a brief description of it in Section 5.7.

1.6.5 Integrality Gap for Uniform Sparsest Cut

In this thesis we disprove of the ARV-Conjecture. We construct an $\Omega(\log \log n)$ integrality gap instance for Balanced Separator which implies the same gap for Uniform Sparsest Cut.

Theorem 1.6.8. *The standard SDP relaxations of* Uniform Sparsest Cut *and* Balanced Separator *with the triangle inequality constraints, on an n-vertex graph, have an integrality gap of at least $\Omega(\log \log n)$.*

The above theorem proved in this thesis subsumes the results by Khot and Vishnoi [54], and Krauthgamer and Rabani [57]. As in [57, 18], our lower bound proof uses a Fourier analytic theorem of Kahn, Kalai and Linial [42] whereas Khot and Vishnoi use a theorem of Bourgain [16]. A very recent result of Raghavendra and Steurer [69] builds upon this construction and shows an integrality gap example for Balanced Separator and Uniform Sparsest Cut problems with gap of $\Omega(\log \log^{\gamma} n)$ for the SDP relaxation augmented with k-gonal inequalities for $k = O(2^{\log \log^{\delta} n})$ for some constants $\delta, \gamma > 0$. Thus, their integrality gap holds for a stronger SDP relaxation, while losing marginally on the value of the gap.

Our result appears as part of joint work with Nikhil Devanur, Subhash Khot and Nisheeth Vishnoi [25], and a detailed proof of Theorem 1.6.8 is given in Chapter 6.

CHAPTER II

HARDNESS OF MINIMIZING AND LEARNING DNF EXPRESSIONS

In this Chapter we give the proofs of Theorems 1.6.1, 1.6.2 and 1.6.3. The proofs are based on hardness reductions from known intractable problems. We begin by devoting the next section to a brief introduction to the PAC model of learning which forms the context behind the hardness of learning results presented in this thesis.

2.1 The PAC Learning Model

The Probably Approximately Correct (PAC) model was introduced by Valiant [79] and is a widely studied model of learning. In this model a *concept class* $\mathcal{C}$ is a class of functions (concepts) over some domain X and for some range Y. Many well studied concepts are common boolean functions over (say) the boolean hypercube or the real vector space. For every function f in a concept class $\mathcal{C}$ and distribution $\mathcal{D}$ over X, $\mathrm{EX}(\mathcal{D}, f)$ is an *example oracle* which, if queried, outputs an example $(x, f(x))$ where x is chosen at random from the distribution $\mathcal{D}$. The formal definition of PAC learning is given below. For simplicity, we consider only boolean concepts, i.e. $Y = \{0, 1\}$.

Definition 2.1.1. *A concept class $\mathcal{C}$ over a domain X is said to be PAC learnable if there is an randomized algorithm $\mathcal{A}$ such that for every $\varepsilon > 0$, $f \in \mathcal{C} (f : X \mapsto \{0,1\})$ and distribution $\mathcal{D}$ over X, given access to $EX(\mathcal{C}, \mathcal{D})$, it outputs a hypothesis h such that,*

$$\Pr_{\mathcal{D}}[f(x) = h(x)] \geq 1 - \varepsilon.$$

The algorithm A is said to be efficient if it runs in time polynomial in $1/\varepsilon$, the size of the representation of elements of X and the representation of f in the concept class $\mathcal{C}$.

We refer to the parameter $1 - \varepsilon$ as the *accuracy* of the hypothesis h. Note that there is no restriction on the hypothesis h in the PAC model. However, it may be desirable to expect h to be either from the same class $\mathcal{C}$ as f. If such additional condition is imposed

on $\mathcal{A}$ then it is said to *properly* PAC learn $\mathcal{C}$. If on the other hand, the accuracy required is relaxed to be $\frac{1}{2} + \delta$ for some $\delta > \text{poly}(s)$, (where s is the size of the problem) then $\mathcal{A}$ is said to weakly PAC learn $\mathcal{C}$.

A related model known as *agnostic* PAC model of learning was introduced by Haussler [39] and Kearns *et al.* [44]. In this model the unknown function f is not necessarily a member of the class $\mathcal{C}$. The goal is to determine a hypothesis h that approximates, under the distribution $\mathcal{D}$, f almost as well as any member of $\mathcal{D}$ (to within an error of ε). More formally the following condition is enforced on h,

$$\Pr_{\mathcal{D}}[h(x) = f(x)] \;\geq\; \sup_{g \in \mathcal{D}} \Pr[g(x) = f(x)] - \varepsilon.$$

The agnostic PAC model is identical to an adversarial noise model in which the target function f is a version of some $f^* \in \mathcal{C}$ at which has been corrupted by an unknown adversary at ε-fraction of inputs (under $\mathcal{D}$).

In this thesis we study several problems related to PAC learning of concept classes with restricted hypotheses. We show inapproximability results in terms of the accuracy of learning achievable in polynomial time under complexity assumptions. Throughout this thesis when we refer to learning, we imply PAC learning unless explicitly mentioned. In this chapter we prove Theorems 1.6.2 and 1.6.3 on hardness of PAC learning small DNF formulas, in addition to other results. In the next section we provide an overview of the main techniques involved in the proofs contained in this chapter.

2.2 Overview

We restate the main results of this chapter and give an overview of the proof techniques involved. The result for minimizing DNF formulas is a simple reduction from a new Probabilistically Checkable Proof (PCP) that is constructed by a straightforward composition of known PCPs. The other two results are direct reductions from the Label-Cover problem.

2.2.1 Minimizing DNF Expressions

We prove the following theorem.

Theorem. (1.6.1 restated) *For any $\varepsilon > 0$, there is no polynomial time algorithm that, given the truth table of a boolean function $f : \{0,1\}^d \mapsto \{0,1\}$, over d variables, computes an equivalent DNF formula for f of size within $d^{1-\varepsilon}$ of the minimum size equivalent DNF formula for f, unless* NP $\subseteq$ DTIME$(n^{\text{poly}(\log n)})$.

As mentioned in Section, the $d^{1-\varepsilon}$ hardness of approximation factor is essentially optimal for this problem. Our reduction actually proves hardness of approximation factor of $d^{1-\varepsilon}$ for a related problem, PHC-Cover of covering a subset $\mathcal{S}$ of the hypercube $\{0,1\}^d$ using minimum number of terms from a given set $\mathcal{T}$ of terms. Feldman [29] showed that this implies the same hardness of approximation factor for the problem of minimizing the size of DNF formulas.

Overview of Reduction: The reduction proceeds by first constructing a specialized version of a constraint satisfaction problem (or PCP) and then reducing it to PHC-Cover. However, for simplicity let us assume that we begin with a bipartite Label-Cover problem over the label set $[k]$, with n vertices in each bipartition. Consider the vertices of the U layer. It is easy to see that we require at most $\log n$ variables so that every vertex in U is mapped to a unique setting of these variables. Call these variables *vertex variables* for the U layer. In the set of terms $\mathcal{T}$ of the PHC-Cover instance, we would like to have k unique terms for every vertex u in U, corresponding to the k labels for u. For this purpose we create k *label variables*, one for each label. For each vertex u and label i, there is a term which is 1 exactly on the unique setting, corresponding to u, of the vertex variables and when the label variable for label i is set to 0. Therefore for the U layer there are $\log n + k$ variables and nk terms, k for each vertex, where each term is over $\log n$ vertex variables and one label variable. We similarly construct distinct variables and terms for the V layer. In total we have $2(\log n + k)$ variables and $2nk$ terms.

Now, we construct the subset of points of the hypercube to be covered as follows. Pick an edge $e = (u, v)$ and two sets $S_1, S_2 \subseteq [k]$ such that $S_1 \times S_2$ does not contain any satisfying assignment to e. Set the coordinates such that only the terms corresponding to u and v are active. Set the coordinates corresponding to the labels in S_1 (in the U layer) and those

corresponding to labels in S_2 (in the V layer) to be 1. Do this for all edges e of the Label-Cover instance, and all such subsets S_1 and S_2 corresponding to C_e. It is easy to see that for a given edge $e = (u, v)$, if all points corresponding to such sets S_1, S_2 are covered then the set of terms corresponding to u and to v, must 'contain' a labeling to u and v, respectively, satisfying the edge e. Moreover, unless the number of terms chosen to cover the points is large enough, our analysis gives a way to pick a 'good' labeling to the vertices of the Label-Cover instance. Therefore, in the YES case, the number of terms required to cover all points is small, in the NO case it is necessarily large.

While this reduction works even with the standard bipartite Label-Cover, it does not yield the desired hardness of approximation factor. In order to achieve that, we combine it with a multi layered constraint system based on a variant of the query efficient PCP of Samorodnitsky and Trevisan [73]. The PCP we construct is similar to the one constructed by Khot [45] as it uses Hadamard encodings instead of Long Codes. We need this crucially as using Long Codes would blow up the size of the PCP in relation to the size of the label set. In order to use Hadamard encodings, we need to start with an instance with linear constraints. As a result, we lose perfect completeness. However, our reduction tolerates the loss of perfect completeness as long as the completeness parameter is suitably close to 1. In order to achieve this we start the construction of the PCP using the Max-3LIN instance constructed by Khot and Ponnuswami [49], which has completeness very close to 1 which we desire. We also need to ensure a large sized label set. For this purpose, the Hadamard encodings are over an appropriately large field extension of $\mathbb{F}[2]$. The PCP thus constructed is transformed into a multi layered constraint system via standard reductions.

We note that the previous hardness reductions of Feldman [29] and Allender *et al.* [2] used a construction of certain *union free families of sets*, similar to the *partition systems* used in the reductions for the Set-Cover problem [60, 27]. Our result does not need such constructions (which we find interesting, since we in particular obtain $\log^{1-\varepsilon} N$ hardness for Set-Cover without using partition systems). In [29, 2], the parameters involved in constructing union free families limits the hardness factor achievable to $\sqrt{d}$ in addition to the limitation on γ (in the d^γ hardness) imposed by the parameters in Raz's parallel

repetition theorem. Our reduction bypasses both these limitations.

2.2.2 Learning 2-term DNF by t-term DNF

We prove the following theorem.

Theorem. (1.6.2 restated) *For any $\varepsilon > 0$ and any given positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is a 2 term DNF formula that is consistent with all the examples of $\mathcal{D}$, unless* NP = RP *there is no polynomial time PAC learning algorithm to compute a DNF formula of up to t terms that is consistent with the examples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$.*

As mentioned earlier, this result is essentially optimal since a trivial formula that is either the constant 1 or the constant 0 satisfies the examples with probability $\frac{1}{2}$.

Overview of Reduction: Our reduction proves an equivalent result for learning 2-clause CNF by t-clause CNF. We give a direct reduction from the Label-Cover problem with vertex sets U and V, and label sets $[m]$ and $[k]$ respectively. The examples of the distribution $\mathcal{D}$ simulate the junta and consistency tests. We create one coordinate for every vertex and its potential label. So we have $m|U| + k|V|$ coordinates. The 1 examples have the property that there is an edge (u, v) such that all the m coordinates corresponding to u and k coordinates corresponding to v are set to 1 and all other coordinates are set to 0. The 0 examples are constructed by choosing a vertex $u \in U$ and a set $\alpha \subseteq [m]$ and setting all the coordinates of u corresponding to $[m] \setminus \alpha$ to be 1. Moreover for every neighbor v of u, all coordinates corresponding to $\pi_{uv}^{-1}(\alpha)$ are set to 1, where π_{uv} is the projection map for the edge (u, v). All the other coordinates are set to 0.

Suppose there is a labeling σ to the vertices that satisfies all edges. Now consider the clause C_U consisting of the variables corresponding to vertex u and its label $\sigma(u)$ for all $u \in U$. Let clause C_V be similarly defined for V. It is easy to see that the formula $C_U \wedge C_V$ satisfies all the examples. In the NO case we show that if there a t clause CNF that is consistent with the examples with probability at least $\frac{1}{2} + \varepsilon$, then one can construct a

labeling to the vertices of Label-Cover instance which satisfies a significant fraction of edges. This leads to a contradiction if we choose the soundness parameter of the Label-Cover instance to be small enough.

2.2.3 Learning AND by t-CNF under adversarial noise

We prove the following theorem.

Theorem. (1.6.3 restated) *For any constants $\varepsilon, \mu > 0$ and any positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is an AND formula that is consistent with the examples with probability (under $\mathcal{D}$) at least $1 - \mu$, unless* NP = RP *there is no polynomial time PAC learning algorithm to compute a t-CNF formula, i.e. a CNF formula with at most t literals in each clause, that is consistent with examples with probability(under $\mathcal{D}$) at least $\frac{1}{2} + \varepsilon$.*

Again the result is essentially optimal since it is trivial to output a formula that is consistent with half the examples. Moreover, without any noise an AND formula can be properly learnt in polynomial time. Our reduction proves an equivalent result for learning OR by t-DNF, i.e. DNF formula with at most t literals in each term. The reduction is similar to the one described in Section 2.2.2 and starts with an instance of bipartite Label-Cover. In a similar manner the examples simulate the junta and consistency tests, with the property that in the YES instance, the labeling gives a OR formula that is consistent with the examples with probability close to 1. In the NO case, any DNF formula with at most t literals in each term that is consistent with the examples with probability at least $\frac{1}{2} + \varepsilon$ yields a labeling to the vertices of the Label-Cover instance that satisfies a significant fraction of edges, and choosing the soundness parameter of the Label-Cover instance to be small enough, this leads to a contradiction.

Organization of the Chapter. In Section 2.3 we formally define the problems considered, and the tools we require for our reductions. We present the hardness result for minimizing DNF formulas in Section 2.4. It is a reduction from a multi-layered CSP to PHC-Cover. The results for learning 2 term-DNF, learning AND under adversarial noise

and the construction of the multi-layered CSP are presented in Sections 2.5, 2.6 and 2.7 respectively.

2.3 Preliminaries

Let $f : \{0,1\}^d \mapsto \{0,1\}$ be a boolean function. We say that a boolean function g is *equivalent* to f if it agrees with f at every point of the hypercube. A DNF formula is a OR of *terms* where a *term* is an AND of literals. Similarly, a CNF formula is a AND of clauses, where each clause is an OR of literals. We define the problem TT-MinDNF as follows.

Definition 2.3.1. *The problem* TT-MinDNF *is the following: given the truth table of a boolean function f on d variables, to find an equivalent DNF formula ϕ with the minimum number of terms.*

In our reduction we prove a hardness of approximation factor of $d^{1-\varepsilon}$ for any $\varepsilon > 0$, for the partial hypercube cover (PHC-Cover) problem which is defined as follows.

Definition 2.3.2. *The problem* PHC-Cover *is the following: given a subset $\mathcal{S} \subseteq \{0,1\}^d$, and a set of terms $\mathcal{T}$, to find a minimum subset of terms $\mathcal{T}^* \subseteq \mathcal{T}$ that covers all the points in $\mathcal{S}$.*

Feldman [29] showed that a hardness of approximation factor of d^{γ} for PHC-Cover implies same hardness factor for TT-MinDNF, for any constant $\gamma > 0$. Therefore, our result implies hardness of approximation factor of $d^{1-\varepsilon}$ for TT-MinDNF.

We also define the following problems related to learning boolean functions.

Definition 2.3.3. *For any positive integer t, the problem of* Learn-t-term-DNF *is the following: given a distribution $\mathcal{D}$ on point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, the goal is to find a DNF formula with up to t terms that is consistent with the examples with maximum probability under the distribution $\mathcal{D}$.*

Definition 2.3.4. *For any positive integer t, the problem of* Learn-t-CNF *is the following: given a distribution $\mathcal{D}$ on point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, the goal is to find a CNF formula with up to t literals in each clause that is consistent with the examples with maximum probability under the distribution $\mathcal{D}$.*

The starting point for our inapproximability results for LEARN-t-TERM-DNF and LEARN-t-CNF is the LABEL-COVER problem, which is defined below.

Definition 2.3.5. *An instance $\mathcal{L}$ of* LABEL-COVER(m, k) *consists of a bipartite graph $G(U, V, E)$ and a set of projections $\{\pi_{uv}\}_{(u,v)\in E}$, where $\pi_{uv} : [k] \mapsto [m]$ for every edge $(u, v) \in E$, where $u \in U$ and $v \in V$. A labeling $\sigma_U : U \mapsto [m]$ and $\sigma_V : V \mapsto [k]$ satisfies the edge (u, v), iff $\pi_{uv}(\sigma_V(v)) = \sigma_U(u)$. The goal is to find a labeling that satisfies maximum number of edges of $\mathcal{L}$.*

For notational clarity we shall, frequently in this thesis, drop the parameters (m, k) and refer to the problem simply as LABEL-COVERİt may also refer to variants of the above problem which will be clear from the context. The following theorem is a consequence of the PCP Theorem [8, 6] and Raz's Parallel Repetition Theorem [71].

Theorem 2.3.1. *For any constant $\delta > 0$, there exist m and k such that, given an instance $\mathcal{L}$ of* LABEL-COVER(m, k)*, it is* NP*-hard to distinguish between the following two cases,*

- YES Case*: There is a labeling to the vertices of $\mathcal{L}$ that satisfies all the edges.*
- NO case*: Any labeling to the vertices of $\mathcal{L}$ satisfies at most δ fraction of the edges.*

The following theorem is proved in Section 2.5 and implies Theorem 1.6.2.

Theorem 2.3.2. *For any $\varepsilon > 0$ and any positive integer $t > 0$, there is a polynomial time reduction from an instance $\mathcal{L}$ of* LABEL-COVER(m, k)*, for appropriately chosen m and k, to an instance $\mathcal{I}$ of* LEARN-t-TERM-DNF *such that,*

- YES Case*: If $\mathcal{L}$ is a YES instance then there is a two term DNF ϕ that is consistent with all the examples of $\mathcal{I}$ w.r.t $\mathcal{D}$.*
- NO Case*: If $\mathcal{L}$ is a NO instance then there is no DNF formula ϕ' of up to t terms that is consistent with the examples of $\mathcal{I}$ with probability $\frac{1}{2} + \varepsilon$ w.r.t $\mathcal{D}$.*

The following theorem is proved in Section 2.6 and implies Theorem 1.6.3.

Theorem 2.3.3. *For any $\mu, \varepsilon > 0$ and any positive integer $t > 0$,there is a polynomial time reduction from an instance $\mathcal{L}$ of* Label-Cover(m, k)*, for appropriately chosen m and k, to an instance $\mathcal{I}$ of* Learn-t-CNF *such that,*

- YES Case*: If $\mathcal{L}$ is a YES instance the there is a AND formula that is consistent with all the examples of $\mathcal{I}$ w.r.t $\mathcal{D}$ with probability at least $1 - \mu$.*
- NO Case*: If $\mathcal{L}$ us a NO instance then there is no CNF formula ϕ' with up to t literals in each clause that is consistent with the examples of $\mathcal{I}$ with probability $\frac{1}{2} + \varepsilon$ w.r.t $\mathcal{D}$.*

For the reduction to TT-MinDNF we require a more specialized constraint satisfaction problem which we define below. Let t be a parameter. We define the problem t-Layered-CSP as follows.

Definition 2.3.6. *An instance of* t-Layered-CSP *consists of the following,*

1. *A t-uniform hypergraph $G(V, E)$ which has the following properties,*
   a. *Let V be the vertex set of the hypergraph. Then V can be partitioned into sets $V_1, \ldots, V_t$ such that each edge of the hypergraph has exactly one vertex from each V_i for $i = 1, \ldots, t$. Moreover $|V_1| = |V_2| = \cdots = |V_t|$.*
   b. *Every vertex in V has the same degree.*
2. *A set of labels $[k]$, and constraints for each hyperedge of the graph defined as follows,*
   a. *Let $e = (v_1, v_2, \ldots, v_t)$ be a hyperedge such that $v_i \in V_i$ for all $i = 1, \ldots, t$. Then the constraint C_e is a non empty subset of $[k]^t$.*
   b. *Let $\sigma : V \mapsto [k]$ be a labeling of the vertices in V. Then the hyperedge $e = (v_1, v_2, \ldots, v_t)$, where $v_i \in V_i$ for all $i = 1, \ldots, t$, is satisfied iff $(\sigma(v_1), \ldots, \sigma(v_t)) \in C_e$.*

The goal is to find a labeling $\sigma : V \mapsto [k]$ to the vertices of V that satisfies the maximum number of hyperedges in E.

The following theorem is proved in Section 2.7.

Theorem 2.3.4. *There is an absolute constant $\xi > 0$ such that, for a given arbitrarily large integer $t > 0$, there is a* DTIME($n^{\text{poly}(\log n)}$) *time reduction from* 3SAT *to an instance of* t-Layered-CSP *with $|V| = n$ and $k = \theta(\log^2 n)$ such that,*

YES Case*: If the* 3SAT *formula is satisfiable then there is a set $V' \subset V$ of vertices of size at most $n/(2^{(\log n)^{\xi}})$ and a labeling $\sigma^* : V \setminus V' \mapsto [k]$ such that,*

1. *(Strong Completeness) σ^* satisfies all hyperedges induced by $V \setminus V'$.*
2. *(Extendability) For any hyperedge $e \in E$ (possibly containing vertices from V'), there is an labeling σ'_e to vertices in $e \cap V'$ such that σ^* extended by σ'_e satisfies hyperedge e.*

NO Case*: If the* 3SAT *formula is not satisfiable then any labeling σ to the vertices of V satisfies at most $k^{-t+O(\sqrt{t})}$ fraction of the hyperedges.*

The following theorem is proved in Section 2.4 via a reduction from t-Layered-CSP and it, combined with the result of Feldman [29], implies Theorem 1.6.1.

Theorem 2.3.5. *For any $\varepsilon > 0$, there exists a function $h : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ such that given an instance of* PHC-Cover *consisting of a subset $\mathcal{S}$ of $\{0,1\}^d$ and a set of terms $\mathcal{T}$,unless* NP $\subseteq$ DTIME($n^{\text{poly}(\log n)}$)*, there is no polynomial time algorithm to distinguish between the following two cases,*
YES Case. There is a subset $\mathcal{T}^* \subseteq \mathcal{T}$ of size at most $h(d)$ that covers all the points in $\mathcal{S}$.
NO Case. There is no subset $\mathcal{T}' \subseteq \mathcal{T}$ of size at most $d^{1-\varepsilon}h(d)$ that covers all the points in $\mathcal{S}$.

2.4 Reduction from t-Layered-CSP *to* PHC-Cover

In this section we show a reduction from the problem t-Layered-CSP to PHC-Cover. With the t-Layered-CSP problem as defined in Def 2.3.6, we first construct the set of variables.

Vertex Variables: For every layer V_i $(1 \leq i \leq t)$, we have a set $P^i = \{x_{ij}\}_{1 \leq j \leq D}$ of D variables where $D = \lceil \log |V_i| \rceil$. We refer to them as *vertex variables* for layer i. Clearly we

have a one to one mapping from every vertex $u \in V_i$ to a setting of the variables in P^i for every layer $1 \leq i \leq t$. Call this setting $s^i(u)$. Thus, we have a set of variables for every layer whose settings encode all the vertices of that layer.

Label variables: For every layer V_i $(1 \leq i \leq t)$, we have a set $Q^i = \{y_{ij}\}_{1 \leq j \leq k}$ of k variables each corresponding to a label. We refer to this set as *label variables* for the layer i.

Let $\mathcal{M} = \bigcup_{i=1}^{t}(P^i \cup Q^i)$ be the set of all the variables, and let $d := |\mathcal{M}| = t(D+k)$. We now describe the set of terms $\mathcal{T}$.

Terms: Let V_i $(1 \leq i \leq t)$ be a layer of vertices and let $u \in V_i$. Then there is a set $T^i(u)$ of k terms corresponding to u as follows. Let $t^i(u)$ be the unique AND of the literals corresponding to the variables in $P^i = \{x_{ij}\}_{1 \leq j \leq D}$ such that $t^i(u)$ is 1 only on the setting $s^i(u)$ of the variables in P^i corresponding to u. Let,

$$T^i(u) := \{t^i(u) \wedge \overline{y_{ij}} \ \mid \ 1 \leq j \leq k\}.$$

Therefore, for every layer i $(1 \leq i \leq t)$ and every $u \in V_i$, there is a set of k terms $T^i(u)$. We define,

$$\mathcal{T} = \bigcup_{i=1}^{t} \bigcup_{u \in V_i} T^i(u).$$

In all there are nk terms. Next we define the set of points $\mathcal{S} \subseteq \{0,1\}^{\mathcal{M}}$ for our instance of PHC-Cover.

Points: Let $e = (v_1, v_2, \ldots, v_t)$ be a hyperedge in the graph G, where $v_i \in V_i$ for $1 \leq i \leq t$, and let $C_e \subseteq [k]^t$ be its constraint. Let $I = (I_1, I_2, \ldots, I_t)$, be a t tuple where $I_i \subseteq [k]$ and let $\omega(I) := I_1 \times I_2 \times \cdots \times I_t$. We consider those $I \in (2^{[k]})^t$ such that $\omega(I) \cap C_e = \emptyset$. Note that this is trivially true if any of I_i is empty. In other words, the set $\omega(I)$ does not 'contain' any satisfying assignment to the hyperedge e. Let $\mathcal{I}_e$ be the set of all such t-tuples I corresponding to hyperedge e. Formally,

$$\mathcal{I}_e = \{I \in (2^{[k]})^t \ \mid \ \omega(I) \cap C_e = \emptyset\}.$$

For every such $I \in \mathcal{I}_e$, we create the following point $\gamma_e(I) \in \{0,1\}^{\mathcal{M}}$ as follows. The coordinates corresponding to P^i are set to $s^i(v_i)$ for all $1 \leq i \leq t$. For $1 \leq i \leq t$, the

coordinate corresponding to $y_{ij} \in Q^i$ is set to 1 if $j \in I_i$ and 0 otherwise, for every $1 \leq j \leq k$. We define,

$$\mathcal{S} := \bigcup_{e \in E} \bigcup_{I \in \mathcal{I}_e} \{\gamma_e(I)\}.$$

Now consider any subset $\mathcal{T}^* \subseteq \mathcal{T}$. Let i $(1 \leq i \leq t)$ be a layer, and $u \in V_i$ be a vertex. Define,

$$L^i_{\mathcal{T}^*}(u) := \{j \mid t^i(u) \wedge \overline{y_{ij}} \in \mathcal{T}^*\},$$

for all $1 \leq i \leq t$ and $u \in V_i$. Thus, $L^i_{\mathcal{T}^*}(u)$ is precisely the set of labels of u such that the corresponding terms are present in $\mathcal{T}^*$, where $u \in V_i$. Additionally, for every hyperedge $e = (v_1, v_2, \ldots, v_t) \in E$, let,

$$L_{\mathcal{T}^*}(e) = L^1_{\mathcal{T}^*}(v_1) \times \cdots \times L^t_{\mathcal{T}^*}(v_t).$$

The following is a simple lemma.

Lemma 2.4.1. *Let $\mathcal{T}^* \subseteq \mathcal{T}$. Then $\mathcal{T}^*$ covers all the points in $\mathcal{S}$ if and only if for every hyperedge $e = (v_1, v_2, \ldots, v_t) \in E$, where $v_i \in V_i$ for $1 \leq i \leq t$, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.*

Proof: Let us fix a hyperedge $e = (v_1, \ldots, v_t)$ where $v_i \in V_i$ for all $1 \leq i \leq t$. Consider any point $\gamma_e(I)$ for $I \in \mathcal{I}_e$. First we show that $\gamma_e(I)$ can be covered by terms only from the sets $T^i(v_i)$ for $1 \leq i \leq t$. Let u be any vertex such that $u \neq v_i$ for $1 \leq i \leq t$. Assume that $u \in V_{i'}$ for some $1 \leq i' \leq t$. By the construction of $\gamma_e(I)$, the coordinates corresponding to $P^{i'}$ are set to $s^{i'}(v_{i'})$, and the AND formula $t^{i'}(u)$ is 0 on this setting since $v_{i'} \neq u$.

Since, for any point $\gamma_e(I)$, the variables P^i are set to $s^i(v_i)$ for all $1 \leq i \leq t$, all the AND formulas, $t^i(v_i)$ are set to 1. Therefore, $\gamma_e(I)$ is not covered by $\mathcal{T}^*$ if and only if, for all layers i $(1 \leq i \leq t)$, the coordinates corresponding to $\{y_{ij} \mid j \in L^i_{\mathcal{T}^*}(v_i)\}$ are set to 1. Equivalently, $\omega(I) \supseteq L^1_{\mathcal{T}^*}(v_1) \times \cdots \times L^t_{\mathcal{T}^*}(v_t) = L_{\mathcal{T}^*}(e)$. By the definition of $\mathcal{I}_e$, $\omega(I) \cap C_e = \emptyset$. Therefore, if there is an $I \in \mathcal{I}_e$ such that $\gamma_e(I)$ is not covered by $\mathcal{T}^*$, then $L_{\mathcal{T}^*}(e) \cap C_e = \emptyset$. For the reverse direction, we note that if $L_{\mathcal{T}^*}(e) \cap C_e = \emptyset$, then we can set $I = (L^1_{\mathcal{T}^*}(v_1), \ldots, L^t_{\mathcal{T}^*}(v_t))$, and $\gamma_e(I)$ is not covered by $\mathcal{T}^*$. This completes the proof. ∎

2.4.1 Analysis

To prove our hardness of approximation result for PHC-Cover we reduce from the t-Layered-CSP instance obtained from the Theorem 2.3.4 to an instance of PHC-Cover via the reduction described above. Next we present the analysis of the YES and NO cases.

2.4.1.1 YES Case. In the YES case we have a set of vertices $V' \subseteq V$ of size at most $n/2^{(\log n)^{\xi}}$, and a labeling σ^* to the vertices $V \setminus V'$ satisfying the properties in Theorem 2.3.4. Now we construct a set of terms $\mathcal{T}^* \subseteq \mathcal{T}$ as follows. For every layer i, $(1 \leq i \leq t)$, do the following. For every vertex $u \in V_i$, if $u \in V'$ then $\mathcal{T}^*$ contains the k terms in the set $T^i(u)$ corresponding to u. Otherwise, if $u \notin V'$, then $\mathcal{T}^*$ contains only the term $t^i(u) \wedge \overline{y_{i\sigma^*(u)}}$, i.e. the term in $T^i(u)$ corresponding to the label of u given by σ^*.

We show that $\mathcal{T}^*$ covers all the points in $\mathcal{S}$. Let $e = (v_1, \ldots, v_t)$ be a hyperedge, where $v_i \in V_i$ for $1 \leq i \leq t$. We have two cases.

Case 1. e is induced by $V \setminus V'$. The labeling σ satisfies e. Then $L^i_{\mathcal{T}^*}(v_i) = \{\sigma^*(v_i)\}$, for $1 \leq i \leq t$ and $L_{\mathcal{T}^*}(e) = \{(\sigma^*(v_1), \ldots, \sigma^*(v_t))\}$. And therefore $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.

Case 2. e contains vertices from V'. Then, $L^i_{\mathcal{T}^*}(v_i) = \{\sigma^*(v_i)\}$ if $v_i \in V \setminus V'$ and $L^i_{\mathcal{T}^*}(v_i) = \{1, 2, \ldots, k\}$ otherwise. Now, by the Extendability property in Theorem 2.3.4, there is a labeling σ'_e to vertices in $e \cap V'$ such that σ^* extended by σ'_e satisfies e. Clearly, this implies there is a labeling to the vertices v_i in e from the sets $L^i_{\mathcal{T}^*}(v_i)$ for $1 \leq i \leq t$ that satisfies the hyperedge e. Therefore, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.

Therefore, for every edge e, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$. And by Lemma 2.4.1 the set of terms $\mathcal{T}^*$ covers all the points in $\mathcal{S}$. The number of terms in $\mathcal{T}^*$ is,

$$\begin{aligned} &|V \setminus V'| + k|V'| \\ &\leq n\left(1 - \frac{1}{2^{(\log n)^{\xi}}}\right) + k\left(\frac{n}{2^{(\log n)^{\xi}}}\right). \end{aligned}$$

Since, we have $k = \theta(\log^2 n)$, the above expression is at most $2n$ for large enough n. Therefore, the number of terms in $\mathcal{T}^*$ is at most $2n$.

2.4.1.2 NO Case. Suppose that there is a set of terms $\mathcal{T}' \subseteq \mathcal{T}$ that covers all the points in $\mathcal{S}$. By Lemma 2.4.1, for every hyperedge e, $L_{\mathcal{T}'}(e) \cap C_e \neq \emptyset$. Now, consider the

labeling σ' constructed in a randomized manner as follows. Let u be a vertex in, say, V_i for some $1 \leq i \leq t$. Select $\sigma'(u)$ to be a random label from $L^i_{\mathcal{T}'}(u)$. Suppose $e = (v_1, \ldots, v_t)$ is a hyperedge where $v_i \in V_i$ for $1 \leq i \leq t$. Since $L_{\mathcal{T}'}(e)$ contains a satisfying assignment from C_e we have the following,

$$\Pr\left[e \text{ is satisfied by } \sigma'\right] \geq \frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}.$$

Therefore, the expected fraction of edges satisfied is at least,

$$\mathrm{E}_{\sigma'}\left[\text{Fraction of edges satisfied by } \sigma'\right] \geq \mathrm{E}_{e=(v_1,\ldots,v_t)}\left[\frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}\right]. \tag{1}$$

The left hand side of the above expression is less than the soundness $\delta = k^{-t+O(\sqrt{t})}$ of the NO case. Therefore,

$$\mathrm{E}_{e=(v_1,\ldots,v_t)}\left[\frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}\right] \leq \delta.$$

Therefore, for at least $\frac{1}{2}$ fraction of the hyperedges $e = (v_1, \ldots, v_t)$ we have,

$$\begin{aligned}
& \frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|} \leq 2\delta \\
\Rightarrow \quad & \prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)| \geq \frac{1}{2\delta} \\
\Rightarrow \quad & \frac{\sum_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}{t} \geq \frac{1}{(2\delta)^{\frac{1}{t}}}.
\end{aligned} \tag{2}$$

Now, since each vertex in V has the same degree,

$$\begin{aligned}
|\mathcal{T}'| &= \sum_{i=1}^{t} \sum_{u \in V_i} L^i_{\mathcal{T}'}(u) \\
&= n\mathrm{E}_{e=(v_1,\ldots,v_t)}\left[\frac{\sum_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}{t}\right].
\end{aligned} \tag{3}$$

And combining Equations (2) and (3), we have,

$$|\mathcal{T}'| \geq n\left(\frac{1}{2}\right)\left(\frac{1}{(2\delta)^{\frac{1}{t}}}\right)$$

Substituting the value of δ, we obtain that,

$$|\mathcal{T}'| \geq \frac{nk^{1-O\left(\frac{1}{\sqrt{t}}\right)}}{2^{1+\frac{1}{t}}}.$$

Since t can be made to be an arbitrarily large constant, combining the above with the analysis of the YES case, we get a gap of $k^{1-\varepsilon}$ for the optimum of the instance of PHC-Cover, for any constant $\varepsilon > 0$. Also, the number of variables d is at most $t(\log n + k) = O(k)$, since $k = \theta(\log^2 n)$. In terms of d, we obtain a gap of $d^{1-\varepsilon}$. Clearly the reduction runs in time $O(2^d)$, which $2^{O(k)} = O(2^{\log^3 n})$. Therefore, along with the inapproximability of t-Layered-CSP given in Theorem 2.3.4, this proves Theorem 2.3.5.

2.5 Hardness of Learning 2-clause CNF by t-clause CNF

In this section we prove Theorem 2.3.2 which implies Theorem 1.6.2. For convenience we shall prove an equivalent result for learning 2-clause CNF by t-clause CNF.

We start with an instance $\mathcal{L}$ of Label-Cover(m, k) consisting of a bipartite graph $G(U, V, E)$, set of labels $[m]$ (for vertices in U), $[k]$ (for vertices in V), the projections $\pi_{uv} : [k] \mapsto [m]$ for every edge $e = (u, v) \in E$, where vertices in U have degree d_U, and those in V have degree d_V. All the parameters are constants independent of the sizes $N_U = |U|$ and $N_V = V$. Let $N = N_U + N_V$.

2.5.1 Reduction

Variables. First we define the set of variables. Let v be any vertex in V. We have the set of variables $S_v = \{x_i^v\}_{i=1}^k$. Similarly, let u be any vertex in U, and let $S_u = \{y_i^u\}_{i=1}^m$. Thus, we have one variable for every vertex and every potential label for that vertex. Let,

$$\mathcal{S} = (\cup_{u \in U} S_u) \bigcup (\cup_{v \in V} S_v)$$

be the set of all variables. Let the corresponding boolean hypercube be $\{0, 1\}^{\mathcal{S}}$ where the coordinates are indexed by the variables in $\mathcal{S}$.

Distribution. We now describe how the oracle generates a sample point. This describes the distribution $\mathcal{D}$ on the samples. Let $\mu \in (0, 1)$ be a 'perturbation' parameter, which we will fix later. On being queried for a sample, the oracle does the following,

1. Chooses a vertex $u \in U$ at random from the vertices in U. Let $N(u) \subseteq V$ be the neighborhood of u.

2. With probability $\frac{1}{2}$ does the following,

   2a. Picks $v \in N(u)$ at random.

   2b. Creates the following point $Z_1^{uv} \in \{0,1\}^{\mathcal{S}}$ as follows,

$$\forall j \in [k], \qquad Z_1^{uv}(x_j^{v'}) = \begin{cases} 1 & \text{if } v' = v \\ 0 & \text{otherwise} \end{cases}$$

   and,

$$\forall j \in [m], \qquad Z_1^{uv}(y_j^{u'}) = \begin{cases} 1 & \text{if } u' = u \\ 0 & \text{otherwise} \end{cases}$$

   2.c Output the sample $(Z_1^{uv}, 1)$.

3. With probability $\frac{1}{2}$ does the following,

   3.a Chooses a set $\alpha \subseteq [m]$ by picking every $i \in [m]$ independently with probability μ.

   3.b Creates the following point $Z_0^{u\alpha} \in \{0,1\}^{\mathcal{S}}$ as follows,

$$\forall j \in [k], \qquad Z_0^{u\alpha}(x_j^{v'}) = \begin{cases} 0 & \text{if } v' \notin N(u) \\ 1 & \text{if } v' \in N(u) \text{ and } \pi_{uv'}(j) \in \alpha \\ 0 & \text{if } v' \in N(u) \text{ and } \pi_{uv'}(j) \notin \alpha \end{cases}$$

   and,

$$\forall j \in [m], \qquad Z_0^{u\alpha}(y_j^{u'}) = \begin{cases} 0 & \text{if } u' \neq u \\ 0 & \text{if } u' = u \text{ and } j \in \alpha \\ 1 & \text{if } u' = u \text{ and } j \notin \alpha \end{cases}$$

   3.c Output the sample $(Z_0^{u\alpha}, 0)$.

We note that the distribution has a polynomial (in $|\mathcal{S}|$) support, and therefore can be given explicitly.

Let $t > 0$ be a given integer and $\varepsilon > 0$ be a given parameter. We will show that if $\mathcal{L}$ is a YES instance of Label-Cover(m, k), i.e if there is a labeling that satisfies all edges

then there is a 2-clause CNF which is consistent with all the samples. On the other hand, if $\mathcal{L}$ is a NO instance then there is no t clause CNF that is consistent with the samples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$ provided the soundness η of the $\mathcal{L}$ is chosen to be suitably small.

2.5.2 Analysis

We present the analysis of the YES and the NO cases.

2.5.2.1 YES Case. Let $\mathcal{L}$ be a YES instance of Label-Cover(m, k). Then there is a labeling σ to the vertices of G that satisfies all the edges. Consider the following two clauses,

$$C_V = \bigvee_{v \in V} x^v_{\sigma(v)} \ ,$$

and,

$$C_U = \bigvee_{u \in U} y^u_{\sigma(u)}.$$

Let $\phi = C_V \wedge C_U$. We will show that ϕ is consistent with all the data points.

Consider any data point of the form $(Z_1^{uv}, 1)$ where $Z_1^{uv} \in \{0,1\}^{\mathcal{S}}$. Recall that Z_1^{uv} was generated by picking a vertex $u \in U$ then a vertex $v \in N(u)$. By the construction of Z_1^{uv}, clearly $Z_1^{uv}(x^v_{\sigma(v)}) = 1$ and $Z_1^{uv}(y^u_{\sigma(u)}) = 1$. Therefore, the clauses C_V and C_U, both are 1 on the point Z_1^{uv}, and therefore ϕ is also 1 at the point Z_1^{uv}. So the formula ϕ is consistent with all the data points of the form $(Z_1^{uv}, 1)$.

Now consider any data point $(Z_0^{u\alpha}, 0)$. Recall that $Z_0^{u\alpha}$ was constructed by first picking a vertex $u \in U$ and then a set $\alpha \subseteq [m]$. We consider two cases.

Case 1. Let $\sigma(u) \in \alpha$. We observe that in this case $Z_0^{u\alpha}(y^u_{\sigma(u)}) = 0$, and further, for all $u' \in U$, $Z_0^{u\alpha}(y^{u'}_{\sigma(u')}) = 0$. And so C_U evaluates to 0 on $Z_0^{u\alpha}$, and therefore ϕ evaluates to 0 on $Z_0^{u\alpha}$.

Case 2. Let $\sigma(u) \notin \alpha$. Then $\forall v \in N(u)$, $\pi_{uv}(\sigma(v)) = \sigma(u) \notin \alpha$ by construction of the point $Z_0^{u\alpha}$. Therefore, for all $v \in N(u)$, $Z_0^{u\alpha}(x^v_{\sigma(v)}) = 0$, and moreover for all $v' \in V \setminus N(u)$, $Z_0^{u\alpha}(x^{v'}_{\sigma(v')}) = 0$. Therefore, C_V evaluates to 0 on $Z_0^{u\alpha}$ and therefore ϕ evaluates to 0 on $Z_0^{u\alpha}$.

Therefore, ϕ is consistent with all the data points of the form $(Z_0^{u\alpha}, 0)$.

From the above analysis we conclude that the 2-clause CNF formula ϕ is consistent with all the data points of $\mathcal{D}$.

2.5.2.2 NO Case. For the sake of contradiction we assume that there is a t clause CNF formula ϕ^* which is consistent with the data points with probability at least $\frac{1}{2}+\varepsilon$ for some given constants $\varepsilon, t > 0$. We will set the perturbation parameter $\mu = \frac{\varepsilon^2}{16t^3}$.

Let the given t clause CNF formula be $\phi^* = C_1 \wedge \cdots \wedge C_t$. We will first show that not all the clauses $C_1, \ldots, C_t$ can contain a negative literal.

From the construction of the data points it is easy to see that any given coordinate of $\{0,1\}^{\mathcal{S}}$ is set to 1 with probability at most $\frac{d_U+d_V}{\min\{|U|,|V|\}} = \xi(N) = o(1)$. Therefore, if all the clauses in ϕ^* had a negative literal, then ϕ^* would evaluate to 1 with probability at least $1 - t\xi(N) = 1 - o(1)$ over the distribution $\mathcal{D}$, which is a contradiction to the assumption that ϕ^* is consistent with the data points with probability at least $\frac{1}{2}+\varepsilon$ for constant $\varepsilon > 0$, since the 0 and 1 data points are equally likely in $\mathcal{D}$. This implies that there is a non empty subset Q of clauses of ϕ^*, such that none of the clauses in Q contains a negative literal. W.l.o.g. we may assume that $Q = \{C_1, \ldots, C_\ell\}$, where $\ell \leq t$. Moreover, the formula $\phi = C_1 \wedge \cdots \wedge C_\ell$ must be consistent with the data points of the oracle with probability at least $\frac{1}{2} + \varepsilon - t\xi(N) \geq \frac{1}{2} + \varepsilon/2$, for large enough size of instance. For the remainder of the argument we shall only consider the CNF ϕ and use it to construct a 'good' labeling to the vertices of $\mathcal{L}$.

Before proceeding we first define ℓ distinguished labels from $[k] \cup \{0\} : \{q_i^v\}_{i=1}^{\ell}$ for each $v \in V$. Let q_i^v be any arbitrary label $j \in [k]$ such that the positive literal x_j^v is present in clause C_i of ϕ, and 0 if there is no such variable in C_i. We call this setting of distinguished labels Γ.

Since ϕ is consistent with the data points of the oracle with probability at least $\frac{1}{2} + \frac{\varepsilon}{2}$, by an averaging argument we have that there is a set $U' \subseteq U$ such that $|U'| \geq \frac{\varepsilon}{4}|U|$, such that for every vertex $u \in U'$, ϕ is consistent with probability at least $\frac{1}{2} + \frac{\varepsilon}{4}$ with the data points generated by the oracle on picking u in step 1. Call such vertices $u \in U'$ as 'good'.

Analysis for a fixed 'good' vertex $u \in U'$. We now fix one such 'good' vertex u. The rest of the analysis is with respect to this 'good' vertex. Let $N(u)$ be its neighborhood. After picking u in step 1, the oracle outputs a 0 example and a 1 example with equal probability. Therefore, again by averaging, it must be the case that ϕ is consistent with the 1 examples (of u) with probability (over choice of $v \in N(u)$ in step 2a) at least $\frac{\varepsilon}{4}$; and consistent with the 0 examples (of u) with probability (over the choice of the set α in step 3a) at least $\frac{\varepsilon}{4}$.

Suppose C_i is a clause in ϕ for some $1 \leq i \leq \ell$, such that C_i contains a positive literal y_j^u for some $j \in [m]$. Then, C_i will be 0 with probability at most μ on the 0 examples of u. Therefore, by union bound, the probability that any of the clauses of ϕ containing a positive literal evaluates to 0 on the 0 examples is at most $t\mu$, which is at most $\frac{\varepsilon}{8}$ for our setting of the parameter μ. Therefore, there is a subformula ϕ_u of ϕ containing the clauses $\{C_i\}_{i \in L_u}$, where $L_u \subseteq [\ell]$, such that none of the clauses of ϕ_u contains a variable of the form y_j^u for $j \in [m]$, and moreover ϕ_u is consistent with the 0 examples with probability at least $\varepsilon' = \frac{\varepsilon}{4} - t\mu \geq \frac{\varepsilon}{8}$, and with the 1 examples also with probability at least ε'. The rest of the analysis will show that there is an appropriate clause in ϕ_u which gives a good labeling for a significant fraction of the vertices in $N(u)$.

Since ϕ_u is consistent with the 1 examples of u, with probability ε', there must be a set $M(u) \subseteq N(u)$ such that $|M(u)| \geq \varepsilon' |N(u)|$ and for every $v \in M(u)$, ϕ_u is 1 on the point Z_1^{uv} constructed on choosing v in step 2a. Call such vertices 'good neighbors' of u. We have shown that ϕ_u does not contain any negative literal, or any positive literal of the form y_j^u for any $j \in [m]$, in any of its clauses. From the construction of the point Z_1^{uv}, this implies that every clause C_i ($i \in L_u$) contains a positive literal from the set $\{x_j^v\}_{j=1}^k$, for all 'good neighbors' v of u. So the setting q_i^v given by Γ, of distinguished labels for the 'good neighbors' v corresponding to the clauses C_i of ϕ_u is not 0.

We also have that with probability ε' over the sets α chosen in step 3a, ϕ_u is 0 on the points $Z_0^{u\alpha}$. This implies that there is a clause C_{i_u} of ϕ_u, for some $i_u \in L_u$, such that C_{i_u} is 0 on the points $Z_0^{u\alpha}$ with probability at least $\frac{\varepsilon'}{\ell}$. We have,

$$\Pr_{\alpha}[C_{i_u} \text{ is 0 on } Z_0^{u\alpha}] \geq \frac{\varepsilon'}{\ell}. \tag{4}$$

Now, since C_{i_u} is a clause of ϕ_u, it contains positive literals corresponding to all the 'good neighbors' $v \in M(u)$, and therefore $q_{i_u}^v \in [k]$ for all $v \in M(u)$. Define the set $T_u \subseteq [m]$ as,

$$T_u = \{\pi_{uv}(q_{i_u}^v) \mid v \in M(u)\}.$$

In other words, T_u is the subset of $[m]$ onto which the distinguished labels of the vertices $v \in M(u)$ corresponding to the clause C_{i_u} project. From the construction of the points $Z_0^{u\alpha}$, we have the following observation.

Observation 2.5.1. *If $\alpha \cap T_u \neq \emptyset$ then C_{i_u} is 1 on the point $Z_0^{u\alpha}$.*

We will show that the above observation implies that the set T_u cannot be too large. We have,

$$\begin{aligned} \Pr_{\alpha}[\alpha \cap T_u = \emptyset] &= (1-\mu)^{|T_u|} \\ &\geq \Pr_{\alpha}[C_{i_u} \text{ is 0 on } Z_0^{u\alpha}] \end{aligned}$$

and combining the above with Equation (4), we have,

$$\begin{aligned} (1-\mu)^{|T_u|} &\geq \frac{\varepsilon'}{\ell} \\ &\geq \frac{\varepsilon'}{t}. \end{aligned}$$

Therefore,

$$|T_u| \leq \frac{1}{\mu} \ln\left(\frac{t}{\varepsilon'}\right).$$

For convenience let $\nu = \left(\frac{1}{\mu}\ln\left(\frac{t}{\varepsilon'}\right)\right)^{-1}$. Define,

$$\Lambda_j^u = \{v \in M(u) \mid \pi_{uv}(q_{i_u}^v) = j\}$$

for all $j \in [m]$. Essentially, Λ_j^u is the subset of the 'good' neighbors v of u whose distinguished label corresponding to the clause C_{i_u} projects onto j. We have the following simple lemma.

Lemma 2.5.2. *$\exists j_u \in T_u$ such that $|\Lambda_{j_u}^u| \geq \nu |M(u)|$.*

Proof: Note that $M(u) = \bigcup_{j \in T_u} \Lambda_j^u$. And since $|T_u| \leq \frac{1}{\nu}$, the lemma follows. ∎

Labeling. We now define the labeling. The partial labeling $\sigma_V : V \mapsto [k]$ is constructed in a randomized manner as follows. For every vertex $v \in V$, choose i_v randomly from $\{1, \ldots, \ell\}$. If $q^v_{i_v} \in [k]$ then set $\sigma(v) = q^v_{i_v}$. Essentially, for every vertex v, we label it by its distinguished label (given by the setting Γ) corresponding to a random clause of ϕ (if the label is not 0).

We construct the partial labeling $\sigma_U : U \mapsto [m]$ as follows. For every 'good' vertex $u \in U'$, let $\sigma(u) = j_u$ as in Lemma 2.5.2.

Now we analyze how many edges are satisfied by the partial assignment σ_V, σ_U. Let (u, v) be a random edge chosen by picking u randomly from U and then choosing v randomly from V. With probability $\frac{\varepsilon}{4}$, u is a good vertex. With probability at least $\varepsilon'\nu$, the vertex v is selected from $\Lambda^u_{j_u}$, and with a further probability at least $\frac{1}{\ell} \geq \frac{1}{t}$, the vertex v is labeled with the label $q^v_{i_u}$ which projects onto j_u via the map π_{uv}. Therefore, the edge is satisfied with probability at least

$$\begin{aligned} p^* &= \left(\frac{\varepsilon}{4}\right)\varepsilon'\nu\left(\frac{1}{t}\right) \\ &\geq \left(\frac{\varepsilon}{4}\right)\left(\frac{\varepsilon}{8}\right)\nu\left(\frac{1}{t}\right) \end{aligned}$$

which, by the definition of ν and our choice of μ, is a constant depending only on ε and t. Since a random edge is satisfied with probability p^*, the expected fraction of edges satisfied is p^*. This implies that there must be a labeling that satisfies at least p^* fraction of the edges. The soundness η of $\mathcal{L}$ can be chosen arbitrarily small to obtain a contradiction.

2.6 Hardness of Learning OR by t-DNF under adversarial noise

In this section we prove Theorem 2.3 which implies Theorem 1.6.3. For convenience, we shall prove an equivalent result for learning OR by a t-DNF under adversarial noise.

We start with an instance $\mathcal{L}$ of Label-Cover(m, k) consisting of a bipartite graph $G(U, V, E)$, set of labels $[m]$ (for vertices in U), $[k]$ (for vertices in V), the projections $\pi_{uv} : [k] \mapsto [m]$ for every edge $e = (u, v) \in E$, where vertices in U have degree d_U, and those in V have degree d_V. All the parameters are constants independent of the sizes $N_U = |U|$ and $N_V = |V|$. Let $N = N_U + N_V$. Let the soundness parameter be η.

2.6.1 Reduction

Variables. First we define the set of variables. Let v be any vertex in V. We have a set k variables, $S_v = \{x_i^v\}_{i=1}^k$ for every vertex $v \in V$, with one variable for every (potential) label for that vertex. Let,

$$\mathcal{S} = \bigcup_{v \in V} S_v$$

be the set of all variables. Let the corresponding boolean hypercube be $\{0,1\}^{\mathcal{S}}$ where the coordinates are indexed by the variables in $\mathcal{S}$.

Distribution. We now describe how the oracle generates a sample point. This describes the distribution $\mathcal{D}$ on the samples. Let $\mu \in (0,1)$ be a given parameter. Let $\ell > 0$ be a positive integer to be fixed later. On being queried for a sample, the oracle does the following,

1. Chooses a vertex $u \in U$ at random from the vertices in U. Let $N(u) \subseteq V$ be the neighborhood of u.

2. With probability $\frac{1}{2}$ does the following,

   2a. Picks $v \in N(u)$ at random.

   2b. Creates the following point $Z_1^u[v] \in \{0,1\}^{\mathcal{S}}$ as follows,

$$\forall j \in [k], \qquad Z_1^u[v](x_j^{v'}) = \begin{cases} 1 & \text{if } v' = v \\ 0 & \text{otherwise} \end{cases}$$

   2c. Output $(Z_1^u[v], 1)$ as a data point.

3. With probability $\frac{1}{2}$ does the following,

   3a. Picks a ℓ tuple $(v_1, \ldots, v_\ell)$ such that each v_i is chosen uniformly at random from $N(u)$ for $1 \leq i \leq \ell$.

   3b. Picks a set $\alpha \subseteq [m]$ by picking every element of $[m]$ independently at random with probability μ.

3c. Creates the following point $Z_0^u[\alpha, (v_1, \ldots, v_\ell)]$ as follows,

$$\forall j \in [k], \qquad Z_0^u[\alpha, (v_1, \ldots, v_\ell)](x_j^{v'}) = \begin{cases} 1 & \text{if } \exists i \in [\ell] \text{ s.t } v' = v_i \ \& \ \pi_{uv'}(j) \in \alpha \\ 0 & \text{otherwise} \end{cases}$$

3d. Outputs $(Z_0^u[\alpha, (v_1, \ldots, v_\ell)], 0)$ as a data point.

Note that the support of $\mathcal{D}$ is polynomial in the size of the Label-Cover instance $\mathcal{L}$ and hence $\mathcal{D}$ can be given explicitly. Let $t > 0$ be a given positive integer and $\varepsilon, \mu > 0$ be given parameters that may be arbitrarily small constants. We will show that if $\mathcal{L}$ is a YES instance, i.e there is a labeling that satisfies all edges then there is a OR formula which is consistent with the samples with probability $1 - \mu$. On the other hand, if $\mathcal{L}$ is a NO instance then there is no t-DNF formula that is consistent with the samples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$ with the soundness η and the degrees d_U and d_V suitably chosen.

2.6.2 Analysis

We present the analysis of the YES and the NO cases.

2.6.2.1 YES Case. Suppose the instance $\mathcal{L}$ of Label-Cover is a YES instance. In this case, there is a labeling σ to the vertices of $\mathcal{L}$ that satisfies all the edges. Consider the following OR formula,

$$\phi = \bigvee_{v \in V} x^v_{\sigma(v)}. \tag{5}$$

The formula ϕ contains one positive literal for every vertex $v \in V$, corresponding to the label assigned to v by σ. Clearly, ϕ is consistent with any 1 example $(Z_1^u[v], 1)$ generated by the oracle. This is because in $Z_1^u[v]$, the coordinates corresponding to all the labels of v are set to 1, and therefore the literal $x^v_{\sigma(v)}$ is 1 on $Z_1^u[v]$.

Now suppose the oracle selects a vertex $u \in U$ and then generates a 0 example $(Z_0^u[\alpha, (v_1, \ldots, v_\ell)], 0)$. In the point $Z_0^u[\alpha, (v_1, \ldots, v_\ell)]$ all the variables $x^v_{\sigma(v)}$ are set to 0 where $v \neq v_i$ for all $1 \leq i \leq \ell$. Now suppose $\sigma(u) \notin \alpha$. Then $x^{v_i}_{\sigma(v_i)}$ is set to 0 for all $1 \leq i \leq \ell$. Therefore, ϕ evaluates to 0 in this case. Now the probability that $\sigma(u) \notin \alpha$ is

exactly $1-\mu$, by the construction of the set α. Therefore, with probability at least $1-\mu$, ϕ is consistent with the 0 examples of u.

The above analysis holds for any vertex u as the choice in step 1. Therefore, overall, ϕ is consistent with the data points of the verifier with probability at least $1-\mu$.

2.6.2.2 NO Case. Suppose that $\mathcal{L}$ is a NO instance, i.e. no labeling to the vertices of $\mathcal{L}$ satisfies η fraction of the edges, where η is the soundness parameter which will be chosen to be small enough later. We assume that there is a t-DNF formula ϕ^* that is consistent with the examples of the oracle with probability at least $\frac{1}{2}+\varepsilon$, under the distribution $\mathcal{D}$. We have that,

$$\phi^* = \bigvee_{j=1}^{M} T_j \tag{6}$$

for some M, and each term T_j is the AND of at most t literals. Suppose there is a term T' of ϕ^* such that it is an AND of only negative literals. Now such a term will be 1 with probability at least $1-\frac{td_U}{|V|}$, which would imply that ϕ^* is 1 with probability at least $1-\frac{td_U}{|V|}$. Since the oracle outputs 0 and 1 examples equally often, this is a contradiction to the assumption that ϕ^* is consistent with the examples of the oracle with probability at least $\frac{1}{2}+\varepsilon$ for large enough $|V|$. Therefore, we may assume that every term of ϕ^* has at least one positive literal. We now make this simple observation.

Observation 2.6.1. *If a given term T_j is never 1 on any of the 1 examples of the oracle, then $\phi \setminus \{T_j\}$, is also consistent with the examples of the oracle with probability $\frac{1}{2}+\varepsilon$.*

This is because removing a term can hurt us only in the case of 1 examples, so we can remove all the terms that are never 1 on the 1 examples. This leads to the following simple lemma.

Lemma 2.6.2. *Let ϕ be the OR of all the terms T_j of ϕ^* such that for each T_j there is a vertex $v \in V$, such that all the positive literals in the term T_j are of the form x_i^v for some $1 \le i \le k$, and T_j does not contain any negative literal of the form $\bar{x}_{i'}^v$ for any $1 \le i' \le k$. Then ϕ is also consistent with the examples of the oracle with probability $\frac{1}{2}+\varepsilon$.*

Proof: Suppose there is a term T_j of ϕ such that it contains positive literals of the form $x_{i_1}^{v_1}$ and $x_{i_2}^{v_2}$, where $v_1 \neq v_2$ and $1 \leq i_1, i_2, \leq k$. Since all the 1 data points of the oracle have the property that all the coordinates that are set to 1 correspond to the variables x_i^v $1 \leq i \leq k$, for exactly one such vertex $v \in V$, the term T_j will be 0 on all such points as it contains positive literals corresponding to two different vertices.

Moreover, if T_j contains a positive literal of the form $x_{i_1}^v$ and a negative literal of the form $\bar{x}_{i_2}^v$, then again T_j will always be 0 on the 1 examples since in the 1 data points, for any vertex $v' \in V$, either all the coordinates corresponding to $\{x_i^{v'}\}_{i=1}^k$ are set to 1 or all of them are set to 0. Therefore, removing T_j does not hurt us in the 1 examples and clearly $\phi^* \setminus \{T_j\}$ is as good as ϕ^* on the 0 examples. Therefore, we can remove all such terms and obtain the t-DNF formula ϕ which is also consistent with the examples of the oracle with probability at least $\frac{1}{2} + \varepsilon$. ∎

In the rest of the analysis we will use the formula ϕ to construct a good labeling for the vertices of $\mathcal{L}$.

Before proceeding, we will construct the following assignment of terms to vertices. For every vertex $v \in V$, let $T^v = T_{j'}$ be any arbitrary term of ϕ containing at least one positive literal of the form x_i^v. If no such term exists for v in ϕ let $T^v = 0$. Call this assignment Γ. Clearly Γ is well defined since, every term has at least one positive literal of the form x_i^v for exactly one $v \in V$. For every vertex $v \in V$, let us also define the set $W(v) := \{i \in [k] \mid x_i^v \text{ is a positive literal of } T^v\}$. As mentioned, all the positive literals of T^v are necessarily of the form x_i^v for $1 \leq i \leq k$. Therefore, unless $T^v = 0$ the set $W(v)$ is non empty. Rest of the analysis will be with respect to this assignment Γ.

Since ϕ is consistent with the data points of the oracle with probability at least $\frac{1}{2} + \varepsilon$, by an averaging argument we have that there is a set $U' \subseteq U$ such that $|U'| \geq \frac{\varepsilon}{2}|U|$, such that for every vertex $u \in U'$, ϕ is consistent with probability at least $\frac{1}{2} + \frac{\varepsilon}{2}$ with the data points generated by the oracle on picking u in step 1. Call such vertices $u \in U'$ as 'good'. We fix one such 'good' vertex u and do the analysis for the 0 and 1 examples output by the oracle after choosing u in the initial step.

Analysis for a fixed 'good' vertex $u \in U'$. Let $N(u)$ be the neighborhood of u. After picking u in step 1, the oracle outputs a 0 example and a 1 example with equal probability. Therefore, again by averaging, it must be the case that ϕ is consistent with the 1 examples (of u) with probability (over choice of $v \in N(u)$ in step 2a) at least $\frac{\varepsilon}{2}$; and consistent with the 0 examples (of u) with probability (over the choice of the set α, and the ℓ tuple $(v_1, \ldots, v_\ell)$ in step 3a and 3b) at least $\frac{\varepsilon}{2}$. For convenience, let $\varepsilon' = \frac{\varepsilon}{2}$.

Since ϕ is consistent with the 1 examples with probability at least ε', this implies that ϕ is 1 on the points $Z_1^u[v]$ for at least ε' fraction of the neighbors $v \in N(u)$. Let the set of such vertices v be $M(u)$, where $|M(u)| \geq \varepsilon'|N(u)|$, and call such v as 'good neighbors' of u. We have shown that ϕ does not have any term with all negative literals, and every term of ϕ must contain positive literals, all of them from exactly one vertex of V. Since the only coordinates of $Z_1^u[v]$ that are set to 1 correspond to x_i^v for $1 \leq i \leq k$, it must be that for every 'good neighbor' v, there is a term of ϕ containing positive literals only of the form x_i^v for some $1 \leq i \leq k$. This implies that for such vertices v, $T^v \neq 0$ and $W(v) \neq \emptyset$ in our setting Γ.

Consider an ℓ-tuple $\bar{v} = (v_1, \ldots, v_\ell)$ chosen randomly by choosing every v_i uniformly at random from $N(u)$. Let $D_{\bar{v}} := \{r \in [\ell] \mid v_r \in M(u)\}$. Essentially, $D_{\bar{v}}$ is the set of indices r such that v_r is a 'good' vertex. We call $\bar{v}$ as 'dense' if $|D_{\bar{v}}| \geq \frac{\varepsilon'}{2}\ell$. Since each coordinate of $\bar{v}$ is chosen uniformly at random from $N(u)$ and $|M(u)| \geq \varepsilon'|N(u)|$, we expect $\bar{v}$ to be 'dense' with high probability. Indeed, using the Chernoff bound, we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is not dense }] \leq \exp(-\varepsilon'\ell/8).$$

Consider the ordered pair (r_1, r_2) such that $1 \leq r_1 \neq r_2 \leq \ell$. Call such a pair intersecting for an ℓ-tuple $\bar{v}$ if $T^{v_{r_1}}$ contains a literal of the form $\bar{x}_i^{v_{r_2}}$ for some $1 \leq i \leq k$. Now, the number of literals in $T^{v_{r_1}}$ is at most t. And since v_{r_2} is chosen independently at random from $N(u)$, we have,

$$\Pr_{\bar{v}}[(r_1, r_2) \text{ is intersecting }] \leq \frac{t}{d_U}$$

for every $1 \leq r_1 \neq r_2 \leq \ell$. Call $\bar{v}$ intersection-free, if it contains no intersecting pair of

coordinates. Since, there are ℓ^2 such pairs,

$$\Pr_{\bar{v}}[\bar{v} \text{ is not intersection-free }] \leq \frac{t\ell^2}{d_U}.$$

Now ϕ is consistent with the 0 examples with probability at least ε'. Again, by averaging, we have that for $\frac{\varepsilon'}{2}$ of the ℓ-tuples $\bar{v}$, ϕ is 0 on the points $Z_0^u[\alpha, \bar{v}]$ generated after choosing $\bar{v}$ in step 3a, with probability at least $\frac{\varepsilon'}{2}$. We call such ℓ-tuples $\bar{v}$ as 'good'. More formally we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is 'good' }] \geq \frac{\varepsilon'}{2},$$

where, for a given 'good' ℓ-tuple $\bar{v}$,

$$\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha, \bar{v}] \geq \frac{\varepsilon'}{2}.$$

Using union bound, we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is good, dense and intersection free}] \geq \nu \tag{7}$$

where,

$$\nu = \frac{\varepsilon'}{2} - \exp(-\varepsilon'\ell/8) - \frac{t\ell^2}{d_U}. \tag{8}$$

We now fix a good, dense and intersection-free ℓ-tuple $\bar{v} = (v_1, \ldots, v_t)$. Consider any $r \in D_{\bar{v}}$, $v_r \in M(u)$ and so $T^{v_r} \neq 0$. Moreover, since $\bar{v}$ is intersection free, the negative literals in T^{v_r} correspond to vertices that are not contained in any coordinate of $\bar{v}$. Therefore, the negative literals in T^{v_r} are always set to 1 on the points $Z_0^u[\alpha, \bar{v}]$ for any $\alpha \subseteq [m]$. Therefore, the term T^{v_r} will be 1 if all the variables (positive literals) in T^{v_r} are set to 1, which happens if $\pi_{uv_r}(W(v_r)) \subseteq \alpha$. This leads to the following key lemma.

Lemma 2.6.3. *If $\ell > \left(\frac{2}{\varepsilon'\mu^t}\right) \ln\left(\frac{2}{\varepsilon'}\right)$, then there must exist $r_1, r_2 \in D_{\bar{v}}$, $r_1 \neq r_2$ such that $\pi_{uv_{r_1}}(W(v_{r_1})) \cap \pi_{uv_{r_2}}(W(v_{r_2})) \neq \emptyset$.*

Proof: Assume that there is no such pair r_1 and r_2. Therefore, the events $(\pi_{uv_r}(W(v_r)) \subseteq \alpha)$ are independent events for $r \in D_{\bar{v}}$. From the discussion above, we have for any $r \in D_{\bar{v}}$,

$$\begin{aligned} \Pr_{\alpha}[T^{v_r} \text{ is 1 on } Z_0^u[\alpha, \bar{v}]] &= \Pr_{\alpha}[\pi_{uv_r}(W(v_r)) \subseteq \alpha] \\ &= \mu^{|\pi_{uv_r}(W(v_r))|} \\ &\geq \mu^t. \end{aligned} \tag{9}$$

Therefore, we have,

$$\begin{aligned}\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha,\bar{v}]] &\leq \Pr_{\alpha}\left[\bigwedge_{r=1}^{|D_{\bar{v}}|}(T^{v_r} \text{ is 0 on } Z_0^u[\alpha,\bar{v}])\right] \\ &= \Pr_{\alpha}\left[\bigwedge_{r=1}^{|D_{\bar{v}}|}(W(v_r) \not\subseteq \alpha)\right]\end{aligned}$$

and combining the independence of the events $(\pi_{uv_r}(W(v_r)) \subseteq \alpha)$ with Equation (9), we obtain,

$$\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha,\bar{v}]] \leq (1-\mu^t)^{|D_{\bar{v}}|}.$$

Now, since the left hand side is at least $\frac{\varepsilon'}{2}$, the above implies,

$$\begin{aligned}&|D_{\bar{v}}| \leq \left(\frac{1}{\mu^t}\right)\ln\left(\frac{2}{\varepsilon'}\right) \\ \Rightarrow\quad &\ell \leq \left(\frac{2}{\varepsilon'\mu^t}\right)\ln\left(\frac{2}{\varepsilon'}\right) \qquad (10)\end{aligned}$$

since $\bar{v}$ is dense. This proves the lemma.

■

In our construction we choose ℓ large enough depending on μ, ε and t, and then (independently of ℓ), choose d_U large enough (by parallel repetition), to ensure the following.

$$\ell > \left(\frac{2}{\varepsilon'\mu^t}\right)\ln\left(\frac{2}{\varepsilon'}\right) \qquad (11)$$

and

$$\nu \geq \frac{\varepsilon'}{4}. \qquad (12)$$

Note that the above analysis holds for any valid assignment Γ of terms to vertices. We are now ready to define a labeling to the vertices of $\mathcal{L}$.

Construction of labeling We will define a partial labeling σ_U, σ_V to the vertices in U and V respectively in the following randomized manner.

1. Let $u \in U$ be any given vertex. Choose a random vertex v' from $N(u)$. If $W(v') = \emptyset$ then do not assign any label to u. If not, select $i \in W(v')$ randomly, and let $\sigma_U(u) = \pi_{uv'}(i)$.

2. Let $v \in V$ be any vertex. If $W(v)$ is empty then do not assign any label to v, otherwise, let $\sigma_V(v) = i$ where i is randomly chosen from $W(v)$.

We will now analyze how many edges this labeling satisfies in expectation. Consider a random edge (u, v) of $\mathcal{L}$, selected by first choosing v randomly from U and then selecting v randomly from $N(u)$. Now, u is labeled by choosing a vertex v' at random from $N(u)$ and labeling u by $\pi_{uv'}(i)$ where i is chosen randomly from $W(v')$, unless $W(v') = \emptyset$. Therefore, the probability that, after fixing u, a random edge (u, v) is satisfied is same as the probability that $\pi_{uv}(\sigma_V(v)) = \pi_{uv'}(\sigma_V(v'))$ where v and v' are vertices selected uniformly at random from $N(u)$.

With probability $\frac{\varepsilon}{2}$, u is a 'good' vertex. Also, choosing two neighbors of u uniformly at random is same as choosing a random ℓ-tuple $\bar{v}$ (for $\ell \geq 2$) and then selecting two distinct coordinates of $\bar{v}$. In this process, with probability ν, a good, dense and intersection-free ℓ-tuple $\bar{v}$ is picked. From our choice of ℓ depending on μ, ε and t, we have the bounds given by (11), and (12) and combining with Lemma 2.6.3, we have that with probability $\frac{1}{\ell^2}$, the vertices v and v' are such that $\pi_{uv}(W(v)) \cap \pi_{uv'}(W(v')) \neq \emptyset$. And with further $\frac{1}{t^2}$ probability the labels for v and v' are consistent, i.e. $\pi_{uv}(\sigma_V(v)) = \pi_{uv'}(\sigma_V(v'))$.

Combining everything we have that the probability that a random edge (u, v) is satisfied is,

$$p^* = \left(\frac{\varepsilon}{2}\right) \nu \left(\frac{1}{\ell^2}\right) \left(\frac{1}{t^2}\right).$$

Now, since $\nu \geq \frac{\varepsilon'}{4} \geq \frac{\varepsilon}{8}$ and ℓ is chosen to depend only on μ, ε and t, the above probability depends only on μ, ε and t. Also, it implies that there is a labeling that satisfies p^* fraction of edges of $\mathcal{L}$, where p^* depends only on μ, ε and t. By choosing the soundness parameter of the NO instance η to be small enough, we obtain a contradiction.

2.7 Construction of t-Layered-CSP

In this section we will construct an instance of t-Layered-CSP we require for our reduction. First we will construct an appropriate PCP and then we will transform the PCP to a Multi Prover System with some desired properties. The Multi Prover System thus constructed can be thought of as an instance of t-Layered-CSP in a natural way.

We begin with the construction of the PCP. Our construction is very similar to the query efficient PCP constructed in [45]. The construction starts with an instance of MAX-3LIN problem which is: given a set of linear equations on n variables over a finite field, where each equation contains at most 3 variables, to find an assignment to the variables to satisfy the maximum number of equations. We construct the Raz Verifier using parallel repetition and the proofs are then encoded using Hadamard Codes. In order to obtain a PCP with a large alphabet, we take the encoding using Hadamard Codes over a large field. The analysis is similar to [45] and relies heavily on the techniques developed in [77] and [73] and a similar construction over finite Abelian groups in [26].

We start with the instance of MAX-3LIN constructed in [49] with completeness $1 - \left(2^{-\Omega(\sqrt{\log n})}\right)$ and soundness $1 - \Omega\left(\log^{-3} n\right)$. The prove the following theorem.

Theorem 2.7.1. *Given a 7-regular instance $\mathcal{A}$ of* MAX-3LIN *over $\mathbb{F}[2]$ on n variables such that unless* NP $\subseteq$ DTIME$(2^{O(\log^2 N)})$ *there is no polynomial time algorithm to distinguish between the following two cases,*

- YES Case*: There is an assignment to the variables of $\mathcal{A}$ that satisfies $1 - 2^{-\Omega(\sqrt{\log n})}$ fraction of the equations.*
- NO Case*: No assignment to the variables of $\mathcal{A}$ satisfies more than $1 - \Omega(\log^{-3} n)$ fraction of the equations.*

Note that the equations of MAX-3LIN are over $\mathbb{F}[2]$. However, we may consider them to be over $\mathbb{F}[2^r]$ where r is some parameter and still the above theorem still holds. This is because the additive group $(\mathbb{F}[2^r], +)$ is isomorphic to $(\mathbb{F}[2]^r, +)$. Therefore, we can substitute the equation $x_1 + x_2 + x_3 = b$, where $x_1, x_2, x_3, b \in \mathbb{F}[2]$ with the equation over $\mathbb{F}[2^r]$, $x'_1 + x'_2 + x'_3 = b_r$, where b_r is the element of $\mathbb{F}[2]^r$ with b in each or the r coordinates. Clearly, any assignment over $\mathbb{F}[2]$ can be extended to an assignment over $\mathbb{F}[2]^r$ by replicating it in every coordinate. Moreover, any assignment over $\mathbb{F}[2]^r$ that satisfies a particular equation must satisfy it in every coordinate, and so we can pick any coordinate and the corresponding assignment over $\mathbb{F}[2]$ will also satisfy all equations satisfied earlier.

And since $(\mathbb{F}[2^r], +) \cong (\mathbb{F}[2]^r, +)$, we can write the entire system of equations over the field $\mathbb{F}[2^r]$.

2.7.1 Raz Verifier

We construct the Raz Verifier starting with an instance $\mathcal{A}$ of Max-3LIN obtained from Theorem 2.7.1. For convenience we let the completeness of $\mathcal{A}$ be $1 - c(n)$ and soundness be $1 - s(n)$. The construction in [73] started with a Gap-3SAT instance, however we require constraints to be linear to be able to use Hadamard Codes instead of Long Codes, similar to the construction in [45]. Note that our instance $\mathcal{A}$ of Max-3LIN is over the field $\mathbb{F}[2^r]$ for some $r > 0$ to be fixed later, and has the same completeness and soundness as in Theorem 2.7.1.

Let $m > 0$ be a parameter to be fixed later. The Raz Verifier is given an instance $\mathcal{A}$ of Max-3LIN. It expects two proofs, P and Q. The proof P is supposed to contain, for every set U of m variables, a length m vector $P(U)$ over $\mathbb{F}[2^r]$ giving the assignment to the variables in U. Similarly, for every set W of m equations, $Q(W)$ is supposed to be a length $3m$ vector giving the assignment to the $3m$ variables in the set of equations W.

The verifier works by picking a set of $U = (x_i)_{i=1}^m$ of m variables and then picking a set of m equations $W = (C_i)_{i=1}^m$ where each equation C_i is selected randomly from the constantly many equations containing the variable x_i. The verifier reads $P(U)$ and $Q(W)$ from the proof and accepts iff $Q(W)$ satisfies all the equations $(C_i)_{i=1}^m$ and the values of the variables $(x_i)_{i=1}^m$ in $P(U)$ and $Q(W)$ are the same (call this projection test).

Completeness. In the YES case $\mathcal{A}$ has an assignment that satisfies $1 - c(n)$ fraction of the equations. Let both proofs P and Q be consistent with that assignment. Since, the instance $\mathcal{A}$ is regular, with probability at least $(1 - c(n))^m$ all the equations $W = (C_i)_{i=1}^m$ chosen in the construction above will be satisfied by the proof Q. Therefore, the completeness is at least $(1 - c(n))^m \geq (1 - mc(n))$.

Soundness. In the NO case any assignment to the variables of $\mathcal{A}$ satisfies at most $1 - s(n)$ fraction of the equations. Using Raz's Parallel Repetition Theorem [71], and the fact that each equation contains exactly 3 variables, we have the following upper bound.

Theorem 2.7.2. *There is an absolute constant $\kappa > 0$ such that, the soundness of the Raz Verifier on the instance of* MAX-3LIN *(over $\mathbb{F}[2^r]$) with soundness $(1 - s(n))$ is at most $(1 - s(n)^\kappa)^{(m/(\kappa r))}$.* [1]

2.7.2 Fourier Analysis

We will be working over the field $\mathbb{F}[2^r]$ for $r > 0$, which is a field extension of $\mathbb{F}[2]$. Let φ be the isomorphism from the additive group $(\mathbb{F}[2^r], +)$ to $(\mathbb{F}[2]^r, +)$. Define the following homomorphism ϕ from $(\mathbb{F}[2^r], +)$ to the multiplicative group $(\{-1, 1\}, .)$.

$$\phi(a) = \begin{cases} 1 & \text{if } \varphi(a) \text{ contains even number of 1s} \\ -1 & \text{otherwise} \end{cases}$$

for any $a \in \mathbb{F}[2^r]$. We now define the 'characters' $\psi_a : \mathbb{F}[2^r] \mapsto \{-1, 1\}$ for $a \in \mathbb{F}[2^r]$ as follows.

$$\psi_a(b) := \phi(ab)$$

The characters ψ_a satisfy the following properties.

$$\psi_0(b) = 1 \qquad \forall b \in \mathbb{F}[2^r]$$

$$\psi_a(0) = 1 \qquad \forall a \in \mathbb{F}[2^r]$$

$$\psi_{a+b}(c) = \psi_a(c)\psi_b(c)$$

and,

$$\sum_{a \in \mathbb{F}[2^r]} \psi_a(b) = \begin{cases} |\mathbb{F}[2^r]| & \text{if } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

We note that the 'character' functions form an orthonormal basis for the space $L^2(\mathbb{F}[2^r])$. We have that,

$$\langle \psi_a, \psi_b \rangle = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

[1] *Since in our case the constraints are projections, using Rao's [70] proof of parallel repetition we can eliminate the dependence over r.*

where,

$$\langle \psi_a, \psi_b \rangle := \mathrm{E}_{c \in \mathbb{F}[2^r]} \left[\psi_a(c) \psi_b(c) \right].$$

We now consider the vector space $\mathbb{F}[2^r]^m$ for some positive integer m. We define the 'characters' $\chi_\alpha : \mathbb{F}[2^r]^m \mapsto \{-1, 1\}$ for every $\alpha \in \mathbb{F}[2^r]^m$ as,

$$\chi_\alpha(f) := \phi(\alpha \cdot f), \qquad f \in \mathbb{F}[2^r]^m$$

where '$\cdot$' is the inner product in the vector space $\mathbb{F}[2^r]^m$. From the way we defined the characters ψ_a, we have,

$$\chi_\alpha(f) = \prod_{i=1}^{m} \psi_{\alpha_i}(f_i),$$

where α_i and f_i are the i^{th} coordinates of α and f respectively. The characters χ_α satisfy the following properties,

$$\begin{aligned} \chi_0(f) &= 1 \qquad && \forall f \in \mathbb{F}[2^r]^m \\ \chi_\alpha(0) &= 1 \qquad && \forall \alpha \in \mathbb{F}[2^r]^m \\ \chi_{\alpha+\beta}(f) &= \chi_\alpha(f)\chi_\beta(f) \\ \chi_\alpha(f+g) &= \chi_\alpha(f)\chi_\alpha(g) \end{aligned}$$

and,

$$\mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\chi_\alpha(f) \right] = \begin{cases} 1 & \text{if } \alpha = 0 \\ 0 & \text{otherwise} \end{cases}$$

The characters χ_α form an orthonormal basis for $L^2(\mathbb{F}[2^r]^m)$. We have,

$$\langle \chi_\alpha, \chi_\beta \rangle = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{otherwise} \end{cases}$$

where,

$$\langle \chi_\alpha, \chi_\beta \rangle := \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\chi_\alpha(f) \chi_\beta(f) \right].$$

Let $A : \mathbb{F}[2^r]^m \mapsto \mathbb{F}[2^r]$ be a function. We define $\widehat{A}_{\gamma,\alpha}$ to be the Fourier coefficient of the function $\psi_\gamma \circ A$ corresponding to the element χ_α of the basis, for $\alpha \in \mathbb{F}[2^r]^m$ and $\gamma \in \mathbb{F}[2^r]$. Formally,

$$\widehat{A}_{\gamma,\alpha} = \langle \psi_\gamma \circ A, \chi_\alpha \rangle = \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f)) \chi_\alpha(f) \right]$$

and therefore,

$$\psi_\gamma \circ A = \sum_{\alpha \in \mathbb{F}[2^r]^m} \widehat{A}_{\gamma,\alpha} \chi_\alpha.$$

The following is a useful lemma.

Lemma 2.7.3. *Let $A : \mathbb{F}[2^r]^m \mapsto \mathbb{F}[2^r]$ be a function such that $\exists h \in \mathbb{F}[2^r]^m$ and $\zeta \in \mathbb{F}[2^r]$ such that $A(f + \delta h) = A(f) + \delta\zeta$, for all $\delta \in \mathbb{F}[2^r]$. Then, if $\widehat{A}_{\gamma,\alpha} \neq 0$ for some $\alpha \in \mathbb{F}[2^r]^m$ and $\gamma \in \mathbb{F}[2^r]$, then $\alpha \cdot h = \gamma\zeta$.*

Proof: We have,

$$\begin{aligned}
\widehat{A}_{\gamma,\alpha} &= \langle \psi_\gamma \circ A, \chi_\alpha \rangle \\
&= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} [\psi_\gamma(A(f))\chi_\alpha(f)] \\
&= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} [\psi_\gamma(A(f + \delta h))\chi_\alpha(f + \delta h)]
\end{aligned}$$

for any $\delta \in \mathbb{F}[2^r]$. Therefore using the property of A we have,

$$\begin{aligned}
\widehat{A}_{\gamma,\alpha} &= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} [\psi_\gamma(A(f) + \delta\zeta)\chi_\alpha(f)\chi_\alpha(\delta h)] \\
&= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} [\psi_\gamma(A(f))\psi_\gamma(\delta\zeta)\chi_\alpha(f)\chi_\alpha(\delta h)] \\
&= \psi_\gamma(\delta\zeta)\chi_\alpha(\delta h)\mathrm{E}_{f \in \mathbb{F}[2^r]^m} [\psi_\gamma(A(f))\chi_\alpha(f)] \\
&= \psi_\gamma(\delta\zeta)\chi_\alpha(\delta h)\widehat{A}_{\gamma,\alpha}
\end{aligned}$$

and since $\widehat{A}_{\gamma,\alpha} \neq 0$, this implies,

$$\begin{aligned}
& \psi_\gamma(\delta\zeta) = \chi_\alpha(\delta h) \\
\Rightarrow \quad & \phi(\delta\gamma\zeta) = \phi(\alpha \cdot (\delta h)) \\
\Rightarrow \quad & \phi(\delta(\gamma\zeta))\phi(\delta(\alpha \cdot h)) = 1 \\
\Rightarrow \quad & \phi(\delta(\gamma\zeta + \alpha \cdot h)) = 1
\end{aligned}$$

for all $\delta \in \mathbb{F}[2^r]$. But since $\phi \not\equiv 1$, we must have that $\gamma\zeta + \alpha \cdot h = 0$, i.e. $\gamma\zeta = \alpha \cdot h$. This completes the proof. ∎

Hadamard Codes. In the construction of the PCP, the prover expects the Hadamard encodings of the vectors $P(U)$ and $Q(W)$ for the sets U and W in the construction of the Raz Verifier.

Definition 2.7.1. *For any positive integer t, the Hadamard Code of $p \in \mathbb{F}[2^r]^t$ is given by a function $Had_p : \mathbb{F}[2^r]^t \mapsto \mathbb{F}[2^r]$ where,*

$$Had_p(a) = p \cdot a$$

for all $a \in \mathbb{F}[2^r]^t$.

Note that the string $x = Q(W)$, $x \in \mathbb{F}[2^r]^{3m}$ that the Raz Verifier reads is supposed to satisfy certain linear constraints over $\mathbb{F}[2^r]$, given by $h_i \cdot x = \zeta_i$, where $h_i \in \mathbb{F}[2^r]^{3m}$ and $\zeta_i \in \mathbb{F}[2^r]$ for $1 \leq i \leq m$.

Let $\pi : \mathbb{F}[2^r]^{3m} \mapsto \mathbb{F}[2^r]^m$ be a *projection* that maps vectors in $\mathbb{F}[2^r]^{3m}$ to some fixed m coordinates. Let $\pi^{-1}(a)$ denote the unique vector $b \in \mathbb{F}[2^r]^{3m}$ such that $\pi(b) = a$ and is 0 on all other coordinates other than those that are projected by π.

Folding. Let B_x be the Hadamard Code of a vector $x \in \mathbb{F}[2^r]^{3m}$ that satisfies the constraints $h_i \cdot x = \zeta_i$ for $1 \leq i \leq m$. Let H be the subspace of $\mathbb{F}[2^r]^{3m}$ spanned by $\{h_i\}_{i=1}^m$. Let $h \in H$ be such that $h = \sum_{i=1}^m \rho_i h_i$, where $\rho_i \in \mathbb{F}[2^r]$ for $1 \leq i \leq m$. Then, we have that for any $a \in \mathbb{F}[2^r]^{3m}$,

$$B_x(a + h) = B_x(a) + \sum_{i=1}^{m} \rho_i \zeta_i.$$

So, we can enforce the folding over linear constraints in the following manner. For any $a \in \mathbb{F}[2^r]^{3m}$, let,

$$a = v_a + \sum_{i=1}^{m} \rho_i h_i$$

where v_a is the lexicographically smallest vector in the coset $a + H$. The verifier expects a function $B' : \mathbb{F}[2^r]^{3m} \mapsto \mathbb{F}[2^r]$ defined only on the distinguished vectors v_a for the coset $a + H$, and then computes the value of $B(a)$ as follows,

$$B(a) = B'(v_a) + \sum_{i=1}^{m} \rho_i \zeta_i.$$

We say that B is 'folded' over the linear constraints. Therefore, we can enforce the folding of the supposed Hadamard encodings of the assignments $Q(W)$, over the linear constraints given by the equations in W. The following crucial lemma follows from directly from Lemma 2.7.3.

Lemma 2.7.4. *For any $\gamma \in \mathbb{F}[2^r]$, if $\widehat{B}_{\gamma,\beta} \neq 0$ for some $\beta \in \mathbb{F}[2^r]^{3m}$, then $\beta \cdot h_i = \gamma\zeta_i$ for all $1 \leq i \leq m$.*

Eventually our analysis will show that the supposed Hadamard Code B for $Q(W)$ can be decoded to obtain the vectors β with probability proportional to $\widehat{B}^2_{\gamma,\beta}$. Since we have ensured the folding, Lemma 2.7.4 would imply that $\gamma^{-1}\beta$ is a valid assignment to the variables in $Q(W)$ that satisfies all the linear constraints.

2.7.3 Construction of the PCP

We now construct the PCP verifier. The verifier V_{lin} is given an instance of MAX-3LIN over $\mathbb{F}[2^r]$ with the completeness and soundness parameters as before. The verifier expects proofs (P', Q') which are Hadamard encodings of the proofs (P, Q) given to the Raz Verifier. For sets U and W of the Raz Verifier, $P'(U)$ and $Q'(W)$ are supposed to be Hadamard codes of $P(U)$ and $Q(W)$ respectively. The verifier V_{lin} proceeds as follows,

1. Pick a set U of m variables and ℓ sets $(W_j)_{j=1}^{\ell}$ independently in a manner similar to the Raz Verifier. Let π_j be the projection function between W_j and U for $1 \leq j \leq \ell$.

2. Let A be the supposed Hadamard Code of $P(U)$ and B_j be the supposed Hadamard code of $Q(W_j)$. The codes B_j are assumed to be folded over the linear constraints.

3. Pick $a_1, \ldots, a_\ell \in \mathbb{F}[2^r]^m$ and $b_1, \ldots, b_\ell \in \mathbb{F}[2^r]^{3m}$ randomly.

4. Accept iff for $1 \leq i, j, \leq \ell$

$$A(a_i) + B_j(b_j) = B_j(\pi_j^{-1}(a_i) + b_j).$$

The following is the main theorem about the properties of this PCP.

Theorem 2.7.5. *Given an instance $\mathcal{A}$ of* MAX-3LIN *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$,*

1. *V_{lin} uses $m \log n + O(\ell m r)$ random bits.*

2. *V_{lin} queries $\ell^2 + 2\ell$ positions from the proof.*

3. *If the instance $\mathcal{A}$ is a YES instance then there is a set $\bar{S}$ consisting of all the positions of the supposed encodings of $Q(W')$ for at most $mc(n)$ fraction of sets W', and an assignment τ^* to all the positions of the proof except those in $\bar{S}$ such that,*

   a. *(Strong Completeness) The verifier accepts on τ^* whenever none of the positions in $\bar{S}$ are queried.*

   b. *(Extendability) For any constraint q of the verifier which (possibly) queries positions from $\bar{S}$, there is an assignment τ_q to the positions in $\bar{S}$ queried in q, such that τ^* extended by τ_q satisfies the constraint q.*

4. *If the instance $\mathcal{A}$ is a NO instance then the probability that the verifier accepts is at most $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$, for $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(|\mathbb{F}[2^r]| - 1)^{\ell^2}$, for some universal constant κ.*

Proof: Properties 1 and 2 of verifier are clear. Assume that the MAX-3LIN $\mathcal{A}$ was a YES instance and had an assignment σ to the variables such that $1-c(n)$ fraction of the equations were satisfied. Call the equations not satisfied as 'bad'. Therefore, at most $mc(n)$ fraction of the sets W' of the Raz Verifier are 'bad' i.e. they contain a 'bad' equation. Let the assignments $P(U)$ and $Q(W)$ be consistent with σ and $P'(U)$ and $Q'(W)$ be the respective Hadamard encodings given to verifier V_{lin}, for all sets of variables U and all sets W that are not 'bad'. We let the set $\bar{S}$ of positions in the proof correspond to the supposed Hadamard encodings of the assignment to the 'bad' sets W'.

Let U and $(W_j)_{j=1}^{\ell}$ be such that none of the W_j are 'bad', and A and $(B)_{j=1}^{\ell}$ be the Hadamard encodings of the assignments given by σ, which is a satisfying assignment for the sets U and $(W_j)_{j=1}^{\ell}$.

$$A(a_i) = a_i \cdot P(U) \qquad B_j(b_j) = b_j \cdot Q(W_j)$$

$$\begin{aligned} B_j(\pi_j^{-1}(a_i) + b_j) &= (\pi_j^{-1}(a_i) + b_j) \cdot Q(W_j) \\ &= (\pi_j^{-1}(a_i) \cdot Q(W_j) + b_j \cdot Q(W_j) \\ &= a_i \cdot \pi_j(Q(W_j)) + b_j \cdot Q(W_j) \\ &= A(a_i) + B_j(b_j) \end{aligned} \tag{13}$$

since $\pi_j(Q(W_j)) = P(U)$ as σ satisfies U and W_j. This proves the Strong Completeness property. Observe that every constraint of the verifier is a set of linear equalities of the form $A(a_i) = B(b_j) + B(\pi_j^{-1}(a_i) + b_j)$. Also, for $a_i \neq a_{i'}$, $\pi_j^{-1}(a_i) - \pi_j^{-1}(a_{i'}) \notin H^j$, where H^j is the subspace spanned by the linear constraints over which the supposed encoding B_j is folded. Therefore, if $a_i \neq a_{i'}$ then $B_j(\pi_j^{-1}(a_i) + b_j)$ and $B_j(\pi_j^{-1}(a_{i'}) + b_j)$ are distinct positions in the B_j. So, within any constraint every equation has a unique variable. Then, if B_j is an encoding corresponding to a 'bad' set W_j', the proof Q' can be extended to satisfy equations involving positions in B_j, in a given constraint involving B_j. This implies that for any given constraint (possibly involving positions from $\bar{S}$), the encodings P', Q'given by σ, can be extended to the positions in $\bar{S}$ queried by the given constraint so that the constraint is satisfied. This proves the Extendability property, and completes the analysis for the YES case.

We now analyze the NO case. We assume that the verifier accepts with probability $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$. It was shown in [26] that the probability of acceptance of the verifier is,

$$\frac{1}{|\mathbb{F}[2^r]|^{\ell^2}} \sum_{S \subseteq [\ell] \times [\ell]} \mathrm{E}\left[T_S\right] \tag{14}$$

where,

$$T_S = \prod_{(i,j) \in S} \left(\sum_{\gamma \in \mathbb{F}[2^r] \setminus \{0\}} \psi_\gamma(A(a_i) + B_j(b_j) + B_j(\pi^{-1}(a_i) + b_j)) \right), \tag{15}$$

and the expectation is over the choice of $U, (W_j)_{j=1}^{\ell}, (a_i)_{i=1}^{\ell}, (b_j)_{j=1}^{\ell}$ and where $T_\emptyset = 1$.

If the above probability is $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$, there must be a nonempty set $S \subseteq [\ell] \times [\ell]$, such that $|\mathrm{E}[T_S]| \geq \delta$. This term was analyzed in [26] and we use their analysis. In [26], since Long Codes are analyzed, the notion of projections is slightly different from ours, but the proof is exactly the same even for our case. The analysis in [26] also had a certain perturbation parameter, which is 0 in our case. We state the following theorem and refer the reader to the proof in Section 4.5 of [26].

Theorem 2.7.6. *Suppose that $|\mathrm{E}[T_S]| \geq \delta > 0$ for some nonempty set $S \subseteq [\ell] \times [\ell]$. Number the elements in S such that there is at least one element of the form $(1, j)$ and all the elements of that form are $(1,1), \ldots, (1,d)$, where $d \leq \ell$. Then there exist $\gamma_1, \ldots, \gamma_d \in$*

$\mathbb{F}[2^r] \setminus \{0\}$ *such that,*

$$\mathrm{E}_{U,W_1,\ldots,W_d}[\Delta] \geq \frac{\delta^2}{(|\mathbb{F}[2^r]|-1)^{|S|}}$$

where,

$$\Delta = \sum_{\substack{\alpha,\beta_1,\ldots,\beta_d \\ \alpha=\pi_1(\beta_1)+\cdots+\pi_d(\beta_d)}} \widehat{A}^2_{\gamma,\alpha}\widehat{B}^2_{1,\gamma_1,\beta_1}\ldots\widehat{B}^2_{d,\gamma_d,\beta_d}$$

and,

$$\gamma = \gamma_1 + \cdots + \gamma_d$$
$$\widehat{A}_{\gamma,\alpha} = \langle \psi_\gamma \circ A, \chi_\alpha \rangle$$
$$\widehat{B}_{j,\gamma_j,\beta_j} = \left\langle \psi_{\gamma_j} \circ B_j, \chi_{\beta_j} \right\rangle$$

We now define proofs (P,Q) for the Raz Verifier as follows. For a set W, pick β with probability $\widehat{B}^2_{\gamma_1,\beta}$ and define $Q(W)$ to be $\gamma_1^{-1}\beta$, where B is the supposed encoding of $Q(W)$. Note that since $\widehat{B}_{\gamma_1,\beta} \neq 0$ for any set we pick, and $\gamma_1 \neq 0$ by Lemma 2.7.4, $\gamma_1^{-1}\beta$ satisfies all the equations of W. For a set U, pick sets $(W_j)_{j=2}^d$ at random as in the Raz Verifier, and pick $(\beta_j)_{j=2}^d$ with probability $\prod_{j=2}^d \widehat{B}^2_{j,\gamma_j,\beta_j}$, and choose α with probability $A^2_{\gamma,\alpha}$. Define $P(U)$ to be $\gamma_1^{-1}(\alpha + \sum_{j=2}^d \pi_j(\beta_j))$. Since $\gamma_1 \neq 0$, we have,

$$\begin{aligned}
& \gamma_1^{-1}(\alpha + \sum_{j=2}^d \pi_j(\beta_j)) = \pi_1(\gamma_1^{-1}\beta_1) \\
\iff\ & \gamma_1^{-1}(\alpha + \sum_{j=2}^d \pi_j(\beta_j)) = \gamma_1^{-1}(\pi_1(\beta_1)) \\
\iff\ & \alpha + \sum_{j=2}^d \pi_j(\beta_j) = \pi_1(\beta_1) \\
\iff\ & \alpha = \sum_{j=1}^d \pi_j(\beta_j).
\end{aligned}$$

Using the above observation it is easy to see that the acceptance probability of the Raz Verifier is given by $\mathrm{E}[\Delta]$ and therefore,

$$\begin{aligned}
\Pr[\text{Raz Verifier accepts}] &\geq \frac{\delta^2}{(|\mathbb{F}[2^r]|-1)^{|S|}} \\
&\geq \frac{\delta^2}{(|\mathbb{F}[2^r]|-1)^{\ell^2}}
\end{aligned}$$

since $|S| \leq \ell^2$. Using the bound given by Theorem 2.7.2, we obtain,

$$\delta^2 \leq (1 - s(n)^\kappa)^{(m/(\kappa r))}(|\mathbb{F}[2^r]| - 1)^{\ell^2}$$

which completes the analysis of the NO case. ∎

2.7.4 Construction of Multi Prover System

We will now give a reduction from the PCP system constructed to an appropriate Multi Prover System. For convenience we shall call the PCP system constructed in the previous subsection as PCP_1. Also, we let $t = \ell^2 + 2\ell$ and $k = |\mathbb{F}[2^r]|$. Clearly, PCP_1 is a t-query PCP where the answers are from $[k]$, with the properties specified in Theorem 2.7.5. We construct a Multi Prover System MIPS_1 as follows. Let $P_1, \ldots, P_t$ be t provers. The verifier V_{MIPS_1} computes the t queries of V_{lin}, say $q_1, \ldots, q_t$. It computes a random permutation $\nu : [t] \mapsto [t]$ and sends q_i to $P_{\nu(i)}$, for all $1 \leq i \leq t$ and expects answers from each prover from the set $[k]$. The acceptance predicate of V_{MIPS_1} is the same as V_{lin}. Let $\mathcal{Q}$ be the set of queries that V_{lin} makes, which is the set of positions in the proof expected by V_{lin}. Let $\mathcal{Q}_i$ be the set of queries sent to P_i by V_{MIPS_1}. Clearly, $\mathcal{Q}_i = \mathcal{Q}$ for all $1 \leq i \leq t$. It is easy to see that the completeness of V_{MIPS_1} is same as that of V_{lin}. It can be shown [78] that if the soundness of PCP_1 is ε then the soundness of V_{MIPS_1} is at most $t^t\varepsilon$. It is easy to check that the properties of Strong Completeness and Extendability hold. Analogous to the PCP construction, for every prover P_i, there is a set of 'bad' queries $\bar{S}_i = \bar{S}$ consisting of the positions of the encodings of $Q(W')$ for 'bad' sets W'. Let $\mu_i(\bar{S}_i)$ be the probability that the i^th query $q_i \in \bar{S}_i$. From the construction of PCP_1, it can be seen that $\mu_i(\bar{S}_i) \leq mc(n)$ for $1 \leq i \leq t$. We summarize the properties in the following theorem.

Theorem 2.7.7. *Given a 7-regular instance of* Max-3Lin *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$, for parameters m, k and t, (where $k = 2^r$ and $t = \ell^2 + 2\ell$), there is t prover system $MIPS_1$ with provers $P_1, \ldots, P_t$ and verifier V_{MIPS_1} such that,*

1. *The verifier uses $m \log n + O(\ell m r) + t \log t$ random bits to compute a query $\bar{q} =$*

$(q_1, \ldots, q_t)$ *where* q_i *is sent to* P_i *and an answer from* $[k] = [2^r]$ *is expected, for all* $1 \leq i \leq t$. *Let* $\mathcal{Q}_i$ *be the set of queries given to prover* P_i. *Then* $|\mathcal{Q}_1| = \ldots |\mathcal{Q}_t|$.

2. *If the* MAX-3LIN *instance is a YES instance, then there is a set* $\bar{S}_i \subseteq \mathcal{Q}_i$ *such that* $\mu_i(\bar{S}_i) \leq mc(n)$ *where* $\mu_i(\bar{S}_i)$ *is the probability that* $q_i \in \bar{S}_i$. *Furthermore,*

   a. *(Strong Completeness) There is a strategy* $\sigma_i^* : \mathcal{Q}_i \setminus \bar{S}_i \mapsto [k]$ $(1 \leq i \leq t)$ *of the provers such that the verifier accepts on all queries* $\bar{q}$ *such that* $q_j \notin \bar{S}_j$ *for all* $1 \leq j \leq t$.

   b. *(Extendability) For any given query* $\bar{q}$ *of the verifier (possibly containing query* $q_i \in \bar{S}_i$ *to individual provers* P_i*), the strategy given by* σ_i^* *can be extended to the queries from* $\bar{S}_i$ *contained in* $\bar{q}$ *so that the verifier accepts on the query* $\bar{q}$.

4. *If the instance of* MAX-3LIN *is a NO instance then the probability that the verifier accepts is at most* $t^t(k^{-\ell^2} + \delta)$, *where* $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2}$, *for some universal constant* κ.

We also need the condition that the queries are uniformly distributed over the set of all possible queries to prover P_i for all $1 \leq i \leq t$. For this construct another verifier V_{MIPS_2} for a Multi Prover System MIPS_2. Let R_{i,q_i} be the set of all random strings to V_{MIPS_1} that generate the query q_i to prover P_i. Then the verifier V_{MIPS_2} computes a query $\bar{q} = (q_1, \ldots, q_t)$ of V_{MIPS_1}, and sends the query $\bar{q}' = ((q_1, r_{1,q_1}), \ldots, (q_t, r_{t,q_t}))$ where r_{i,q_i} is a string uniformly chosen from R_{i,q_i}. The verifier expects answer to q_i from prover P_i, and the acceptance predicate remains the same. Clearly, sending uniformly chosen random strings r_{i,q_i} does not change the completeness, since provers can disregard them, and they do not provide any information to provers, so the soundness remains the same. In MIPS_2, let $\mathcal{Q}_i'$ be the set of all queries to P_i. It can be seen that $|\mathcal{Q}_1'| = \cdots = |\mathcal{Q}_t'|$ and the queries are uniformly distributed over the sets $|\mathcal{Q}_i'|$. Essentially, every query of MIPS_1 is replicated proportional to the probability it is queried. Let the corresponding 'bad' set $\bar{S}_i'$ be the set

of all queries (q_i, r) such that $q_i \in \bar{S}_i$ for all $1 \leq i \leq t$. We have the following,

$$\begin{aligned}
\frac{|\bar{S}'_i|}{|\mathcal{Q}'_i|} &= \Pr_{V_{MIPS_2} \to (q_i,r)}[(q_i, r) \in \bar{S}'_i] \\
&= \Pr_{V_{MIPS_1} \to q_i}[q_i \in \bar{S}_i] \\
&= \mu_i(\bar{S}_i) \\
&\leq mc(n) \qquad (16)
\end{aligned}$$

for all $1 \leq i \leq t$. It is easy to see that the properties of Strong Completeness and Extendability are also satisfied. The number of random bits used by V_{MIPS_2} is at most t times that of V_{MIPS_1}. We summarize the properties of MIPS_2 in the following theorem.

Theorem 2.7.8. *Given a 7-regular instance of* Max-3LIN *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$, for parameters m, k and t, (where $k = 2^r$ and $t = \ell^2 + 2\ell$), there is t prover system $MIPS_2$ with provers $P_1, \ldots, P_t$ and verifier V_{MIPS_2} such that,*

1. *The verifier uses $t(m \log n + O(\ell m r) + t \log t)$ random bits to compute a query $\bar{q}' = (q'_1, \ldots, q'_t)$ where q'_i is sent to P_i and an answer from $[k] = [2^r]$ is expected, for all $1 \leq i \leq t$. Let $\mathcal{Q}'_i$ be the set of queries given to prover P_i. Then $|\mathcal{Q}'_1| = \cdots = |\mathcal{Q}'_t|$ and the queries are uniformly distributed over each $\mathcal{Q}'_i$.*

2. *If the* Max-3LIN *instance is a YES instance, then there is a set $\bar{S}'_i \subseteq \mathcal{Q}'_i$ such that*

$$\frac{|\bar{S}'_i|}{|\mathcal{Q}'_i|} \leq mc(n).$$

   Furthermore,

   a. *(Strong Completeness) There is a strategy $\sigma'^*_i : \mathcal{Q}'_i \setminus \bar{S}'_i \mapsto [k]$ $(1 \leq i \leq t)$ of the provers such that the verifier accepts on all queries $\bar{q}'$ such that $q'_j \notin \bar{S}'_j$ for all $1 \leq j \leq t$.*

   b. *(Extendability) For any given query $\bar{q}'$ of the verifier (possibly containing query $q'_i \in \bar{S}'_i$ to individual provers P_i), the strategy given by σ'^*_i can be extended to the queries from $\bar{S}'_i$ contained in $\bar{q}'$ so that the verifier accepts on the query $\bar{q}'$.*

4. *If the instance of* Max-3LIN *is a NO instance then the probability that the verifier accepts is at most* $t^t(k^{-\ell^2} + \delta)$*, where* $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2}$*, for some universal constant* κ*.*

There is a canonical reduction from the above Multi Prover System, MIPS_2 to a t-Layered-CSP instance with the vertices of ith layer being $\mathcal{Q}'_i$ and the hyperedges being the constraints over t vertices, one from each layer, corresponding to the queries made by the verifier. The set of vertices in V' of t-Layered-CSP corresponds to $\cup_{i=1}^{t} \bar{S}'_i$. We now set the parameters used in our reduction, which along with reduction to the Max-3LIN instance in [49] would prove Theorem 2.3.4.

We start with the instance Max-3LIN of [49] on n variables with $c(n) = 2^{-\Omega(\sqrt{\log n})}$ and $s(n) = \Omega(\log^{-3} n)$. We take $m = \theta(\log^{3\kappa+3} n)$ and $r = \theta(\log \log n)$ such that $k = \theta(\log^{6\kappa+8} n)$. Now let $N = |V|$ be the number of vertices in the t-Layered-CSP instance. From the properties of MIPS_2, we have $\log N = \theta(\log^{3\kappa+4} n)$. Moreover, the size of the vertex set V' of the t-Layered-CSP is $Nmc(n) \leq N/(2^{(\log N)^{(1/(10\kappa+20))}})$ for large enough N. The size of the label set $k = \theta(\log^{6\kappa+8} n) = \theta(\log^2 N)$. Since $s(n) = \Omega(\log^{-3} n)$, we have $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2} = 2^{-\Omega(\log^2 n)}(k-1)^{\ell^2}$. Therefore, the soundness $t^t(k^{-\ell^2} + \delta) = k^{-t+O(\sqrt{t})}$.

The above analysis completes the construction of the t-Layered-CSP instance with the desired properties in Theorem 2.3.4.

2.8 Conclusion

We show that, given the truth table of a function on d variables, it is hard to minimize the size of an equivalent DNF formula to within $\Omega(d^{1-\varepsilon})$ for any $\varepsilon > 0$, unless NPis in quasipolynomial time. The result matches the best known approximation factor of $O(d)$ up to lower order multiplicative terms. Our technique combines a simple gadget reduction with a specialized multilayered CSP which might have applications in other contexts. It would be interesting to study whether there is a $\Omega(d)$ factor hardness of approximation for this problem.

We also study the problem of learning small DNFs and for any constants $\varepsilon, t > 0$, we show that it is hard (unless NP = RP) to PAC-learn within accuracy $\frac{1}{2} + \varepsilon$ (i) a two term DNF by a t-term DNF, and (ii) a AND formula by a t-CNF under arbitrarily small adversarial noise. It would be interesting to extend hardness of weak learning results for polynomial size DNFs. For properly learning polynomial size DNFs only a $1 - o(1)$ hardness factor for the accuracy is known [29].

One limitation of the $\frac{1}{2} + \varepsilon$ inapproximability results obtained in this chapter is that the error ε is an arbitrarily small *constant*, and can at the most be pushed down to some sub-polynomially decreasing function of the dimension of the problem. However, it is desirable to obtain a result where ε is a polynomially decreasing function as it would (more emphatically) rule out boosting as a means of efficiently learning. Unfortunately, the current techniques do not seem adequate enough to prove such a strong result. The biggest hurdle in this direction is the lack of a Label-Cover instance with polynomially low soundness, the construction of which is amongst the most challenging problems in PCP theory. This limitation is pervasive to all known threshold hardness of learning results based on complexity assumptions, including the ones in this thesis, and resolving it remains an important open question.

CHAPTER III

HARDNESS OF LEARNING INTERSECTION OF TWO HALFSPACES

In this chapter we prove the hardness result for learning intersection of two halfspaces given by Theorem 1.6.4. We begin by giving an overview of the main techniques involved in the proof.

3.1 Overview

We start by restating the main result proved in this chapter.

Theorem. (1.6.4 restated) *Let ℓ be any fixed integer and $\varepsilon > 0$ be an arbitrarily small constant. Then, given a set of labeled points in $\mathbb{R}^n$ with a guarantee that there is an intersection of two halfspaces that classifies all the points correctly, there is no polynomial time algorithm to find a function f of up to ℓ linear threshold functions that classifies $\frac{1}{2} + \varepsilon$ fraction of points correctly, unless* NP = RP.

As stated in Section 1.6.2 the result is optimal since an arbitrary halfspace or its complement has a success rate of $\frac{1}{2}$ on any given data set, and it implies hardness of PAC learning intersection of two halfspaces by a function of constantly many linear thresholds, to within an accuracy of $\frac{1}{2} + \varepsilon$, for any $\varepsilon > 0$. Consequently it provides evidence that the approach of weak learning intersection of two halfspaces with a function of a constant number of halfspaces followed by boosting may not work.

Informal description of the reduction

The reduction starts with an instance of (a variant) of the Label-Cover problem on n vertices and label set $[k]$. It produces an instance in nk dimensional space with a block of k dimensions for each vertex. In the first step we create a set of points for each vertex which are $\{-1, 1\}$ combinations in the k dimensions corresponding to the vertex and 0

everywhere else. These points simulate a junta test in the following manner. Suppose that a hyperplane $\langle \mathbf{r}, \mathbf{x} \rangle = c$ passes close to many points corresponding to a particular vertex v, then $\mathbf{r}$ has the property that whatever mass it has in the k coordinates corresponding to v is concentrated in a few coordinates. However, for this to work, one has to ensure that $\mathbf{r}$ has *some* non-negligible mass in those k coordinates to begin with.

To ensure this, in the second step we replace each point created in the first step with two small spheres of random points, where points in one of the spheres are labeled '+' and those in the other are labeled '−'. If these spheres are sufficiently 'dense', one can show that any hyperplane that separates a pair of such spheres must have non negligible mass in the coordinates corresponding to the vertex v. We use the fact that the class of halfspaces has polynomially bounded VC dimension and therefore with high probability a polynomially large set of random points on a sphere is an ε-sample for all halfspaces.

In order to enforce consistency between the labels of different vertices of the instance, the third step involves a folding procedure over a subspace defined by the constraints of the instance. A similar folding technique (albeit over $\mathbb{F}[2]$) was used in the reduction of [35], which is included in Chapter 4 of this thesis. A broader overview of this technique (over $\mathbb{F}[2]$) is included in Section 4.1. Our construction is such that the existence of a 'good' function of ℓ halfspaces gives us one single halfspace which can be used to extract a good labeling to the instance.

Remark. The gap instance that we construct is such that in the YES case, the set of points is classified correctly by the intersection of two parallel halfspaces of the form $\langle \mathbf{r}, \mathbf{x} \rangle \geq -c$ and $\langle \mathbf{r}, \mathbf{x} \rangle \leq c$, for some $c > 0$. This implies that the points are correctly classified by the degree 2 polynomial given by $(\langle \mathbf{r}, \mathbf{x} \rangle)^2 \leq c^2$. So, using linear programming a degree 2 polynomial can be efficiently found that classifies the points in our instance. However, the polynomial obtained may not be factorizable into two parallel hyperplanes.

3.2 Preliminaries

We start by formally defining the problem.

Definition 3.2.1. *An instance of* Intersection-Halfspace$_\ell$ *is a set of points in* $\mathbb{R}^n$

each labeled either '+' or '−' and the goal is to find a function of at most ℓ linear threshold functions (halfspaces) which correctly classifies the maximum number of points, where a '+' point is classified correctly if it lies inside the intersection and a '−' point is classified correctly if it lies outside of it.

We show that the problem INTERSECTION-HALFSPACE$_\ell$ is hard by giving a gap-preserving reduction from a variant of the LABEL-COVER problem. For the purposes of our reduction we require the LABEL-COVER instance to satisfy a certain 'smoothness' property. Similar 'smooth' versions of LABEL-COVER have been used in earlier hardness reductions [46, 40, 50]. In addition to smoothness we also require that sufficiently large induced subgraphs of the instance have a large number of edges. We define the following version of the LABEL-COVER problem that captures both these additional properties.

Definition 3.2.2. *An instance $\mathcal{L}$ of* SMOOTH-LABEL-COVER(t, μ, ν, k, m) *consists of a (multi)graph $G(V, E)$ and mappings $\{\pi_{u,e}\}_{e \in E, u \in e}$ where $\pi_{u,e} : [k] \mapsto [m]$. A labeling $\sigma : V \mapsto [k]$ is said to satisfy an edge e between u and v if $\pi_{u,e}(\sigma(u)) = \pi_{v,e}(\sigma(v))$. The instance satisfies:*

- *(Smoothness:) For any vertex $u \in V$ and any set $S \subseteq [k]$ of size at most t,*

$$\Pr_{e:u \in e}[\exists i, j \in S, i \neq j, \text{ s.t. } \pi_{u,e}(i) = \pi_{u,e}(j)] \leq \mu,$$

  where the probability is taken over a random edge incident on u.

- *For any $V' \subseteq V$ such that $|V'| = \xi|V|$, the induced subgraph on V' has at least $(\xi^2/2)|E|$ edges, for any $\xi \geq \nu$.*

The following theorem is proved using the PCP Theorem [8, 6] combined with Raz's Parallel Repetition Theorem [71]. We give a proof of the theorem in Section 3.6 based on the smooth version of LABEL-COVER constructed in [50].

Theorem 3.2.1. *For any constant t and arbitrarily small constants $\mu, \nu, \eta > 0$, there exist constants k and m such that given an instance $\mathcal{L}$ of* SMOOTH-LABEL-COVER(t, μ, ν, k, m) *it is* NP*-hard to distinguish between the following two cases:*

- YES Case/Completeness*: There is a labeling to the vertices of $\mathcal{L}$ which satisfies all the edges.*

- NO Case/Soundness*: No labeling to the vertices of $\mathcal{L}$ satisfies more than η fraction of the edges.*

We give a gap-preserving reduction from Smooth-Label-Cover to Intersection--Halfspace$_\ell$ stated in the following theorem, which along with Theorem 3.2.1 implies Theorem 1.6.4.

Theorem 3.2.2. *For any constant $\varepsilon > 0$ and integer $\ell > 0$, there is a randomized polynomial time reduction from an instance $\mathcal{L}$ of* Smooth-Label-Cover(t, μ, ν, k, m) *to an instance $\mathcal{I}$ of* Intersection-Halfspace$_\ell$ *for appropriately chosen parameters t, μ, ν and soundness η, such that,*

- YES Case/Completeness*: If $\mathcal{L}$ is a* YES *instance, then there is an intersection of two halfspaces which correctly classifies all the points in instance $\mathcal{I}$.*

- NO Case/Soundness*: If $\mathcal{L}$ is a* NO *instance, then with probability at least $\frac{9}{10}$, there is no function of up to ℓ linear threshold functions that correctly classifies more than $\frac{1}{2} + \varepsilon$ fraction of points in instance $\mathcal{I}$.*

3.3 Reduction

The reduction proceeds in three steps. In the first step we construct an initial set of unlabeled points, from the instance of Smooth-Label-Cover. In the second step we replace each initial point with two small spheres of points, with points in one sphere labeled '+' and points in the other labeled '−'. The third step consists of reducing the problem into a lower dimensional space via a *folding* over the subspace induced by the consistency constraints of the Label-Cover instance. In the end we obtain a set of points each labeled either '+' or '−' as an instance of Intersection-Halfspace$_\ell$ for a given constant ℓ.

3.3.1 Step 1: Initial Unlabeled Point Set

We start with an instance $\mathcal{L}$ of Smooth-Label-Cover(t, μ, ν, k, m), where we will fix t, μ and ν later. Let $|V| = n$. First we define the space in which the points lie. For every vertex $v \in V$, we have a set of k coordinates labeled by $M(v) := \{v(i)\}_{i=1}^{k}$. The complete set of coordinates is the union of these sets over all vertices, say $\mathcal{M} := \bigcup_{v \in V} M(v)$. Therefore, the points we construct lie in nk dimensional real space $\mathbb{R}^{\mathcal{M}}$. The construction of the points is as follows.

1. For every vertex v, define,

$$s(v) := \quad \{\mathbf{x} \in \mathbb{R}^{\mathcal{M}} \;\mid\; \forall i \in [k],\ \mathbf{x}(v(i)) \in \{-1, 1\} \text{ and } \quad \mathbf{x}(u(i)) = 0, \forall u \neq v\}.$$

   Thus, $s(v)$ is the set of all vectors that are $\{-1, 1\}$ combinations on the coordinates $M(v)$ and 0 on all other coordinates. Note that $|s(v)| = 2^k$ for all $v \in V$.

2. Let $S = \bigcup_{v \in V} s(v)$. Clearly $|S| = n \cdot 2^k$, since the sets $s(v)$ are disjoint for all $v \in V$.

We output S, as the initial set of points at the end of Step 1. One would like to say that if any hyperplane say $\langle \mathbf{r}, \mathbf{x} \rangle = a$, passes through a significant fraction of the points in S, then $\mathbf{r}$ should be close to a 'junta' i.e. most of the mass of the vector $\mathbf{r}$ should be concentrated in a few coordinates of $M(v)$ for a significant fraction of vertices $v \in V$. However, it is possible that $\mathbf{r}$ has zero mass in the coordinates $M(v)$ for almost all $v \in V$ and yet the hyperplane $\langle \mathbf{r}, \mathbf{x} \rangle = 0$ passes through most of the points in S. To overcome this problem, we replace each point in S with two spheres of points as described in Step 2.

3.3.2 Step 2: Constructing Spheres of Labeled Points

We start with the set of points $S = \bigcup_{v \in V} s(v)$ constructed in Step 1. For every point in S, we create two 'spheres' of points separated by a small distance. This step is randomized as it requires sampling a suitable number of points from a unit sphere. The following lemma is proved in Section 3.5.

Lemma 3.3.1. *Let $\varepsilon' > 0$ be any constant and n be a sufficiently large integer. Let R be a set of $(nk)^2$ unit vectors chosen uniformly at random in nk-dimensional real space. Then*

with probability at least $1 - 1/n$*, the set* R *satisfies the following property: for any subset* $T \subseteq R$ *such that* $|T| = \varepsilon'|R|$ *and for any unit vector* $\mathbf{r}$*, there exist* $\mathbf{z}', \mathbf{z}'' \in T$*, such that* $|\langle \mathbf{r}, \mathbf{z}' \rangle - \langle \mathbf{r}, \mathbf{z}'' \rangle| \geq \varepsilon'/(100\sqrt{nk})$.

Now we describe our construction in Step 2.

1. Set parameters [1] $\delta = 2^{-(nk)^{100}}$ and $\gamma = 1/(100\sqrt{n})$.

2. Let R be the set of $(nk)^2$ unit vectors in $\mathbb{R}^{\mathcal{M}}$ as in Lemma 3.3.1 (ε' will be chosen later and is related to the soundness parameter of the reduction).

3. Let $\mathbf{x}$ be a point in S. Construct two sets $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ as follows,

$$\alpha(\mathbf{x}) := \{(1-\delta)\mathbf{x} + \delta\gamma\mathbf{z} \mid \mathbf{z} \in R\}$$

   and,

$$\beta(\mathbf{x}) := \{(1+\delta)\mathbf{x} + \delta\gamma\mathbf{z} \mid \mathbf{z} \in R\}.$$

4. For every vertex $v \in V$, let $A(v) := \bigcup_{\mathbf{x} \in s(v)} \alpha(\mathbf{x})$ and $B(v) := \bigcup_{\mathbf{x} \in s(v)} \beta(\mathbf{x})$.

5. Output the sets $A := \bigcup_{v \in V} A(v)$ and $B := \bigcup_{v \in V} B(v)$.

The points created have the property that any hyperplane that separates the sets $\alpha(\mathbf{x})$ from $\beta(\mathbf{x})$ for a significant fraction of points $\mathbf{x} \in S$, must essentially be a 'junta' in the coordinates $M(v)$ for a significant fraction of vertices $v \in V$. This property will be formally stated and used in the soundness analysis to decode a labeling for the instance $\mathcal{L}$. In conjunction with this property, one needs to enforce the consistency constraints of the instance $\mathcal{L}$. We achieve this in the third step of the reduction, by folding over a subspace defined by these constraints.

3.3.3 Step 3: Folding and Final Labeled Point Set

For the sake of convenience, let $<$ be any arbitrary total order on V. Let e be an edge between u and v in G, with $u < v$. Let $\mathbf{h}_e^j \in \mathbb{R}^{\mathcal{M}}$, for $j \in [m]$, be defined in the following

[1] The parameter δ here is essentially the *margin* in the YES case(up to a polynomial factor). We set δ to be exponentially small in n and k . However, our reduction goes through even if δ is taken to be $O(\frac{1}{n^2})$.

manner: set $\mathbf{h}_e^j(u(i)) = 1$ for all $i \in \pi_{u,e}^{-1}(j)$, set $\mathbf{h}_e^j(v(i)) = -1$ for all $i \in \pi_{v,e}^{-1}(j)$ and set all the other coordinates to 0. Note that for any vector $\mathbf{r} \in \mathbb{R}^{\mathcal{M}}$,

$$\forall u, v \in e, e \in E, \forall j \in [m], \quad \left\langle \mathbf{r}, \mathbf{h}_e^j \right\rangle = 0 \iff \sum_{i \in \pi_{u,e}^{-1}(j)} \mathbf{r}(u(i)) = \sum_{i \in \pi_{v,e}^{-1}(j)} \mathbf{r}(v(i)). \tag{17}$$

The folding is done as follows.

1. Let $T = \bigcup_{e \in E, j \in [m]} \{\mathbf{h}_e^j\}$. Let $H \subset \mathbb{R}^{\mathcal{M}}$, where $H = \text{span}(T)$, and F be the subspace of $\mathbb{R}^{\mathcal{M}}$ orthogonal to H such that $\mathbb{R}^{\mathcal{M}} = F \oplus H$ and $F \perp H$.

2. Let $\{\lambda_i\}_{i=1}^{nk}$ be an orthonormal basis for $\mathbb{R}^{\mathcal{M}}$ such that $F = \text{span}(\{\lambda_i\}_{i=1}^{g})$ for some $g \leq nk$.

3. Write down all the points in the sets A and B in the basis $\{\lambda_i\}_{i=1}^{nk}$ and only consider the coordinates corresponding to the basis $\{\lambda_i\}_{i=1}^{g}$ of the g-dimensional space F. Ignoring the rest of the coordinates, obtain sets A' and B', which are essentially projections (with multiplicities) of sets A and B respectively onto the subspace F.

We label all the points in A' as '+', and all the points in B' as '$-$' and output these points as an instance of Intersection-Halfspace$_\ell$.

3.4 Analysis

3.4.1 YES Case

If the instance $\mathcal{L}$ of Smooth-Label-Cover(t, μ, ν, k, m) is a YES instance, then there is a labeling, say σ to the vertices in V that satisfies all the edges in E. We need to exhibit two halfspaces such that the points in A' lie inside their intersection while the points in B' lie outside their intersection, where A' and B' are the sets obtained through the reduction given above.

Let us consider the vector $\mathbf{r}^* \in \mathbb{R}^{\mathcal{M}}$, where $\mathbf{r}^*(v(\sigma(v))) = 1/\sqrt{n}$ for all $v \in V$, and all other coordinates are set to 0. So, $\mathbf{r}^*$ has exactly n non-zero coordinates, each set to $1/\sqrt{n}$. Clearly $\|\mathbf{r}^*\| = 1$. We prove the following lemma.

Lemma 3.4.1. *For every point* $\mathbf{y} \in A$, $|\langle \mathbf{r}^*, \mathbf{y} \rangle| < 1/\sqrt{n}$, *and for every point* $\mathbf{w} \in B$, $|\langle \mathbf{r}^*, \mathbf{w} \rangle| > 1/\sqrt{n}$.

Proof: Since $\mathbf{r}^*$ is a unit vector, for any unit vector $\mathbf{z}$, we have,

$$|\langle \mathbf{r}^*, \gamma\delta\mathbf{z} \rangle| \leq \gamma\delta = \delta\left(\frac{1}{100\sqrt{n}}\right). \tag{18}$$

Consider a point $\mathbf{y} \in A$, such that $\mathbf{y} = (1-\delta)\mathbf{x} + \gamma\delta\mathbf{z}$, where $\mathbf{x}$ is a $\{-1, 1\}$ vector in the coordinates $M(v)$ for some $v \in V$, and 0 on all other coordinates, and $\mathbf{z}$ is a unit vector. Now since $\mathbf{r}^*$ has a single non-zero coordinate set to $1/\sqrt{n}$ in each set $M(v)$, clearly $|\langle \mathbf{r}^*, \mathbf{x} \rangle| = 1/\sqrt{n}$ and therefore $|\langle \mathbf{r}^*, (1-\delta)\mathbf{x} \rangle| = (1-\delta)(1/\sqrt{n})$. Combining with (18), we get $|\langle \mathbf{r}^*, ((1-\delta)\mathbf{x} + \gamma\delta\mathbf{z}) \rangle| = |\langle \mathbf{r}^*, \mathbf{y} \rangle| < 1/\sqrt{n}$. Similarly, for any $\mathbf{w} \in B$, we obtain $|\langle \mathbf{r}^*, \mathbf{w} \rangle| > 1/\sqrt{n}$. ∎

Consider any edge $e \in E$ between two vertices u and v in V. Since $\mathbf{r}^*$ is $1/\sqrt{n}$ in exactly one coordinate $u(\sigma(u))$ in $M(u)$ and 0 on all others, for any $j \in [m]$, we have that the quantity $\sum_{i \in \pi_{u,e}^{-1}(j)} \mathbf{r}^*(u(i))$ is $1/\sqrt{n}$ iff $\sigma(u) \in \pi_{u,e}^{-1}(j)$ and 0 otherwise. And similarly, $\sum_{i \in \pi_{v,e}^{-1}(j)} \mathbf{r}^*(v(i))$ is $1/\sqrt{n}$ iff $\sigma(v) \in \pi_{v,e}^{-1}(j)$ and 0 otherwise. Since σ is a satisfying assignment, $\exists j' \in [m]$ such that $\sigma(u) \in \pi_{u,e}^{-1}(j')$ and $\sigma(v) \in \pi_{v,e}^{-1}(j')$. Therefore we have,

$$\sum_{i \in \pi_{u,e}^{-1}(j)} \mathbf{r}^*(u(i)) = \sum_{i \in \pi_{v,e}^{-1}(j)} \mathbf{r}^*(v(i)),$$
$$\forall u, v \in e,\ e \in E,\ \forall j \in [m]. \tag{19}$$

Combining the above with (17) we obtain,

$$\left\langle \mathbf{r}^*, \mathbf{h}_e^j \right\rangle = 0, \quad \forall e \in E,\ \forall j \in [m]. \tag{20}$$

Since H was defined to be the span of $\{\mathbf{h}_e^j\}_{e \in E, j \in [m]}$, Equation (20) implies that $\mathbf{r}^* \perp H$. Let $\bar{\mathbf{r}}^*$ be the projection of $\mathbf{r}^*$ onto F, where $F \perp H$ and $\mathbb{R}^{\mathcal{M}} = F \oplus H$. For any $\mathbf{y} \in \mathbb{R}^{\mathcal{M}}$, let $\bar{\mathbf{y}}$ be the projection of $\mathbf{y}$ onto F. Then, since $\mathbf{r}^*$ lies entirely in F we have $\langle \mathbf{r}^*, \mathbf{y} \rangle = \langle \bar{\mathbf{r}}^*, \bar{\mathbf{y}} \rangle$. Combined with Lemma 3.4.1 this implies that for every point $\bar{\mathbf{y}} \in A'$, $|\langle \bar{\mathbf{r}}^*, \bar{\mathbf{y}} \rangle| < 1/\sqrt{n}$ and for every point $\overline{\mathbf{w}} \in B'$, $|\langle \bar{\mathbf{r}}^*, \overline{\mathbf{w}} \rangle| > 1/\sqrt{n}$. Therefore the intersection of the two halfspaces in F, namely $\{\mathbf{y} \mid \langle \bar{\mathbf{r}}^*, \mathbf{y} \rangle \leq 1/\sqrt{n}\}$ and $\{\mathbf{y} \mid \langle \bar{\mathbf{r}}^*, \mathbf{y} \rangle \geq -1/\sqrt{n}\}$, classifies all the points in A' and B' correctly. Note that the intersection of halfspaces that we obtain is the region between two parallel hyperplanes.

3.4.2 NO Case

In this case, we assume that the instance $\mathcal{L}$ of SMOOTH-LABEL-COVER(t, μ, ν, k, m) has soundness η. For a contradiction, we assume that we have a function f of ℓ linear threshold functions in F that classifies $\frac{1}{2} + \varepsilon$ fraction of the points in A' and B' correctly, where A' is the set of '+' points and B' is the set of '−' points. We will henceforth refer to linear threshold functions as halfspaces, since they are exactly the same. Let the halfspaces on which f depends be given by the equations,

$$\langle \bar{\mathbf{r}}_i, \bar{\mathbf{y}} \rangle \leq c_i \qquad \text{for } i = 1, \ldots, \ell,$$

where $\bar{\mathbf{r}}_i \in F$ and $\|\bar{\mathbf{r}}_i\| = 1$ for all $i = 1, \ldots, \ell$. Let $\mathbf{r}_i$ be the vector in $\mathbb{R}^{\mathcal{M}}$ obtained from $\bar{\mathbf{r}}_i \in F$, by adding zeros on the coordinates corresponding to the basis of H, and rewriting it in the coordinates $\mathcal{M} = \bigcup_{v \in V} M(v)$. Clearly $\|\mathbf{r}_i\| = 1$ for $i = 1, \ldots, \ell$. Let f' be the function in $\mathbb{R}^{\mathcal{M}}$ given by the predicate of f applied on the halfspaces $\{\langle \mathbf{r}_i, \mathbf{y} \rangle \leq c_i\}$ for $i = 1, \ldots, \ell$, where the halfspace $\{\langle \bar{\mathbf{r}}_i, \bar{\mathbf{y}} \rangle \leq c_i\}$ in F is replaced by the halfspace $\{\langle \mathbf{r}_i, \mathbf{y} \rangle \leq c_i\}$ in $\mathbb{R}^{\mathcal{M}}$ in the predicate of f. Note that f' is exactly the function f applied on points in $\mathbb{R}^{\mathcal{M}}$ after projection onto F. We have the following simple lemma.

Lemma 3.4.2. *The function f' of the halfspaces $\{\langle \mathbf{r}_i, \mathbf{y} \rangle \leq c_i\}$ for $i = 1, \ldots, \ell$ classifies $\frac{1}{2} + \varepsilon$ fraction of the points in $A \cup B$ correctly, where a point in A is classified correctly if it lies inside the intersection and a point in B is classified correctly if it lies outside.*

Proof: We observe that since $\bar{\mathbf{r}}_i \in F$, if $\mathbf{y} \in \mathbb{R}^{\mathcal{M}}$ has a projection $\bar{\mathbf{y}} \in F$, then $\langle \mathbf{r}_i, \mathbf{y} \rangle = \langle \bar{\mathbf{r}}_i, \bar{\mathbf{y}} \rangle$. Now, since A' and B' are (multi)sets of points in F and are projections of the sets A and B respectively of points in $\mathbb{R}^{\mathcal{M}}$, the lemma follows. ∎

For the rest of the analysis, we will consider only the sets of points A and B and the halfspaces $\{\langle \mathbf{r}_i, \mathbf{y} \rangle \leq c_i\}$ for $i = 1, \ldots, \ell$ in $\mathbb{R}^{\mathcal{M}}$. For every vertex v, and every $\mathbf{x} \in s(v)$, there are $|R|$ pairs of points given by the sets $\{(1-\delta)\mathbf{x} + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x} + \delta\gamma\mathbf{z}\}$. In total there are $|V|2^k|R|$ such pairs, which partition the set $A \cup B$, where each pair has one point from A and one point from B. We say that a pair $\{\mathbf{y}_1, \mathbf{y}_2\}$ where $\mathbf{y}_1 \in A$ and $\mathbf{y}_2 \in B$ is correctly classified by f' if both $\mathbf{y}_1$ and $\mathbf{y}_2$ are correctly classified by f'. Since the function f' of

halfspaces $\{\langle \mathbf{r}_i, \mathbf{y}\rangle \leq c_i\}$ for $i = 1, \ldots, \ell$ correctly classifies $\frac{1}{2} + \varepsilon$ fraction of the points in $A \cup B$, it follows that it correctly classifies $\varepsilon/2$ fraction of pairs $\{(1-\delta)\mathbf{x}+\delta\gamma\mathbf{z}, (1+\delta)\mathbf{x}+\delta\gamma\mathbf{z}\}$. For a pair to be classified correctly by f', it must be separated by at least one of the ℓ halfspaces on which f' depends. Thus, there must be at least one out of the ℓ halfspaces that separates $\varepsilon/(2\ell)$ fraction of the pairs. Without loss of generality, we can assume that the halfspace $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$ separates $\varepsilon/(2\ell)$ fraction of the pairs. The rest of the analysis uses $\mathbf{r}_1$ to deduce a labeling to the vertices of $\mathcal{L}$ that satisfies a significant fraction of the edges.

Let $\varepsilon' = \varepsilon/(32\ell)$. By an averaging argument we have that for ε' fraction of the vertices $v \in V$, for ε' fraction of vectors $\mathbf{x} \in s(v)$, for $2\varepsilon'$ fraction of $\mathbf{z} \in R$, the pair $\{(1-\delta)\mathbf{x} + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x} + \delta\gamma\mathbf{z}\}$ is separated by the halfspace $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$. Call such vertices 'good'. Let u be one such vertex. We will show that the vector $\mathbf{r}_1$ must have a significant mass in the coordinates $M(u)$, and moreover that the mass is concentrated in a few of the coordinates in $M(u)$.

We know from above that for u, there is a vector $\mathbf{x}' \in s(u)$ such that for $2\varepsilon'$ fraction of $\mathbf{z} \in R$ the pair $\{(1-\delta)\mathbf{x}' + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x}' + \delta\gamma\mathbf{z}\}$ is separated by $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$. Let us this fix this particular $\mathbf{x}' \in s(u)$. Let us say that a pair is separated 'correctly' by the halfspace $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$ if the '+' point is inside the halfspace and the '−' point is outside, and otherwise we say that the pair is separated 'incorrectly'. Based on the above, for our choice of $\mathbf{x}' \in s(u)$ we have the following two cases.

Case 1. The halfspace $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$ separates 'correctly' the pair $\{(1-\delta)\mathbf{x}' + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x}' + \delta\gamma\mathbf{z}\}$ for ε' fraction of $\mathbf{z} \in R$. Let T be this set of 'good' vectors $\mathbf{z} \in R$, for which the corresponding pairs are separated correctly. And so $|T| = \varepsilon'|R|$. Since R satisfies the property stated in Lemma 3.3.1, there exist $\mathbf{z}', \mathbf{z}'' \in T$ such that,

$$|\langle \mathbf{r}_1, \mathbf{z}'\rangle - \langle \mathbf{r}_1, \mathbf{z}''\rangle| \geq \varepsilon'/(100\sqrt{nk}). \tag{21}$$

Moreover, since the pairs are separated 'correctly' we have that,

$$\langle \mathbf{r}_1, ((1-\delta)\mathbf{x}' + \gamma\delta\mathbf{z}')\rangle - c_1 \leq 0 \tag{22}$$

$$\langle \mathbf{r}_1, ((1+\delta)\mathbf{x}' + \gamma\delta\mathbf{z}')\rangle - c_1 \geq 0, \tag{23}$$

$$\langle \mathbf{r}_1, ((1-\delta)\mathbf{x}' + \gamma\delta\mathbf{z}'')\rangle - c_1 \leq 0 \tag{24}$$

$$\langle \mathbf{r}_1, ((1+\delta)\mathbf{x}' + \gamma\delta\mathbf{z}'')\rangle - c_1 \geq 0. \tag{25}$$

Subtracting Equation (22) from (25), and (24) from (23), we obtain,

$$2\delta \langle \mathbf{r}_1, \mathbf{x}'\rangle - \delta\gamma \langle \mathbf{r}_1, (\mathbf{z}' - \mathbf{z}'')\rangle \geq 0$$

$$2\delta \langle \mathbf{r}_1, \mathbf{x}'\rangle + \delta\gamma \langle \mathbf{r}_1, (\mathbf{z}' - \mathbf{z}'')\rangle \geq 0.$$

Combining the above with Equation (21) we get that $|2\delta \langle \mathbf{r}_1, \mathbf{x}'\rangle| \geq \gamma\delta\varepsilon'/(100\sqrt{nk})$. Substituting the value of γ and simplifying we have $|\langle \mathbf{r}_1, \mathbf{x}'\rangle| \geq \varepsilon'/(2\cdot 10^4 n\sqrt{k})$. Since $\mathbf{x}'$ takes values 1 or -1 on coordinates in $M(u)$ and is 0 on all other coordinates, this implies,

$$\sum_{i\in k} |\mathbf{r}_1(u(i))| \geq \frac{\varepsilon'}{2\cdot 10^4 n\sqrt{k}} \tag{26}$$

Case 2. In this case we have that the halfspace $\{\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1\}$ separates 'incorrectly' the pair $\{(1-\delta)\mathbf{x}' + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x}' + \delta\gamma\mathbf{z}\}$ for ε' fraction of $\mathbf{z} \in R$. The analysis continues along the same line as Case 1, taking the set T to be the set of 'good' vectors $\mathbf{z} \in R$, for which the corresponding pairs are separated 'incorrectly'. Since the pairs are now separated 'incorrectly', the inequalities (22), (23), (24) and (25) are reversed. We omit the rest of the argument which remains essentially the same as Case 1 and we obtain the same bound given by Equation (26).

The above analysis shows that the vector $\mathbf{r}_1$ has significant mass in the coordinates $M(u)$. To show it is concentrated in a small number of coordinates, we need the following lemma.

Lemma 3.4.3. *There is a set $Q \subseteq s(u)$, s.t. $|Q| \geq \varepsilon'|s(u)|$ and for every $\mathbf{x} \in Q$, $\langle \mathbf{r}_1, \mathbf{x}\rangle \in [c_1 - 2\delta\sqrt{k}, c_1 + 2\delta\sqrt{k}]$.*

Proof: We consider the set Q of points $\mathbf{x} \in s(u)$, such that a pair $\{(1-\delta)\mathbf{x} + \delta\gamma\mathbf{z}, (1+\delta)\mathbf{x} + \delta\gamma\mathbf{z}\}$ for some $\mathbf{z} \in R$ is separated by $\langle \mathbf{r}_1, \mathbf{y}\rangle \leq c_1$. Clearly, $|Q| \geq \varepsilon'|s(u)| = \varepsilon' 2^k$. Now, for

any given $\mathbf{x} \in s(u)$, all the points of the form $(1-\delta)\mathbf{x}+\delta\gamma\mathbf{z}$ and $(1+\delta)\mathbf{x}+\delta\gamma\mathbf{z}$ for any $\mathbf{z} \in R$ lie in a ball of radius $2\delta\sqrt{k}$ around $\mathbf{x}$. Therefore, for all $\mathbf{x} \in Q$, the hyperplane $\langle \mathbf{r}_1, \mathbf{y} \rangle = c_1$ passes at a perpendicular distance of at most $2\delta\sqrt{k}$ from $\mathbf{x}$. The lemma follows. ∎

The following is a well known lemma (see Lemma 7.3 of [37]). We state a version based on Lemma 3.5 proved in [10].

Lemma 3.4.4. *Let $X_1, \ldots, X_p$ be i.i.d $\{-1, 1\}$ valued Bernoulli random variables, with $\Pr[1] = \frac{1}{2}$, and let $\omega_1, \ldots, \omega_p$ be positive real numbers. Then there is a universal constant b such that, for any $c \in \mathbb{R}$ and $\zeta > 0$, if,*

$$\Pr\left[\sum_{i=1}^{p} \omega_i X_i \in [c-\zeta, c+\zeta]\right] \geq \frac{b}{p^{\frac{1}{2}}}$$

then, $\exists i \in [p]$ such that $\omega_i \leq \zeta$.

Let $X_1, \ldots, X_k$ be i.i.d $\{-1, 1\}$ valued Bernoulli random variables with $\Pr[1] = \frac{1}{2}$. For convenience, we let $\delta' = \delta\sqrt{k}$. Observe that Lemma 3.4.3 implies,

$$\Pr\left[\sum_{i=1}^{k} |\mathbf{r}_1(u(i))| X_i \in [c_1 - 2\delta', c_1 + 2\delta']\right] \geq \varepsilon'. \tag{27}$$

Suppose we apply Lemma 3.4.4 to the above. Then it gives us a coordinate in $M(u)$ such that on that coordinate $\mathbf{r}_1$ has very small mass. Removing that coordinate, we can again apply the lemma to the remaining coordinates and do this until a small number of coordinates remain. If we ensure that at each step we remove a coordinate from $M(u)$ on which $\mathbf{r}_1$ has small mass, then the total mass of the coordinates removed is small. Combining this with the lower bound given by Equation (26), this would imply that most of the mass of $\mathbf{r}_1$ in $M(u)$ is concentrated in a small number of coordinates. This would enable us to select a labeling for the vertex u from among those 'large' coordinates.

Therefore, we apply Lemma 3.4.4 iteratively, until the set of coordinates is of size at most b^2/ε'^2, in the following manner. Initialize $I_0 = [k]$.

1. At step j, we have a set of indices I_j of size $k - j$, with the following inequality satisfied,

$$\Pr\left[\sum_{i \in I_j} |\mathbf{r}_1(u(i))| X_i \in [c_1 - 2^{j+1}\delta', c_1 + 2^{j+1}\delta']\right] \geq \varepsilon'.$$

2. If $k - j < b^2/\varepsilon'^2$, then we stop and obtain a set $I^u = I_j$ of indices, such that $|I^u| < b^4/\varepsilon'^4$.

3. If $k - j \geq b^2/\varepsilon'^2$, then we apply Lemma 3.4.4 to obtain $i' \in I_j$ such that $|\mathbf{r}_1(u(i'))| \leq 2^{j+1}\delta'$. Now for any setting $x_i \in \{-1, 1\}$ of variables X_i for $i \in I_j$,

$$\sum_{i \in I_j} |\mathbf{r}_1(u(i))| x_i \in [c_1 - 2^{j+1}\delta', c_1 + 2^{j+1}\delta'] \Longrightarrow \sum_{i \in I_j \setminus \{i'\}} |\mathbf{r}_1(u(i))| x_i \in [c_1 - 2^{j+2}\delta', c_1 + 2^{j+2}\delta'].$$

Therefore we have,

$$\Pr\Big[\sum_{i \in I_j \setminus \{i'\}} |\mathbf{r}_1(u(i))| X_i \in [c_1 - 2^{j+2}\delta', c_1 + 2^{j+2}\delta'] \Big] \geq \varepsilon'.$$

So, we set $I_{j+1} = I_j \setminus \{i'\}$ and proceed to step $j + 1$.

At the jth step, an index corresponding to a coordinate of mass at most $2^{j+1}\delta'$ is removed. There are at most k steps for $j = 0, \ldots, k-1$. Therefore, the total mass of the coordinates removed is at most $2^{k+1}\delta'$. Combining this with (26) and with a small enough choice of δ, we have a set $I^u \subseteq [k]$ such that $|I^u| \leq b^2/(\varepsilon')^2$ and

$$\sum_{i \in I^u} |\mathbf{r}_1(u(i))| \quad \geq \frac{\varepsilon'}{2 \cdot 10^4 n\sqrt{k}} - 2^{k+1}\delta' \geq \quad \frac{\varepsilon'}{4 \cdot 10^4 n\sqrt{k}} \tag{28}$$

and,

$$\sum_{i \in [k] \setminus I^u} |\mathbf{r}_1(u(i))| \quad \leq \quad 2^{k+1}\delta' \quad \leq \quad \frac{\varepsilon'}{16 \cdot 10^4 n\sqrt{k}} \tag{29}$$

Since u was one of the ε' fraction of the vertices of V that are 'good', we can obtain such sets of indices I^v satisfying the above properties for all 'good' vertices v. Construct the labeling σ^* to these vertices by choosing a label for every 'good' vertex $v \in V$ randomly from I^v. From the properties of the instance $\mathcal{L}$ of Smooth-Label-Cover(t, μ, ν, k, m), if we choose $\nu \ll \varepsilon'$, then the set of 'good' vertices induces $(\varepsilon')^2/2$ fraction of edges in E. Let e be a random edge in E, and say e is between vertices v_1 and v_2 in V. Then with probability $(\varepsilon')^2/2$, both v_1 and v_2 are 'good'. Now, suppose that we have chosen $t \gg b^2/(\varepsilon')^2$, then

except with probability 2μ, π_{e,v_1} maps the elements of I^{v_1} to distinct elements $J^{v_1} \subseteq [m]$ and π_{e,v_2} maps the elements of I^{v_2} to distinct elements in $J^{v_2} \subseteq [m]$.

Suppose for a contradiction that J^{v_1} and J^{v_2} are disjoint. This implies that $\pi^{-1}_{e,v_2}(J^{v_1})$ and I^{v_2} are disjoint. Since $\mathbf{r}_1$ is orthogonal to the subspace H, from (17) we have that for every $j \in J^{v_1}$,

$$\sum_{i \in \pi^{-1}_{e,v_1}(j)} \mathbf{r}_1(v_1(i)) \quad = \quad \sum_{i \in \pi^{-1}_{e,v_2}(j)} \mathbf{r}_1(v_2(i))$$

and taking the absolute values and summing over all $j \in J^{v_1}$, we have,

$$\sum_{j \in J^{v_1}} \left| \sum_{i \in \pi^{-1}_{e,v_1}(j)} \mathbf{r}_1(v_1(i)) \right| = \sum_{j \in J^{v_1}} \left| \sum_{i \in \pi^{-1}_{e,v_2}(j)} \mathbf{r}_1(v_2(i)) \right| \tag{30}$$

Now, since π_{e,v_1} maps elements of I^{v_1} to distinct elements $J^{v_1} \subseteq [m]$,

$$\sum_{j \in J^{v_1}} \left| \sum_{i \in \pi^{-1}_{e,v_1}(j)} \mathbf{r}_1(v_1(i)) \right| \geq \sum_{i \in I^{v_1}} |\mathbf{r}_1(v_1(i))| - \sum_{i \in [k] \setminus I^{v_1}} |\mathbf{r}_1(v_1(i))| \,. \tag{31}$$

From Equation (30) and the above we have,

$$\begin{aligned} \sum_{j \in J^{v_1}} \left| \sum_{i \in \pi^{-1}_{e,v_2}(j)} \mathbf{r}_1(v_2(i)) \right| &\geq \sum_{i \in I^{v_1}} |\mathbf{r}_1(v_1(i))| - \sum_{i \in [k] \setminus I^{v_1}} |\mathbf{r}_1(v_1(i))| \\ &\geq \frac{\varepsilon'}{4 \cdot 10^4 n\sqrt{k}} - \frac{\varepsilon'}{16 \cdot 10^4 n\sqrt{k}} \\ &= \frac{3\varepsilon'}{16 \cdot 10^4 n\sqrt{k}} \end{aligned} \tag{32}$$

where we used (28) and (29) applied to v_1. But since, $\pi^{-1}_{e,v_2}(J^{v_1}) \subseteq [k] \setminus I^{v_2}$, Equation (32) is a contradiction to Equation (29) applied to v_2. Therefore, J^{v_1} and J^{v_2} are not disjoint. So, with probability $1/(|I^{v_1}||I^{v_2}|) = (\varepsilon')^4/b^4$, the labeling σ^* satisfies the edge e. Combining everything, we obtain that there is a labeling to the vertices of V that satisfies $((\varepsilon')^2/2 - 2\mu)((\varepsilon')^4/b^4)$ fraction of the edges in E. By choosing the smoothness parameter μ and the soundness parameter η of the instance $\mathcal{L}$ to be arbitrarily small, we obtain a contradiction. Thus, if the instance $\mathcal{L}$ of Smooth-Label-Cover(t, μ, ν, k, m) is a NO instance, then with high probability, there is no function of up to ℓ halfspaces that correctly classifies $\frac{1}{2} + \varepsilon$ fraction of the points in $A' \cup B'$. This, along with the analysis of the YES case proves Theorem 3.2.2, and hence Theorem 1.6.4.

3.5 Sampling from the Unit Sphere

In this section we prove Lemma 3.3.1. First we need some definitions.

Definition 3.5.1. *A range space is a pair $(X, \mathcal{F})$, where X is a set and $\mathcal{F}$ is a family of subsets of X, i.e. $\mathcal{F} \subseteq 2^X$.*

Definition 3.5.2. *For any set $A \subseteq X$, define $P_{\mathcal{F}}(A)$ the projection of $\mathcal{F}$ onto A, as $P_{\mathcal{F}}(A) := \{F \cap A : F \in \mathcal{F}\}$.*

Definition 3.5.3. *We say that a set $A \subseteq X$ is shattered by $(X, \mathcal{F})$ if $P_{\mathcal{F}}(A) = 2^A$.*

The VC dimension of a range space is defined as follows.

Definition 3.5.4. *The VC dimension of $(X, \mathcal{F})$ is the cardinality of the maximum set it shatters, i.e. VC dim $= \sup\{|A| : A \text{ is shattered }\}$. It may be infinite.*

We use the following theorem regarding sampling from range spaces of bounded VC dimension [80].

Theorem 3.5.1. *Let $(X, \mathcal{F})$ a range space of VC dimension d, and let ϕ be a uniform measure on X. There is a universal constant C_{VC} such that with probability at least $1 - \delta$, a random set $S \subseteq X$ of size,*

$$C_{VC}\left(\frac{d}{\tau^2}\log\left(\frac{d}{\tau}\right) + \frac{1}{\tau^2}\log\left(\frac{1}{\delta}\right)\right)$$

satisfies,

$$\left|\frac{|S \cap F|}{|S|} - \phi(F)\right| \leq \tau,$$

for all $F \in \mathcal{F}$.

Let $N = nk$, let $\mathbb{S}^{N-1}$ be the unit sphere in N dimensions, and let ϕ be a uniform measure over $\mathbb{S}^{N-1}$. Define $P(\mathbf{r}, [a, b]) := \{\mathbf{z} \in \mathbb{S}^{N-1} | \langle \mathbf{r}, \mathbf{z} \rangle \in [a, b]\}$. The set $P(\mathbf{r}, [a, b])$ is exactly the set of unit vectors whose dot product with $\mathbf{r}$ lies in the interval $[a, b]$. Let $\mathcal{P} := \{P(\mathbf{r}, [a, b]) \mid \|\mathbf{r}\| = 1,\ b - a = \varepsilon'/(100\sqrt{N})\}$. By Stirling's approximation, we have that for large enough N, the surface area of the $\mathbb{S}^{N-1}$ is at least $1/\sqrt{N}$ times the surface area of $\mathbb{S}^{N-2}$. As a result the following fact is easy to derive.

Fact 3.5.2. *For large enough N, for any $P \in \mathcal{P}$, $\phi(P) \leq \varepsilon'/10$.*

Moreover, we observe that every element $P \in \mathcal{P}$ is a set of points in $\mathbb{S}^{N-1}$ that lie in an intersection of two halfspaces. Since, in $\mathbb{R}^N$, the VC dimension of the class of all N dimensional halfspaces is $N+1$, the VC dimension of the class of N dimensional halfspaces for the set $\mathbb{S}^{N-1}$ is at most $N+1$. Using this we have the following bound which we state without proof.

Lemma 3.5.3. *The VC dimension of the range space $(\mathbb{S}^{N-1}, \mathcal{P})$ is at most $4(N+1)\log(2(N+1))$.*

Now if R is a set of N^2, random unit vectors from $\mathbb{S}^{N-1}$, then for large enough N, Theorem 3.5.1 along with the above lemma implies that with probability at least $1 - 1/n$,

$$\left| \frac{|R \cap P|}{|R|} - \phi(P) \right| \leq \varepsilon'/10,$$

for any $P \in \mathcal{P}$. The above, coupled with Fact 3.5.2 implies that with probability $1 - 1/n$ over the choice of R,

$$\frac{|R \cap P|}{|R|} \leq \varepsilon'/5,$$

for all $P \in \mathcal{P}$. In other words, with probability $1 - 1/n$ over the choice of R, for any unit vector $\mathbf{r}$, at most $\varepsilon'/5$ fraction of points in R are contained in the set $\{\mathbf{z} \mid \langle \mathbf{r}, \mathbf{z} \rangle \in [a, b]\}$ for any a, b s.t. $b - a = \varepsilon'/(100\sqrt{N})$. This implies that with probability $1 - 1/n$ over the choice of R, for any set $T \subseteq R$, such that $|T| = \varepsilon'|R|$, for any unit vector $\mathbf{r}$, there exist $\mathbf{z}', \mathbf{z}'' \in T$, such that $|\langle \mathbf{r}, \mathbf{z}' \rangle - \langle \mathbf{r}, \mathbf{z}'' \rangle| \geq \varepsilon'/(100\sqrt{N})$. This proves Lemma 3.3.1.

3.6 Inapproximability of Smooth-Label-Cover

In this section we prove Theorem 3.2.1. Let us first define the 'smooth' version of the bipartite Label-Cover problem.

Definition 3.6.1. *An instance of* Smooth-Bipartite-Label-Cover(k, m, T) *consists of a bipartite graph $G(U, V, E)$, where the vertices in U have the same degree, and a set of projections $\pi^{vu} : [k] \mapsto [m]$ for all $\{u, v\} \in E$ such that $u \in U, v \in V$. A labeling σ to the*

vertices in G satisfies and edge $e = \{u, v\}$ s.t. $u \in U, v \in V$ iff $\pi^{vu}(\sigma(v)) = \sigma(u)$. Moreover, for any vertex $v \in V$ for any $i, j \in [k], i \neq j$,

$$\Pr_{e=\{u,v\}\in E}[\pi^{vu}(i) = \pi^{vu}(j)] \leq \frac{1}{T}, \tag{33}$$

where the probability is taken over a random edge incident on v.

The following theorem was proved in [50], using the PCP Theorem [6, 8] and Raz's Parallel Repetition Theorem [71].

Theorem 3.6.1. *For any constant $\delta > 0$, for any constant $T > 0$, there exist k and m such that given an instance $\mathcal{L}'$ of* Smooth-Bipartite-Label-Cover(k, m, T) *it is* NP*-hard to distinguish between the following two cases,*

- YES Case/Completeness*: There is a labeling to the vertices of $\mathcal{L}'$ that satisfies all the edges.*
- NO Case/Soundness*: No labeling to the vertices of $\mathcal{L}'$ satisfies more than δ fraction of the edges.*

The construction of an instance $\mathcal{L}$ of Smooth-Label-Cover(t, μ, ν, k, m) is as follows. We start with an instance $\mathcal{L}'$ of Smooth-Bipartite-Label-Cover(k, m, T) where we will fix T later. The vertex set of $\mathcal{L}$ is the V side of $\mathcal{L}'$. An edge of $\mathcal{L}$ is constructed as follows: select a vertex u from U and for every two neighbors v_1 and v_2 of u in $\mathcal{L}'$, add an edge e between them in $\mathcal{L}$. Set $\pi_{e,v_1} = \pi^{v_1 u}$ and $\pi_{e,v_2} = \pi^{v_2 u}$ in E. Let $E(u)$ be the set of such edges added in $\mathcal{L}$ corresponding to a vertex $u \in U$. Note that we are constructing a multigraph, since two vertices v_1 and v_2 in V might share two different neighbors in U, in which case there will be multiple edges between v_1 and v_2 in $\mathcal{L}$. Clearly the sets $E(u)$ for $u \in U$ are a partition of edges in $\mathcal{L}$, and since U side is regular, the sets $E(u)$ are of equal size. Essentially, we are adding a clique of edges $E(u)$ corresponding to u on its neighborhood $N(u) \subseteq V$ for every $u \in U$. Let v be a vertex in V and let $S \subseteq [k]$ be a set of size t, then applying Equation (33) to all pairs in S and taking union bound, we have,

$$\Pr_{e:v\in e}[\exists x, y \in S, x \neq y : \pi_{e,v}(x) = \pi_{e,v}(y)] \leq \frac{t^2}{T}$$

where the probability is over the edges incident on v in $\mathcal{L}$. Note that we have used the fact the vertices in U have the same degree. Now, taking T to be large enough, we can reduce this probability to at most μ. To verify the second property, let V' be a subset of V such that $|V'| = \xi|V|$, for some $\xi > 0$. Now, consider a vertex u in U, and let p_u be the probability that a random neighbor of u falls in V'. From the proof of Theorem 3.6.1 one can see that vertices on U side have the same degree, say d which can be increased to any arbitrary constant by parallel repetition. Therefore, $\mathrm{E}_u[p_u] = \xi$. Moreover, the fraction of edges in $\mathcal{L}$ that lies inside V' is the probability for a random $u \in U$, a random pair of its neighbors lies in V'. For a particular u this is $p_u^2 - 1/d$, where $1/d$ is the probability of selecting the same vertex twice out of d neighbors of u. Hence, we have that the fraction of edges induced by V' in $\mathcal{L}$ is $\mathrm{E}_u[p_u^2 - 1/d] \geq (\mathrm{E}_u[p_u])^2 - 1/d = \xi^2 - 1/d$. This fraction is at least $\xi^2/2$ if $\xi \geq \sqrt{\frac{2}{d}}$, and so we are done by taking $\nu = \sqrt{\frac{2}{d}}$ which can be made arbitrarily small by taking d to be large enough.

Now, if $\mathcal{L}'$ was a YES instance, then there is a labeling σ to vertices $U \cup V$ that satisfies all the edges of $\mathcal{L}'$. This implies,

$$\pi^{v_1 u}(\sigma(v_1)) = \sigma(u) = \pi^{v_2 u}(\sigma(u)),$$

for all edges $e_1 = \{u, v_1\}, e_2 = \{u, v_2\}$ of $\mathcal{L}'$, $u \in U, v_1, v_2 \in N(u) \subseteq V$, where $N(u)$ is the neighborhood of $u \in U$ in $\mathcal{L}'$. Consider the edge $e \in E(u)$ between v_1 and v_2 in $\mathcal{L}$. Clearly, $\pi_{e,v_1}(\sigma(v_1)) = \pi_{e,v_2}(\sigma(v_2))$. Therefore, the labeling σ restricted to V satisfies all the edges of $\mathcal{L}$.

Now consider a labeling σ' to V that satisfies ε fraction of the edges in $\mathcal{L}$. Consider any vertex $u \in U$. For $j \in [m]$, let $S_u^j \subseteq N(u)$ the set of vertices $v \in N(u)$ such that $\pi^{vu}(\sigma'(v)) = j$. It can be seen that the sets S_u^j ($j \in [m]$) form a partition of $N(u)$ and the disjoint union of edges (corresponding to u) induced by each S_u^j in $E(u)$ is exactly the subset of edges of $E(u)$ that are satisfied by σ'. Let $l_u = \mathrm{argmax}_j |S_u^j|$ for each $u \in U$. Observe that seen that any subset S of $N(u)$ containing c ($c < 1$) fraction of vertices of $N(u)$ induces in $E(u)$ at most c^2 fraction of the total edges of $E(u)$. Suppose σ' satisfies ε_u fraction of the edges of $E(u)$, then a simple argument shows that $S_u^{l_u}$ must contain at

least ε_u fraction of vertices in $N(u)$. Now σ' satisfies ε fraction of all the edges of $\mathcal{L}$, and since the U side is regular in $\mathcal{L}'$, we have that $\mathrm{E}_u[|S_u^{l_u}|/|N(u)|] \geq \mathrm{E}_u[\varepsilon_u] \geq \varepsilon$. Therefore, by extending the labeling σ' to U by setting $\sigma'(u) = l_u$ for $u \in U$, we can satisfy the edges of $\mathcal{L}'$ between vertices of $S_u^{l_u}$ and u for all $u \in U$. This would satisfy ε fraction of the edges in $\mathcal{L}'$. So, if the instance of $\mathcal{L}'$ is a NO instance with soundness η then there is no labeling to the vertices of $\mathcal{L}$ which satisfies more than η fraction of the edges of $\mathcal{L}$. This completes the proof of Theorem 3.2.1.

3.7 Conclusion

We proved a tight hardness result for learning intersection of two halfspaces using functions of up to ℓ linear threshold functions (halfspaces) for any constant ℓ. An interesting open question is whether a similar hardness result holds for learning intersection of halfspaces by more general classes of hypotheses such as (functions of) low degree polynomials. As noted in the remark in Section 3.1 our reduction does not extend even to degree 2 polynomials.

In addition, as alluded to in Section 2.8, a limitation of our result is that the parameter ε in the $\frac{1}{2}+\varepsilon$ inapproximability factor is only an arbitrarily small constant. It is an important open problem to obtain a similar result with ε being an inverse *polynomial* in the dimension of the instance.

CHAPTER IV

HARDNESS OF RECONSTRUCTING MULTIVARIATE POLYNOMIALS

In this chapter we prove Theorem 1.6.5 on the hardness of reconstructing multivariate polynomials over $\mathbb{F}[2]$ in the presence of adversarial noise. We start by giving and overview of the proof in the following section.

4.1 Overview

Let us recall from Section 1.3 the main result of this chapter.

Theorem. (1.6.5 restated) *For any constants $\varepsilon, \delta > 0$ and positive integer d, given an instance of* PolyRec(d) *over $\mathbb{F}[2]$, with the guarantee that there is a linear polynomial satisfying $P(\mathbf{x}^i) = \zeta^i$ for $1 - \varepsilon$ fraction of the points, it is* NP*-hard to find a polynomial P of degree at most d that satisfies $P(\mathbf{x}^i) = \zeta^i$ for at most $1 - 2^{-d} + \delta$ fraction of the points.*

The reduction employed in the proof of the above theorem is in many ways similar to the one given in Chapter 3, especially in exploiting the fact that the set of examples lies in a vector space over a field, in this case $\mathbb{F}[2]$. As mentioned before, Theorem 1.6.5 is also a hardness result for PAC-learning in the presence of adversarial noise (or agnostic learning) a linear function by a degree d multivariate polynomial over $\mathbb{F}[2]$.

The main technical contribution of this work is to apply the machinery of PCPs to the polynomial reconstruction problem. Our result is proved by a reduction from (a variant of) Label-Cover. For simplicity we shall use Label-Cover in a generic manner to refer to the variant of Label-Cover as well, and its use shall be clear from the context. However, the fact that polynomial reconstruction for $d \geq 2$ is not a CSP in the usual sense means that there are several obstacles to overcome. To do so, we introduce some new primitives such as Dictatorship Testing for Polynomials and Consistency Testing via Folding.

Dictatorship Testing for low-degree Polynomials: Like most reductions from Label-Cover, our first goal is to give a *dictatorship test* for low-degree polynomials, using constraints of the form $\langle \mathbf{x}, \zeta \rangle$ for $\mathbf{x} \in \mathbb{F}[2]^k$. Our goal is that the polynomials X_i for $i \in [k]$, which we think of as the dictatorship of i will pass this test with good probability. On the other hand, for every polynomial $P(X_1, \ldots, X_k)$ of degree d which passes the test with good probability, we wish to *decode* it to a dictatorship. While this may not always be possible, we will settle for a list of indices from $[k]$ whose length is constant (independent of k).

We propose the following test: we sample a random vector $\eta \in \mathbb{F}[2]^k$ where each η_i is 1 with probability ε, and check that $P(\eta) = 0$. In other words, polynomials passing the test must be noise-stable at $\mathbf{0}^k$. Dictatorships pass this test with probability $1 - \varepsilon$. But there are several polynomials that will do well, for instance $X_1 X_2$ will pass with probability $1 - \varepsilon^2$. While this polynomial is *close* to a dictatorship, the polynomial $X_1(X_2 + \ldots + X_k)$ which depends on all k variables passes w.p. close to $1 - \frac{\varepsilon}{2}$. Indeed, any polynomial which can be written as

$$P(X_1, \ldots, X_n) = X_1 P_1(X_1, \ldots, X_n) + \ldots + X_c P_c(X_1, \ldots, X_n)$$

where the P_is are arbitrary polynomials of degree $d - 1$ will pass the test w.p $1 - c\varepsilon$. If we view the set of monomials as a hypergraph on $[k]$, polynomials whose hypergraphs have small vertex covers will be noise stable at $\mathbf{0}^k$. We will use this as our notion of being *close* to a dictatorship. We prove an inverse theorem: if $P(X_1, \ldots, X_k)$ passes our test with good probability, the corresponding hypergraph must have a small maximum matching and hence a small vertex cover. We view this as a list-decoding of $P(X_1, \ldots, X_k)$.

It is unclear why this decoding should be of any use: indeed running the decoding a second time on the same hypergraph might produce a different matching. Note however that the vertex sets of any two maximal matchings must have some intersection. Indeed, the usefulness of this decoding procedure stems from the fact that given any $d + 1$ vertex covers in a d-regular hypergraph, some two will intersect.

It is interesting to contrast this dictatorship test with Fourier based dictatorship testing [38, 48]. In those tests, one is allowed to query the function being tested in two or more

points, but in our setting we are allowed just *one* query. What makes this possible however is the promise that the function being tested is a low-degree polynomial, as opposed to an arbitrary Boolean function. In a departure from Fourier based dictatorship testing, our analysis uses only basic facts about polynomials. However, giving a test with better might require new algebraic techniques.

Consistency Testing via Folding: Our strategy for reducing from Label-Cover is the following: to each vertex v in the Label-Cover instance, we assign variables $X_1^v, \ldots, X_k^v$ where k is the number of labels possible. In the Yes case, if the labeling of vertices is given by $l : V \to [k]$, then we want the polynomial $\sum X^v_{l(v)}$ to satisfy most of the constraints. Further, given any polynomial Q that satisfies sufficiently many constraints, we want to be able to decode it to a label for each vertex. To assign a label for vertex v, we consider the restriction of Q to the variables $X_1^v, \ldots, X_k^v$ obtained by setting the other variables to 0, which we denote by $Q(X^v)$. We then run the decoding procedure for the dictatorship test on it and pick a random label from the list. Our hope is that this will assign labels in a way that satisfies a constant fraction of the Label-Cover constraints.

The next gadget we need is a way of testing whether two vertices have been assigned consistent labels. For this, let us consider a toy problem where there are just two vertices and we want to test if they are assigned the same label. Following the outline above, we associate them with variables $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$ respectively. We want a test that passes the polynomials $X_i + Y_i$. Further, we want to assign labels to each vertex based on $U(X_1, \ldots, X_k) = Q(X_1, \ldots, X_k, \mathbf{0}^k)$ and $V(Y_1, \ldots, Y_k) = Q(\mathbf{0}^k, Y_1, \ldots, Y_k)$ respectively. If Q passes our test, these labels should be the same with constant probability (independent of k). We can run the dictatorship test on each U using vectors of the form $(\eta, \mathbf{0}^k)$ and similarly on V . Assuming they pass these tests, we want to check that they are identical polynomials after setting $X_i = Y_i$. The obvious approach is to take $\mathbf{r} \xleftarrow{R} \mathbb{F}[2]^k$ (i.e. $\mathbf{r}$ is sampled uniformly at random from $\mathbb{F}[2]^t$) and check that $Q(\mathbf{r}, \mathbf{r}) = 0$. But in fact this will not do, since we have no control on monomials of the form X_iY_j. Indeed, for any choice of restrictions U and V, one can adjust the coefficients of the X_iY_j terms so that the polynomial Q satisfies $Q(\mathbf{r}, \mathbf{r}) = 0$. This strongly suggests that a different approach is

necessary to enforce consistency.

Our solution is to enforce the consistency constraints via what we call global folding. Let us write the vector $(x_1, \ldots, x_k, y_1, \ldots, y_k) \in \mathbb{F}[2]^{2k}$ in a different basis as $(x_1 + y_1, \ldots, x_k + y_k, y_1, \ldots, y_k)$. Observe that in this basis, the polynomials $X_i + Y_i$ that pass the test only depend on the first k co-ordinates. We will enforce this condition on every polynomial. In place of the point-value pair $\langle(\mathbf{x}, \mathbf{y}), \zeta)\rangle$, we add the point-value pair $\langle(x_1 + y_1, \ldots, x_k + y_k), \zeta)\rangle$. Clearly, this does not hurt the completeness of the test. However, one could hope for better soundness, since we have restricted the space of polynomials from all polynomials in X_is and Y_js to those that only depend on $X_i + Y_i$. Equivalently, we are forcing the adversary to pick a polynomial that is constant on cosets of the subspace H defined by $X_i + Y_i = 0$. To analyze the probability that some polynomial P of k variables passes this new test, we *unfold* it and write it as $Q(X_1, \ldots, X_k, Y_1, \ldots, Y_k) = P(X_1 + Y_1, \ldots, X_k + Y_k)$. Note that this enforces the constraint that mapping X_i to Y_i sends U to V. Thus in fact, if P passes the dictatorship tests, then our decoding will assign the same labels to u and v with some probability.

Similarly, we enforce all the Label-Cover constraints via a suitable folding. If a solution to the Label-Cover instance exists, it will give a linear polynomial that lies in a low dimensional subspace of all linear functions on $\mathbb{F}[2]^{nk}$. This sub-space is defined by linear equations that encode the constraints of the Label-Cover instance. We identify this sub-space and perform the dictatorship test for every vertex after projecting points onto it. Assume that some polynomial P in this low dimensional subspace passes our tests with good probability. To decode P, we unfold it to a polynomial Q in nk dimensions. The polynomial Q has some nice symmetry properties which encode the constraints of the label-cover instance. We exploit these symmetries to show that our decoding procedure will find a good solution to the Label-Cover instance. The novelty of our approach is that the Label-Cover constraints are enforced via the folding and unfolding, and not through explicit consistency tests.

This is an idealized view of our reduction, which brushes over several technical issues. The constraints that we must enforce are more complicated than equality constraints (or

even permutations), they are defined in terms of projection maps. For technical reasons, we use a hypergraph version of Label-Cover, as opposed to the usual bipartite graph version. Also, we need to ensure that the polynomials passing our dictatorship tests are not 0, this is done by another kind of folding which we call local folding. Readers familiar with Håstad's PCP will note the similarity between the folding used there and local folding.

4.2 Preliminaries

We formally define the problem PolyRec(d) that we study in this chapter.

Definition 4.2.1. *The Polynomial Reconstruction problem* PolyRec(d) *for multivariate polynomials in n variables over $\mathbb{F}[2]$ of degree at most some constant d, is as follows. The input to is a (multi-)set of point-value pairs $\{\mathbf{x}^i, \zeta^i)\}_{i=1}^m$ where $\mathbf{x}^i \in \mathbb{F}[2]^n$ and $\zeta^i \in \mathbb{F}[2]$ and a degree bound d. The goal is to find the multivariate polynomial $P(X_1, \ldots, X_n)$ of degree at most d that satisfies $P(\mathbf{x}^i) = \zeta^i)$ for most points $\mathbf{x}^i$.*

Our reduction follows from a variant the standard Label-Cover problem which is defined formally as follows.

Definition 4.2.2. *For any $d, t, k \in \mathbb{Z}^+$, $(k \geq t)$ an instance of* Label-Cover$[d+1](t, k)$ *consists of a $d+1$-regular hypergraph (V, E) with vertex set $V = \{v_i\}_{i=1}^n$ and an edge set $E = \{e_j\}_{j=1}^m$, where $|e_j| = d+1$. The hypergraph is connected, and any $S \subset V$ of size δn induces a constant δ^{d+1} fraction of edges. Every vertex in V is to be assigned a label $l(v) \in [k]$. Every hyperedge $e = (v_1^e, \ldots, v_{d+1}^e)$ is associated with a $d+1$-tuple of projection functions $\{\pi_i\}_{i=1}^{d+1}$ where $\pi_i : [k] \to [t]$ and $t < k$. A vertex labeling "strongly satisfies" edge e if $\pi_i(l(v_i^e)) = \pi_j(l(v_j^e))$ for every $v_i^e, v_j^e \in e$. A vertex labeling "weakly satisfies" edge e if $\pi_i(l(v_i^e)) = \pi_j(l(v_j^e))$ for some pair of distinct vertices $v_i^e, v_j^e \in e$. The goal is to find a labeling that satisfies the maximum number of edges.*

This is a slightly non-standard hypergraph version of Label-Cover. A similar kind of acceptance predicate is used by Feige in proving the hardness of Set-Cover [27]. The only reason we cannot use his result directly is because we need to condition that large subsets

of vertices induce many edges. The following theorem is proved using a simple reduction from the standard Label-Cover. We give a proof in Section 4.5 for completeness.

Theorem 4.2.1. *For any $\beta > 0$, there exist constants t and k such that given an instance $\mathcal{L}$ of* Label-Cover$[d+1](t,k)$*, it is* NP*-hard to distinguish between the following cases:*

- YES Case*: There is some vertex labeling l that strongly satisfies every edge of $\mathcal{L}$.*
- NO Case*: There is no vertex labeling that weakly satisfies β fraction of the edges of $\mathcal{L}$.*

The following theorem is proved in Section 4.4, and it along with Theorem 4.2.1 proves Theorem 1.6.5.

Theorem 4.2.2. *For every $\varepsilon, \delta > 0$ and $d \in \mathbb{Z}^+$, there is a reduction from an instance $\mathcal{L}$ of* Label-Cover$[d+1](t,k)$*, for appropriately chosen t and k, to an instance $\mathcal{I}$ of* PolyRec(d) *such that,*

- YES Case*: If $\mathcal{L}$ is a YES instance then there is a linear polynomial satisfying $1-\varepsilon$ fraction of the point-value pairs of $\mathcal{I}$.*
- NO Case*: If $\mathcal{L}$ is a NO instance then every polynomial of degree at most d satisfies at most $1 - 2^{-d} + \delta$ fraction of the point-value pairs of $\mathcal{I}$.*

4.3 Primitives for Testing Polynomials

In this section we describe the primitives we require in the reduction for testing polynomials. We start with the most basic testing procedures and work our way towards the more complicated ones that are eventually utilized in the reduction.

4.3.1 Dictatorship Testing for Low-Degree Polynomials

Linear polynomials are polynomials of degree 1 with no constant. By degree d multivariate polynomials, we mean all polynomials of degree at most d. In particular it includes linear polynomials. Over $\mathbb{F}[2]$ we assume that all polynomials are multilinear. Let $\mathbf{0}^k$ and $\mathbf{1}^k$

denote the all 0s and all 1s vector respectively. We use $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$ to denote sampling η from the ε-biased distribution, where each $\eta_i = 1$ independently w.p. ε. We will use $\eta \xleftarrow{R} \mathbb{F}[2]^k$ to denote sampling from the uniform distribution.

We analyze the following test on polynomials $P(X_1, \ldots, X_k)$ of degree at most d:

Algorithm 4.3.1. BASIC DICTATORSHIP TEST:

1. Pick $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$ and test if $P(\eta) = 0$.

Note that the zero polynomial passes the present test with probability 1; later we will modify the test to ensure that the polynomial in non-zero. We use the following fact about low-degree polynomials:

Fact 4.3.1. *Let $P(X_1, \ldots, X_k)$ be a non-zero polynomial of degree d over $\mathbb{F}[2]$. Then*

$$\Pr_{\eta \xleftarrow{R} \mathbb{F}[2]^k}[P(\eta) = 0] \;\leq\; 1 - 2^{-d}.$$

Given a polynomial $P(X_1, \ldots, X_k)$, we will associate it with a hypergraph $Hyp(P)$, with vertex set is $[k]$ and edge set E. E contains the hyperedge $e \subset [k]$ if the monomial $\prod_{i \in e} X_i$ is present in $Hyp(P)$. The degree bound of d implies that $|e| \leq d$. If we denote the constant term by $c \in \{0, 1\}$, then $P(X_1, \ldots, X_k) = \sum_{e \in E} \prod_{i \in e} X_i + c$. A matching in a hypergraph is a set of independent edges (with no common vertices). It is easy to see that taking all the vertices in a maximal matching gives a vertex cover for the hypergraph.

Theorem 4.3.2. *Let $P(X_1, \ldots, X_k)$ be a degree d polynomial over $\mathbb{F}[2]$ that passes the Basic Dictatorship Test with probability $1 - 2^{-d} + \delta$ for some $\delta > 0$. Then the largest matching in the hypergraph $Hyp(P)$ is of size $\frac{C}{(2\varepsilon)^d}$ where C depends only on δ. Further the constant term c in $P(X_1, \ldots, X_k)$ is 0.*

Proof: Rather than setting each X_i to 1 with probability ε, we will do a two-step sampling procedure, which will have the same effect:

1. Set every variable X_i to 0 independently with probability $1 - 2\varepsilon$.

2. Independently set each remaining variable to a random $\mathbb{F}[2]$ value.

It is clear that this induces the ε-biased distribution on η. Let $S \subset [k]$ be the set of indices corresponding to variables that are not set to 0 in step 1. Let X^S denote the set of these variables. The resulting polynomial $P'(X^S)$ consists of the hypergraph induced by the vertex set S. Also

$$\Pr_{\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k}[P(\eta) = 1] = \Pr_{\eta' \xleftarrow{R} \mathbb{F}[2]^{|S|}}[P'(\eta') = 1]$$

If $P'(X^S)$ is non-zero, then since it has degree at most d, $\Pr[P'(\eta') = 1] \geq 2^{-d}$. Now if $c = 1$, then P' also has the constant term 1, hence it is a non-zero polynomial, so $\Pr[P(\eta) = 1] = \Pr[P'(\eta') = 1] \geq 2^{-d}$, which is a contradiction.

Now assume that the hypergraph $Hyp(P)$ contains a matching M of size $|M| \geq \frac{C}{(2\varepsilon)^d}$ where the constant C will be fixed later. For each hyperedge $e \in M$, the probability that all its vertices are chosen to be in S is $(2\varepsilon)^{|e|}$. Also, since M is a matching, these events are independent for various edges. Thus the probability that none of these edges occurs in the hypergraph induced by S is bounded by

$$\prod_{e \in M} (1 - (2\varepsilon)^{|e|}) \;\leq\; (1 - (2\varepsilon)^d)^{\frac{C}{(2\varepsilon)^d}} \;<\; e^{-C}.$$

Hence, with probability $1 - e^{-C}$, the subgraph induced by S is non-empty. Conditioned on this event, $P'(X^S)$ is a non-zero polynomial of degree at most d, hence $P'(\eta') = 1$ with probability at least 2^{-d}. Thus

$$\Pr[P(\eta) = 1] \geq (1 - e^{-C}) \cdot 2^{-d}$$

For sufficiently large C, this contradicts the fact that $\Pr[P(\eta) = 1] \leq 2^{-d} - \delta$. ∎

Theorem 4.3.2 suggests the following decoding procedure:

Algorithm 4.3.2. Decoding Procedure for the Basic Dictatorship Test:

1. Pick a maximal matching M in $Hyp(P)$.
2. Output a list L of all vertices in this matching.

Clearly the set L is a small vertex cover for $Hyp(P)$. The usefulness of this decoding procedure is because of the following simple lemma.

Lemma 4.3.3. *Let $Hyp(P)$ be a non-empty hypergraph with some edge of size d. Let $L_1, \ldots, L_{d+1}$ be $d+1$ vertex covers for $Hyp(P)$. Then some pair L_i, L_j where $i \neq j$ has a non-empty intersection.*

If all the vertex covers are obtained by taking all the vertices of some maximal matching, then in fact any two of them have non-empty intersection. This is implied by the following lemma.

Lemma 4.3.4. *Let $Hyp(P)$ be a non-empty hypergraph. Let M_1 and M_2 be maximal matchings in $Hyp(P)$. Then the vertex sets of M_1 and M_2 must intersect.*

To see why this is useful in the decoding procedure, consider the following toy problem:
Graph Decoding: Carol has a graph G on k vertices. She relabels the vertices $\sigma(1), \ldots, \sigma(k)$ for some permutation $\sigma \in \mathbb{S}_k$ and gives the (relabeled) graph $\sigma(G)$ to Alice. She relabels vertices according to $\pi \in \mathbb{S}_k$ and gives $\pi(G)$ to Bob. Alice and Bob need to produce vertices i and j so that $\sigma^{-1}(i) = \pi^{-1}(j)$. They do not know σ and π, and they are not allowed to communicate.

While in general, it is hard for Alice and Bob to succeed, suppose they are promised that the maximum matching in the graph G is at most C for $C << k$. Then Alice and Bob can each pick a maximal matching A and B respectively in their graphs and output a random vertex from the vertex set. It is easy to see from Lemma 4.3.4 that the strategy succeeds with probability at least $\frac{1}{4C^2}$.

4.3.2 Consistency Testing via Folding

We now describe the technique of folding polynomials over affine subspaces, which we use to enforce the Label-Cover constraints.

Definition 4.3.1. *$P(X_1, \ldots, X_k)$ is 0-folded over $\mathbf{h} \in \mathbb{F}[2]^k$ if for all $\mathbf{x} \in \mathbb{F}[2]^n$, $P(\mathbf{x}+\mathbf{h}) = P(\mathbf{x})$.*

Every polynomial is 0-folded over $\mathbf{0}$. It is clear that the set of all such vectors $\mathbf{h}$ forms a subspace of $\mathbb{F}[2]^k$ which we denote by H. We say that $P(X_1, \ldots, X_k)$ is 0-folded over H.

Lemma 4.3.5. *Let* $\dim(H) = t$. *A polynomial* $P(X_1, \ldots, X_k)$ *is* 0*-folded over* H *iff it can be written as* $P(\lambda_1, \ldots, \lambda_{k-t})$ *where* $\lambda_i = \lambda_i(X_1, \ldots, X_k)$ *is a linear polynomial and*

$$H = \{\mathbf{x} \in \mathbb{F}[2]^k | \lambda_i(\mathbf{x}) = 0 \text{ for } 1 \leq i \leq k-t\}.$$

Proof: Firstly, consider a polynomial of the above form. Note that $\lambda_i(\mathbf{h}) = 0$, so by linearity $\lambda_i(\mathbf{x}+\mathbf{h}) = \lambda_i(\mathbf{x})$ for all $\mathbf{h} \in H$. Hence

$$P(\mathbf{x}+\mathbf{h}) = P(\lambda_1(\mathbf{x}+\mathbf{h}), \ldots, \lambda_{k-t}(\mathbf{x}+\mathbf{h})) = P(\lambda_1(\mathbf{x}), \ldots, \lambda_{k-t}(\mathbf{x})) = P(\mathbf{x}).$$

For the converse, assume P is 0-folded over H. Pick a basis $\mathbf{h}(1) \ldots, \mathbf{h}(t)$ for H. Complete this to a basis for $\mathbb{F}[2]^k$ by adding $k-t$ vectors $\mathbf{f}(1), \ldots, \mathbf{f}(k-t)$. We can write every $\mathbf{x} \in \mathbb{F}[2]^k$ as

$$\mathbf{x} = \sum_{i=1}^{k-t} \lambda_i \mathbf{f}(i) + \sum_{j=1}^{t} \mu_j \mathbf{h}(j).$$

The co-ordinates $(\lambda_1, \ldots, \lambda_{k-t})$ specify the coset of H in which $\mathbf{x}$ lies, while $\mu_1, \ldots, \mu_t$ specify its position inside the coset. We can rewrite P as a polynomial in these new variables. We claim that P now depends only on $P(\lambda_1, \ldots, \lambda_{k-t})$. Assume for contradiction that P depends on μ_1. Then we can find a point $\mathbf{x} = (\lambda_1, \ldots, \lambda_{k-t}, \mu_1, \ldots, \mu_t) \in \mathbb{F}[2]^k$ where P is sensitive to μ_1, meaning that

$$P(\lambda_1, \ldots, \lambda_{k-t}, \mu_1, \ldots, \mu_t) = 1 + P(\lambda_1, \ldots, \lambda_{k-t}, 1+\mu_1, \ldots, \mu_t)$$

In the standard basis, flipping μ_1 is equivalent to adding $\mathbf{h}(1)$. Thus we have $P(\mathbf{x}) \neq P(\mathbf{x}+\mathbf{h}(1))$ which is a contradiction. ∎

Definition 4.3.2. $P(X_1, \ldots, X_k)$ *is* 1*-folded over* $\mathbf{g} \in \mathbb{F}[2]^k$ *if for all* $\mathbf{x} \in \mathbb{F}[2]^k$, $P(\mathbf{x}+\mathbf{g}) = 1 + P(\mathbf{x})$.

It is easy to see that the set of all such $\mathbf{g}$ (if it is non-empty) is a coset of H. This is because if P is 1-folded over $\mathbf{g}$ and 0-folded over H, then it is in fact 1-folded over $\mathbf{g}+H$. Conversely, if P is folded over $\mathbf{g}$ and $\mathbf{g}'$, then it is 0-folded over $\mathbf{g}+\mathbf{g}'$ since $P(\mathbf{x}+\mathbf{g}+\mathbf{g}') = 1 + P(\mathbf{x}+\mathbf{g}) = P(\mathbf{x})$.

Henceforth, when we say that P is folded over $\mathbf{g}+H$, we mean that it is 0-folded over H and 1-folded over $\mathbf{g}+H$.

Lemma 4.3.6. *A polynomial $P(X_1, \ldots, X_k)$ is folded over $\mathbf{g} + H$ iff it can be written as $P'(\lambda_1, \ldots, \lambda_{k-t-1}) + \lambda_{k-t}$ where $\lambda_i = \lambda_i(X_1, \ldots, X_k)$ is a linear polynomial and*

$$\mathbf{g} + H = \{\mathbf{x} \in \mathbb{F}[2]^k | \lambda_i(\mathbf{x}) = 0 \text{ for } 1 \leq i \leq k-t-1 \text{ and } \lambda_{k-t}(\mathbf{x}) = 1\} \quad (34)$$

$$H = \{\mathbf{x} \in \mathbb{F}[2]^k | \lambda_i(\mathbf{x}) = 0 \text{ for } 1 \leq i \leq k-t\} \quad (35)$$

Proof: Given a polynomial of this form, it is easy to see that $P(\mathbf{x} + \mathbf{h}) = 0$ for $\mathbf{h} \in H$, whereas $P(\mathbf{x} + \mathbf{g}') = 1 + P(\mathbf{x})$ for any $\mathbf{g}'$ in $\mathbf{g} + H$.

For the converse, assume P is folded over $\mathbf{g} + H$. Pick a basis $\mathbf{h}(1) \ldots, \mathbf{h}(t)$ for H. Complete this to a basis for $\mathbb{F}[2]^k$ by adding $\mathbf{g}$ and $k-t-1$ vectors $\mathbf{f}(1), \ldots, \mathbf{f}(k-t-1)$. We can write $\mathbf{x} \in \mathbb{F}[2]^k$ as

$$\mathbf{x} = \sum_{i=1}^{k-t-1} \lambda_i \mathbf{f}(i) + \lambda_{k-t}\mathbf{g} + \sum_{j=1}^{t} \mu_j \mathbf{h}(j).$$

It is clear that in this basis, $\mathbf{g}+H$ and H are described by Equations 34 and 35 respectively. By Lemma 4.3.5, P can be written as $P(\lambda_1, \ldots, \lambda_{k-t})$. Further, the condition $P(\mathbf{x} + \mathbf{g}) = P(\mathbf{x}) + 1$ implies that

$$P(\lambda_1, \ldots, \lambda_{k-t}) = P(\lambda_1, \ldots, \lambda_{k-t-1}, 0) + \lambda_{k-t}.$$

We can check this by substituting values for λ_{k-t}. Setting $P'(\lambda_1, \ldots, \lambda_{k-t-1}) = P(\lambda_1, \ldots, \lambda_{k-t-1}, 0)$ proves the claim. ∎

4.3.2.1 Testing Equality via Folding.

Our next goal is to design a test to check if two vertices have been assigned the same labels. We will do this using folding. Given vertices u and v, each with a label in $[k]$, we wish to check if they have the same label. We assign variables $X_1, \ldots, X_k$ to vertex u, $Y_1, \ldots, Y_k$ to v. If both vertices have the label i assigned to them we expect the polynomial $X_i + Y_i$; so our test should accept all such polynomials. The decoding procedure labels u by looking at the restriction of Q to $X_1, \ldots, X_k$, and labels v by looking at the restriction to $Y_1, \ldots, Y_k$. If the test accepts some polynomial Q with non-trivial probability, we want the same label assigned to both the vertices.

Define the polynomial $D_i = X_i + Y_i$ and let $\mathcal{D}$ denote the set of all such polynomials. These polynomials are 0-folded over the subspace H of $\mathbb{F}[2]^{2k}$ which is defined by $X_i + Y_i = 0$

for all i, which consists of the vectors $(\mathbf{z}, \mathbf{z})$ for $\mathbf{z} \in \mathbb{F}[2]^k$. We want to enforce this condition on the polynomials being tested, which means they should have the form stated in Lemma 4.3.5.

This is done by a suitable projection. Pick a basis $\mathbf{h}(1), \ldots, \mathbf{h}(k)$ for H and complete it to a basis F of $\mathbb{F}[2]^{2k}$ by adding $\mathbf{f}(1), \ldots, \mathbf{f}(k)$. We can write $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}[2]^{2k}$ in this basis as $(\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$. Our test will be on polynomials $P(\lambda_1, \ldots, \lambda_k)$ of degree d. We will run the basic dictatorship test on each vertex. Our test proceeds by generating points in $\mathbb{F}[2]^{2k}$, writing them in the F-basis and projecting onto $(\lambda_1, \ldots, \lambda_k)$ and testing the polynomial P at these points in $\mathbb{F}[2]^k$.

Algorithm 4.3.3. EQUALITY TEST:

1. For vertex u, pick $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
2. Write $(\eta, \mathbf{0}^k) = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$ and test if $P(\lambda_1, \ldots, \lambda_k) = 0$.
3. For vertex v, pick $\eta' \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
4. Write $(\mathbf{0}^k, \eta') = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$ and test if $P(\lambda_1, \ldots, \lambda_k) = 0$.

In order to analyze the test, we *unfold* P and rewrite it as a polynomial in $X_1, \ldots, X_k, Y_1, \ldots, Y_k$ by substituting for each λ_i. We observe that folding enforces the following symmetry on P:

Claim 4.3.7. *The polynomial P satisfies $P(\mathbf{x}, \mathbf{y}) = P(\mathbf{y}, \mathbf{x})$ for $\mathbf{x}, \mathbf{y} \in \mathbb{F}[2]^k$.*

Proof: By Lemma 4.3.5, P is folded over H, and $(\mathbf{x}, \mathbf{y}) + (\mathbf{y}, \mathbf{x}) = (\mathbf{x} + \mathbf{y}, \mathbf{x} + \mathbf{y}) \in H$. Hence $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{y}, \mathbf{x})$ lie in the same coset of H. ∎

The corresponding decoding procedure for the Equality Test essentially runs Algorithm 4.3.2 on both the vertices u and v. We give its formal description below, followed by the analysis.

Algorithm 4.3.4. Decoding Procedure for the Equality Test:

1. Rewrite $P(\lambda_1, \ldots, \lambda_k)$ as a polynomial in $X_1, \ldots, X_k, Y_1, \ldots, Y_k$.
2. Run Algorithm 4.3.2 on $P(X_1, \ldots, X_k, \mathbf{0}^k)$ to get list $L(u)$.
3. Run Algorithm 4.3.2 on $P(\mathbf{0}^k, Y_1, \ldots, Y_k)$ to get list $L(v)$.
4. Assign $l(u) \xleftarrow{R} L(u)$ and $l(v) \xleftarrow{R} L(v)$.

In order to analyze this procedure, let us define the polynomials $U(X_1, \ldots, X_k) = P(X_1, \ldots, X_k, \mathbf{0}^k)$, and $V(Y_1, \ldots, Y_k) = P(\mathbf{0}^k, Y_1, \ldots, Y_k)$. The key observation is that P being independent of H forces the polynomials U and V to be identical.

Lemma 4.3.8. *We have $U(Z_1, \ldots, Z_k) = V(Z_1, \ldots, Z_k)$.*

Proof: The polynomials U and V each define a functions $\mathbb{F}[2]^k \to \mathbb{F}[2]$ given by

$$U(\mathbf{z}) = P(\mathbf{z}, \mathbf{0}^k), \qquad V(\mathbf{z}) = P(\mathbf{0}^k, \mathbf{z}).$$

By Claim 4.3.7, $P(\mathbf{z}, \mathbf{0}^k) = P(\mathbf{0}^k, \mathbf{z})$, hence $U = V$ as functions, and hence also as polynomials. ∎

Theorem 4.3.9. *Let $P(\lambda_1, \ldots, \lambda_k)$ be a degree d polynomial that passes the Folded Tests for both u and v with probability at least $1 - 2^{-d} + \delta$. Then $l(u) = l(v)$ with constant probability (depending on d, δ).*

Proof: Recall that for $Q(X_1, \ldots, X_k)$, $Hyp(Q)$ denotes the hypergraph on $[k]$ corresponding to the monomials in Q. By Lemma 4.3.8, $Hyp(U) = Hyp(V)$. Performing the basic dictatorship test on $U(X_1, \ldots, X_k)$ is equivalent to testing if $P(\eta, \mathbf{0}^k) = 0$, which is the same as testing that $P(\lambda_1, \ldots, \lambda_k) = 0$ for $(\eta, \mathbf{0}^k) = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$. Similarly, the basic dictatorship test on $V(Y_1, \ldots, Y_k)$ is the same as testing whether $P(\mathbf{0}^k, \eta') = 0$. Since both these tests succeed with probability $1 - 2^{-d} + \delta$, each of $L(U)$ and $L(V)$ is a maximal matching in $Hyp(U) = Hyp(V)$ of constant size. Thus by Lemma 4.3.4 choosing a random label from each results in a common label with constant probability. ∎

*4.3.2.2 **Enforcing non-Emptiness.*** We show how one can use folding to ensure that the polynomials that pass the dictatorship test and the equality test are non-zero.

For the dictatorship test, observe that the polynomials X_i are 1-folded over $\mathbf{g} = \mathbf{1}^k$. To enforce this condition on every polynomial, choose a basis $\{\mathbf{f}(1), \ldots, \mathbf{f}(k-1), \mathbf{g}\}$ for $\mathbb{F}[2]^k$. We write vectors in this basis as

$$\mathbf{x} = \sum_{i=1}^{k-1} \lambda_i \mathbf{f}(i) + \lambda_k \mathbf{g}.$$

Polynomials which are folded over $\mathbf{g}$ can be written as $P(\lambda_1, \ldots, \lambda_{k-1}) + \lambda_k$. This suggests the following test:

Algorithm 4.3.5. FOLDED DICTATORSHIP TEST:

1. Sample $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$, and write it as $\eta = (\lambda_1, \ldots, \lambda_k)$.
2. Test if $P(\lambda_1, \ldots, \lambda_{k-1}) = \lambda_k$.

To analyze this test, we define the unfolded polynomial $Q(X_1, \ldots, X_k) = P(\lambda_1, \ldots, \lambda_{k-1}) + \lambda_k$.

Theorem 4.3.10. *The polynomial $Q(X_1, \ldots, X_k)$ is folded. The probability that $P(\lambda_1, \ldots, \lambda_{k-1})$ passes the Folded Dictatorship Test equals the probability that $Q(X_1, \ldots, X_k)$ passes the Basic Dictatorship Test.*

Proof: If $\mathbf{x} = (\lambda_1, \ldots, \lambda_k)$, then $\mathbf{x} + \mathbf{1}^k = (\lambda_1, \ldots, \lambda_{k-1}, 1 + \lambda_k)$. Hence

$$Q(\mathbf{x} + \mathbf{1}^k) = (1 + \lambda_k) + P(\lambda_1, \ldots, \lambda_{k-1}) = 1 + Q(\mathbf{x})$$

so Q is folded over $\mathbf{1}^k$. In fact by Theorem 4.3.2, Q has no constant term, so $Q(\mathbf{0}^k) = 0, Q(\mathbf{1}^k) = 1$.

To show that Q passes the Basic Dictatorship Test, note that

$$P(\lambda_1, \ldots, \lambda_{k-1}) = \lambda_k \iff P(\lambda_1, \ldots, \lambda_{k-1}) + \lambda_k = 0 \iff Q(\eta) = 0.$$

■

In the Equality test, we want to ensure that the polynomials $U(X_1, \ldots, X_k)$ and $V(Y_1, \ldots, Y_k)$ are both non-zero. Define $H = (\mathbf{z}, \mathbf{z})$ as before and let $\mathbf{g} = (\mathbf{1}^k, \mathbf{0}^k)$. The polynomials $X_i + Y_i$ are folded over the coset $\mathbf{g} + H$. We wish to enforce this condition on the polynomials that are being tested, which means they should have the form stated in Lemma 4.3.6. Pick a basis $F = \{\mathbf{f}(1), \ldots, \mathbf{f}(k-t-1), \mathbf{g}, \mathbf{h}(1), \ldots, \mathbf{h}(t)\}$ for $\mathbb{F}[2]^{2k}$ and let

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k-t-1} \lambda_i \mathbf{f}_i + \lambda_{k-t}\mathbf{g} + \sum_{j=1}^{t} \mu_j \mathbf{h}(j).$$

Given a point $(\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$ in this basis, we test if $P(\lambda_1, \ldots, \lambda_{k-1}) = \lambda_k$. Thus the test is on polynomials in $k-1$ variables.

Algorithm 4.3.6. FOLDED EQUALITY TEST:

1. For vertex u, pick $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
2. Write $(\eta, \mathbf{0}^k) = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$ and test if $P(\lambda_1, \ldots, \lambda_{k-1}) = \lambda_k$.
3. For vertex v, pick $\eta' \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
4. Write $(\mathbf{0}^k, \eta') = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k)$ and test if $P(\lambda_1, \ldots, \lambda_{k-1}) = \lambda_k$.

Define the unfolded polynomial $Q(X_1, \ldots, X_k, Y_1, \ldots, Y_k) = P(\lambda_1, \ldots, \lambda_{k-1}) + \lambda_k$. We denote the restriction of Q to $X_1, \ldots, X_k$ by U and $Y_1, \ldots, Y_k$ by V.

Theorem 4.3.11. *The polynomials $U(X_1, \ldots, X_k)$ and $V(Y_1, \ldots, Y_k)$ are both folded over $\mathbf{1}^k$. If $P(\lambda_1, \ldots, \lambda_{k-1})$ passes the Locally Folded Equality Test for both u and v with probability $1 - 2^{-d} + \delta$, then both $U(X_1, \ldots, X_k)$ and $V(Y_1, \ldots, Y_k)$ pass the Basic Dictatorship Test with probability $1 - 2^{-d} + \delta$.*

Proof: The proof that U and V pass the Dictatorship test follows that of Theorem 4.3.10.

Observe that the polynomial Q is folded over $\mathbf{g} + H$, which contains the points $\mathbf{g} = (\mathbf{1}^k, \mathbf{0}^k)$ and $\mathbf{g}' = (\mathbf{0}^k, \mathbf{1}^k)$. Thus

$$U(\mathbf{x} + \mathbf{1}^k) = Q((\mathbf{x}, \mathbf{0}^k) + \mathbf{g}) = 1 + Q((\mathbf{x}, \mathbf{0}^k)) = 1 + U(\mathbf{x}).$$

Similarly one can use $\mathbf{g}'$ to show that V is folded over $\mathbf{1}^k$. ∎

4.3.3 Consistency Testing for Projections

We will consider the following consistency problem: there are two vertices u and v, each of them is assigned a label $l(u), l(v) \in [k]$ respectively. The vertex u is assigned a projection function $\pi : [k] \to [t]$, while the vertex v is assigned a projection function $\sigma : [k] \to [t]$. The goal is to check whether the labels $l(u)$ and $l(v)$ satisfy $\pi(l(u)) = \sigma(l(v))$. We want a test that accepts all polynomials of the form $X_i + Y_j$ where $\pi(i) = \sigma(j)$. Let us denote the set of all such polynomials by $\mathcal{D}$. The test will specify target values for points of the form $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}[2]^{2k}$ projected onto a certain lower dimensional subspace.

We start by constructing a subspace H on which every polynomial in $\mathcal{D}$ vanishes. Consider the subspace H defined by the equations

$$X_i + Y_j = 0, \qquad \pi(i) = \sigma(j) \tag{36}$$

We would like a parametric description of this subspace, for which we need the following definition [38].

Definition 4.3.3. *Given a projection function $\pi : [k] \to [t]$, for $\mathbf{z} \in \mathbb{F}[2]^t$, we define the vector $\mathbf{z} \circ \pi \in \mathbb{F}[2]^k$ by $(\mathbf{z} \circ \pi)_i = \mathbf{z}_{\pi(i)}$.*

This gives a linear map from $\mathbb{F}[2]^t \to \mathbb{F}[2]^k$ since

$$(\mathbf{z_1} + \mathbf{z_2}) \circ \pi = \mathbf{z_1} \circ \pi + \mathbf{z_2} \circ \pi.$$

Lemma 4.3.12. *The subspace H contains the vectors $(\mathbf{z} \circ \pi, \mathbf{z} \circ \sigma)$ for $\mathbf{z} \in \mathbb{F}[2]^t$.*

Proof: We need to check that $(\mathbf{x}, \mathbf{y}) = (\mathbf{z} \circ \pi, \mathbf{z} \circ \sigma)$ satisfies $x_i + y_j = 0$ for all $\pi(i) = \sigma(j)$. But

$$x_i = (\mathbf{z} \circ \pi)_i = z_{\pi(i)}, \quad y_j = (\mathbf{z} \circ \sigma)_j = z_{\sigma(j)} \quad \text{hence} \quad x_i = y_j.$$

In fact a simple dimension argument shows that $H = \{(\mathbf{z} \circ \pi, \mathbf{z} \circ \sigma) \mid \mathbf{z} \in \mathbb{F}[2]^t\}$ but we will not need this fact. ∎

Let $\mathbf{g} = (\mathbf{1}^k, \mathbf{0}^k)$. Every polynomial in $\mathcal{D}$ is folded over $\mathbf{g} + H$. We pick a basis $\mathbf{h}(1), \ldots, \mathbf{h}(t)$ for H and complete this to a basis F of $\mathbb{F}[2]^{2k}$ given by $F = \{\mathbf{f}(1), \ldots, \mathbf{f}(2k -$

$t-1), \mathbf{g}, \mathbf{h}(1), \ldots, \mathbf{h}(t)\}$ for some suitable choice of $\mathbf{f}(i)$s. Set $g = 2k - t$. Then

$$(\mathbf{x}, \mathbf{y}) \ = \sum_{i=1}^{g-1} \lambda_i \mathbf{f}(i) + \lambda_g \mathbf{g} + \sum_{j=1}^{t} \mu_j \mathbf{h}(j)$$

Algorithm 4.3.7. Folded Consistency Test:

1. For vertex u, pick $\eta \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
2. Write $(\eta, \mathbf{0}^k) = (\lambda_1, \ldots, \lambda_g, \mu_1, \ldots, \mu_t)$ and test if $P(\lambda_1, \ldots, \lambda_{g-1}) = \lambda_g$.
3. For vertex v, pick $\eta' \xleftarrow{\varepsilon} \mathbb{F}[2]^k$.
4. Write $(\mathbf{0}^k, \eta') = (\lambda_1, \ldots, \lambda_g, \mu_1, \ldots, \mu_t)$ and test if $P(\lambda_1, \ldots, \lambda_{g-1}) = \lambda_g$.

Algorithm 4.3.8. Decoding Procedure for the Folded Consistency Test:

1. Let $Q(X_1, \ldots, X_k, Y_1, \ldots, Y_k) = P(\lambda_1, \ldots, \lambda_{g-1}) + \lambda_g$.
2. Run Algorithm 4.3.2 on $Q(X_1, \ldots, X_k, \mathbf{0}^k)$ to get list $L(u)$.
3. Run Algorithm 4.3.2 on $Q(\mathbf{0}^k, Y_1, \ldots, Y_k)$ to get list $L(v)$.
4. Assign $l(u) \xleftarrow{R} L(u)$ and $l(v) \xleftarrow{R} L(v)$.

As before we define the polynomials $U(X_1, \ldots, X_k) = Q(X_1, \ldots, X_k, \mathbf{0}^k)$ and $V(Y_1, \ldots, Y_k) = Q(\mathbf{0}^k, Y_1, \ldots, Y_k)$. The relation between the two polynomials enforced by folding is a bit more intricate. The key observation is that their *projections* in $Z_1, \ldots, Z_t$ obtained by replacing X_i by $Z_{\pi(i)}$ in U and Y_j by $Z_{\sigma(j)}$ in V are the same.

Lemma 4.3.13. *Define the projected polynomials*

$$U_\pi(Z_1, \ldots, Z_t) = U(Z_{\pi(1)}, \ldots, Z_{\pi(k)}), \quad V_\sigma(Z_1, \ldots, Z_t) = U(Z_{\sigma(1)}, \ldots, Z_{\sigma(k)}).$$

Then $U_\pi(Z_1, \ldots, Z_t) = V_\sigma(Z_1, \ldots, Z_t)$.

Proof: We can view U_π and V_σ as functions $\mathbb{F}[2]^t \to \mathbb{F}[2]$ given by

$$U_\pi(\mathbf{z}) = Q(\mathbf{z} \circ \pi, \mathbf{0}^k), \qquad V_\sigma(\mathbf{z}) = Q(\mathbf{0}^k, \mathbf{z} \circ \sigma).$$

Since the polynomial Q is folded over H, it satisfies $Q(\mathbf{z} \circ \pi, \mathbf{0}^k) = Q(\mathbf{0}^k, \mathbf{z} \circ \sigma)$ since

$$(\mathbf{z} \circ \pi, \mathbf{0}^k) + (\mathbf{0}^k, \mathbf{z} \circ \sigma) = (\mathbf{z} \circ \pi, \mathbf{z} \circ \sigma) \in H.$$

Hence $U_\pi(\mathbf{z}) = V_\sigma(\mathbf{z})$ as functions, hence $U_\pi(Z_1, \ldots, Z_t) = V_\sigma(Z_1, \ldots, Z_t)$ as polynomials. ∎

We can now analyze Algorithm 4.3.8.

Theorem 4.3.14. *Define the projections of the lists $L(u)$ and $L(v)$ as $L_\pi(u) = \{\pi(i) \mid i \in L(u)\}$ and $L_\sigma(v) = \{\sigma(j) \mid j \in L(v)\}$.*

1. *Both $L_\pi(u))$ and $L_\sigma(v)$ are vertex covers for the hypergraph $Hyp(U_\pi) = Hyp(V_\sigma)$.*
2. *The polynomials U_π and V_σ are each folded over $\mathbf{1}^t$.*
3. *The probability that $P(\lambda_1, \ldots, \lambda_{g-1})$ passes the folded consistency test for vertex u equals the probability that $U(X_1, \ldots, X_k)$ passes the Basic Dictatorship Test.*

Proof: The hypergraph $Hyp(U_\pi)$ is obtained from $Hyp(U)$ by identifying the vertices in $\pi^{-1}(\ell)$ for each $\ell \in [t]$. The edges in this hypergraph are those which have an odd number of pre-images in $Hyp(U)$. Thus the projection of any vertex cover for $Hyp(U)$ is also a vertex cover for $Hyp(U_\pi)$. From Algorithm 4.3.2, $L(u)$ is a vertex cover for $Hyp(U)$, so $L_\pi(u)$ is a vertex cover for $Hyp(U_\pi)$. Similarly $L_\sigma(v)$ is a vertex cover for $Hyp(V_\sigma)$. By Lemma 4.3.13, since $U_\pi = V_\sigma$, both polynomials define the same hypergraph.

By the same argument used for Theorem 4.3.11, we can show that U and V are folded over $\mathbf{1}^k$. But

$$U_\pi(\mathbf{z} + \mathbf{1}^t) = U((\mathbf{z} + \mathbf{1}^t) \circ \pi) = U(\mathbf{z} \circ \pi + \mathbf{1}^k) = 1 + U(\mathbf{z} \circ \pi) = 1 + U_\pi(\mathbf{z}).$$

So U_π is folded over $\mathbf{1}^t$ and similarly for V_σ. This shows that the hypergraph $Hyp(U_\pi) = Hyp(V_\sigma)$ is non-empty.

The proof of Part 3 follows that of Theorem 4.3.10. ∎

Thus, if P passes the test then $L(u)$ and $L(v)$ are small in size, their projections are vertex-covers for the same (non-empty) hypergraph. It is natural to ask if choosing $l(u) \xleftarrow{R} L(u)$ and $l(v) \xleftarrow{R} L(v)$ gives $\pi(l(u)) = \sigma(l(v))$ with some probability. This might not be the case. The reason is that while the vertex cover $L(u)$ obtained by taking all the vertices of a maximal matching, the projection $L_\pi(u)$ need not have this structure. Thus $L_\pi(u)$ and $L_\sigma(v)$ might be disjoint vertex covers of the same hypergraph. However, the fact that they are both vertex covers together with Lemma 4.3.3 will suffice for our analysis. We note however that if $d = 1$, then the vertex covers will intersect, so the random decoding succeeds. This, combined with an appropriate reduction from Label-Cover, would give an alternative proof of Håstad's result for $d = 1$.

4.4 The Reduction from Label-Cover$[d+1]$

We need some notation in order to describe the reduction to PolyRec(d). Let $\mathcal{L}$ be an instance of Label-Cover$[d+1](t, k)$ consisting of a hypergraph $G(V, E)$ on n vertices and m hyperedges, with each hyperedge associated with a $d+1$-tuple of projection functions. To each vertex $v \in V$, we assign k variables $X_1^v, \ldots, X_k^v$. Since there are a total of nk variables $X_1^{v_1}, \ldots, X_k^{v_n}$, our points will be in nk dimensions, partitioned into n groups, one for each vertex, and each group having k dimensions, one for each possible vertex label. Given $\mathbf{x} \in \mathbb{F}[2]^{nk}$, we use $\mathbf{x}^v$ to denote the vector in $\mathbb{F}[2]^k$ obtained by projecting onto the co-ordinates assigned to vertex v. To a labeling l of vertices in V, we associate the polynomial $Q_l(X_1^{v_1}, \ldots, X_k^{v_n}) = \sum_v X^v_{l(v)}$.

Our first goal is to identify a subspace H such that if l satisfies all the constraints of $\mathcal{L}$, then Q_l is 0-folded over H. Unlike for the simple tests considered so far, we do not know what the set of polynomials Q_l is, or whether it is non-empty. However, one can identify vectors that must lie in H from the constraints of $\mathcal{L}$.

Lemma 4.4.1. *Consider a pair of vertices u, w that lie in $e \in E$. Suppose the projections associated with them by e are π and σ respectively. Given $\mathbf{z} \in \mathbb{F}[2]^t$, define the vector*

$\mathbf{h} = \mathbf{h}(\mathbf{z}, e, u, w) \in \mathbb{F}[2]^{nk}$ *where*

$$\mathbf{h}^v = \begin{cases} \mathbf{z} \circ \pi \ \textit{if } v = u \\ \mathbf{z} \circ \sigma \ \textit{if } v = w \\ \mathbf{0}^k \ \textit{otherwise.} \end{cases} \tag{37}$$

If l satisfies $\pi(l(u)) = \sigma(l(v))$ then $Q_l(\mathbf{h}) = 0$.

Proof: Note that

$$Q_l(\mathbf{h}) = \sum_{v \in V} h^v_{l(v)} = h^u_{l(u)} + h^w_{l(w)}.$$

Also

$$h^u_{l(u)} = (\mathbf{z} \circ \pi)_{l(u)} = z_{\pi(l(u))}, \quad h^w_{l(w)} = (\mathbf{z} \circ \sigma)_{l(w)} = z_{\sigma(l(w))}.$$

But $\pi(l(u)) = \sigma(l(w))$, hence $h^u_{l(u)} + h^w_{l(w)} = 0$. ∎

We take H to be the span of all the vectors $\mathbf{h}$ above, over all choices of $e \in E$, $u, w \in e$ and $\mathbf{z} \in \mathbb{F}[2]^t$.

Let $\mathbf{g}(v) = \sum_{i=1}^{k} \mathbf{e}^v_i$ be the indicator for the co-ordinates of the vertex v. Let $\mathbf{g} = \mathbf{g}(v_1)$. Observe that every polynomial associated to a labeling satisfies $Q_l(\mathbf{g}(v)) = 1$.

Lemma 4.4.2. *The affine subspace $\mathbf{g} + H$ contains the vectors $\mathbf{g}(v)$ for all $v \in V$.*

Proof: Assume that $u, w \in e$ for some $e \in E$. Let π and σ denote the associated projections. Then $\mathbf{g}(u) + \mathbf{g}(w) \in H$, since this vector is obtained by setting $\mathbf{z} = \mathbf{1}^t$ in Equation 37. Since the hypergraph is connected, it follows that all the vectors $\mathbf{g}(v)$ lie in the same coset of H. ∎

We will ensure that the polynomials we test are folded over $\mathbf{g} + H$. Let the dimension of the space H be h, and select a basis $\{\mathbf{h}(j)\}_{j=1}^{h}$ for it. Complete this to a basis F of $\mathbb{F}[2]^{nk}$ by adding $\mathbf{g}$ and some other vectors $\mathbf{f}(1), \ldots, \mathbf{f}(nk - h - 1)$. Let $g = nk - h$. One can write any $\mathbf{x} \in \mathbb{F}[2]^{nk}$ as

$$\mathbf{x} \ = \sum_{i=1}^{g-1} \lambda_i \mathbf{f}(i) + \lambda_g \mathbf{g} + \sum_{j=1}^{h} \mu_j \mathbf{h}(j)$$

Let $\eta(v) \in \mathbb{F}[2]^{nk}$ denote the random variable where each co-ordinate corresponding to vertex v is sampled from the ε-biased distribution and all other co-ordinates as 0. The

reduction shall project $\eta(v)$ onto the subspace generated by basis vectors $\mathbf{f}(1), \ldots, \mathbf{f}(g-1)$, which can be represented in terms of the variables $\lambda_1, \ldots, \lambda_{g-1}$. We now state the following algorithm which gives the reduction from $\mathcal{L}$ to an instance $\mathcal{I}$ of PolyRec(d). Note that the instance $\mathcal{I}$ is over $g-1$ variables $\lambda_1, \ldots, \lambda_{g-1}$.

Algorithm 4.4.1. Reduction

1. Compute the basis F described above.
2. Pick a vertex $v \xleftarrow{R} V$ and sample the vector $\eta(v)$.
3. Write $\eta(v) = (\lambda_1, \ldots, \lambda_g, \mu_1, \ldots, \mu_h)$.
4. Output the point-value pair $\langle(\lambda_1, \ldots, \lambda_{g-1}), \lambda_g\rangle$.

We need to massage the above reduction to produce an instance of PolyRec(d). The above algorithm produces a distribution $\mathcal{D}$ on polynomially many constraints of the form $\langle \mathbf{x}, \zeta \rangle$. We now repeat each $\mathbf{x}$ sufficiently many times to simulate the distribution, to get the instance $\mathcal{I}$ of PolyRec(d). Therefore, we only need to analyze the performance of a polynomial with respect to the distribution $\mathcal{D}$ generated by Algorithm 4.4.1. We analyze the YES and the NO cases separately as follows.

4.4.1 YES Case

In the YES case there is a labeling l to the vertices V of $\mathcal{L}$ that strongly satisfies all the hyperedges. Therefore, by the analysis earlier in this section, the associated polynomial $Q_l = Q_l(X_1^{v_1}, \ldots, X_k^{v_n}) = \sum_v X_{l(v)}^v$ is folded over $\mathbf{g} + H = \mathbf{g}(v) + H$ for all $v \in V$. By Lemma 4.3.6,

$$Q_l(X_1^{v_1}, \ldots, X_k^{v_n}) = P_l(\lambda_1, \ldots, \lambda_{g-1}) + \lambda_g, \tag{38}$$

under the appropriate basis transformation from $\mathbb{F}[2]^{nk}$ to F. Moreover, since Q_l is linear, P_l is also linear in $\lambda_1, \ldots, \lambda_{g-1}$. We choose P_l to be the linear function in the YES case.

Now, for a point $\eta(v) \in \mathbb{F}[2]^{nk}$, where $\eta(v) = (\lambda_1, \ldots, \lambda_g, \mu_1, \ldots, \mu_h)$ in the basis F, we

have from Equation (38),

$$P_l(\lambda_1, \ldots, \lambda_{g-1}) = \lambda_g \iff Q_l(\eta(v)) = 0.$$

Therefore, the probability that P_l succeeds under the distribution $\mathcal{D}$ is same as the probability that $Q_l(\eta(v)) = 0$. This is exactly $1 - \varepsilon$, from the definition of Q_l and $\eta(v)$, for any vertex $v \in V$. Therefore, there is a linear function P_l that satisfies $1 - \varepsilon$ fraction of the point value pairs of $\mathcal{I}$.

4.4.2 NO Case

We give a decoding procedure that uses the polynomial P to assign labels to every vertex. If P succeeds w.r.t $\mathcal{D}$ with good probability, then the resulting labeling is guaranteed to weakly satisfy a good fraction of constraints. This implies that if we reduce from a NO instance of Label-Cover then no polynomial succeeds with good probability. Given a polynomial $Q(X_1^{v_1}, \ldots, X_k^{v_n})$, for each vertex $v \in V$, we use $Q(X^v)$ to denote the restriction of Q to the variables $\{X_i^v\}_{i=1}^k$, obtained by setting all other variables to 0.

Algorithm 4.4.2. Decoding Procedure for the Reduction

1. Set $Q(X_1^{v_1}, \ldots, X_k^{v_n}) = P(\lambda_1, \ldots, \lambda_{g-1}) + \lambda_g$.
2. For every vertex $v \in V$,
   (a) Run Algorithm 4.3.2 on $Q(X^v)$ to get list $L(v)$.
   (b) Set $l(v) \xleftarrow{R} L(v)$.

Theorem 4.4.3. *Assume that $P(\lambda_1, \ldots, \lambda_{g-1})$ succeeds w.r.t. the distribution $\mathcal{D}$ generated by Algorithm 4.4.1 with probability $1 - 2^{-d} + 2\delta$ for $\delta > 0$. Then the labeling $l(v)$ weakly satisfies γ' fraction of the constraints in expectation for some $\gamma'(\varepsilon, \delta, d)$.*

Proof: By an averaging argument, for a δ fraction of vertices in V, the probability succeeding w.r.t $\mathcal{D}$ is at least $1 - 2^{-d} + \delta$; denote this set by S and call such vertices *good*.

The *good* set of edges $E(S)$ induced by S is at least a γ fraction of all edges for some constant $\gamma(\delta)$.

Pick an edge $e \in E(S)$, and pick any two vertices $u, w \in e$. Both these will be good vertices. Let $Q(X^u, X^w)$ denote the restriction of the polynomial $Q(X_1^{v_1}, \ldots, X_k^{v_n})$ to the variables $\{X_i^u, X_j^w\}_{i,j=1}^k$ obtained by setting the other variables to 0. This polynomial is 0-folded over the set of vectors $H' = (\mathbf{z} \circ \pi, \mathbf{z} \circ \sigma)$. It is folded over $\mathbf{g} + H'$ where $\mathbf{g} = (\mathbf{1}^k, \mathbf{0}^k)$. So we can apply Theorem 4.3.14 to conclude that the projections of the polynomials $Q(X^u)$ and $Q(X^w)$ under π and σ respectively are identical, to say $\overline{Q}$, and $\pi(L(u))$ and $\sigma(L(w))$ each give a vertex cover for the (non-empty) hypergraph $Hyp(\overline{Q})$ of this projected polynomial $\overline{Q}$. Further, since u and w are good vertices, by Theorem 4.3.2 both $L(u)$ and $L(w)$ are small. Hence their projections $L_\pi(u)$ and $L_\sigma(w)$ are also small.

Since this is true for any pair of vertices in e, we have $d+1$ vertex covers of $Hyp(\overline{Q})$. But each edge of $Hyp(\overline{Q})$ has size at most d, so by Lemma 4.3.3 some two of them intersect, assume that these are $\pi(L(u))$ and $\sigma(L(v))$. In other words, there are labels $\ell_1 \in L(u)$ and $\ell_2 \in L(v)$ so that $\pi(\ell_1) = \sigma(\ell_2)$. Since each of these lists is of constant size (depending only on ε, δ, d), there is a constant probability $p = p(\varepsilon, \delta, d)$ that these are the labels chosen for u and v respectively by the random decoding in Step 2b. In this case, the constraint is weakly satisfied. Thus the expected number of satisfied constraints is $\gamma'(\varepsilon, \delta, d) = p \cdot \gamma$. ∎

By taking the soundness β of the instance $\mathcal{L}$ to be a sufficiently small constant, we conclude that in the No case, there is no polynomial of degree at most d that succeeds with respect to $\mathcal{D}$ with probability $1 - 2^{-d} + \delta$, else we would reach a contradiction. Therefore, no polynomial of degree at most d satisfies $1 - 2^{-d} + \delta$ fraction of the point-value pairs of $\mathcal{I}$. This proves Theorem 4.2.2 and therefore Theorem 1.6.5.

4.5 Inapproximability of Label-Cover$[d+1]$

In this section we give a reduction from a Label-Cover instance to an instance of Label-Cover$[d+1]$ (refer to Defn 4.2.2) thereby proving Theorem 4.2.1. The details of the reduction are as follows.

We start with an instance $\mathcal{L}$ of Label-Cover(t, k), consisting of a bipartite graph

$G(U, V, E)$ and projections $\{\pi_{vu}\}_{(u,v)} \in E)$ where $\pi_{uv} : [k] \mapsto [t]$ for every $(u, v) \in E$ where $u \in U$ and $v \in V$. We construct an instance $\mathcal{L}'$ of Label-Cover$[d + 1]$ in the following manner:

1. The vertex set of $\mathcal{L}'$ is $V' = V$.

2. A hyperedge e' is added in the following manner. Pick a random $u \in U$ and pick vertices $v_1, v_2, \ldots, v_{d+1}$, uniformly at random from the neighbors of u in G. Set $e' = \{v_i\}_{i=1}^{d+1}$, and the associated $d + 1$-tuple of projections to be $\{\pi_i\}_{i=1}^{d+1}$, where $\pi_i = \pi u v_i$ for all $1 \leq i \leq d + 1$.

3. Add all such hyperedges possible to the edge set E'.

Consider a subset $S \subseteq V' = V$ of size $\delta|V'|$. Let u be any vertex in U of the instance $\mathcal{L}$. Let p_u be the fraction of neighbors of u in S. Since, every vertex of U has the same degree and every vertex of V has the same degree, $\mathrm{E}_{u \in_R U}[p_u] = \delta$. The way edge set E' of $\mathcal{L}'$ is constructed implies that the fraction of hyperedges in E' induced by S is the probability that all $d + 1$ vertices uniformly chosen at random from neighbors of a vertex u (which is chosen uniformly at random from U), lie in S. For a given $u \in U$, the probability that $d + 1$ vertices chosen uniformly at random from its neighbors lie in S is p_u^{d+1}. Therefore the fraction of edges of E' induced by S is $\mathrm{E}_{u \in_R U}[p_u^{d+1}] \geq (\mathrm{E}_{u \in_R U}[p_u])^{d+1} = \delta^{d+1}$. Hence, a constant fraction of hyperedges in E' are induced by a subset S of constant fraction of vertices in V'.

Note that by applying Parallel Repetition on Label-Cover we can increase the degrees of vertices in U arbitrarily while reducing the soundness. Since $d + 1$ is a fixed constant, we can arbitrarily reduce the fraction of hyperedges of Label-Cover$[d + 1]$ which have repeated vertices and hence remove these hyperedges from the instance.

Completeness. If $\mathcal{L}$ is a YES instance, then there is a labeling l that satisfies all the edges of $\mathcal{L}$. Clearly, the labeling l restricted to V will strongly satisfy all the hyperedges of $\mathcal{L}'$.

Soundness. If $\mathcal{L}$ is a No instance, then there is no labeling that satisfies β fraction of the edges of $\mathcal{L}$. Now, suppose that there is a labeling l that weakly satisfies α fraction of

hyperedges of $\mathcal{L}'$. For every vertex $u \in U$, define q_u to be the probability that two (distinct) random neighbors of u are labeled consistently by l. Since every vertex in U has equal degree and every vertex of V has equal degree, and by union bound, we obtain, $\mathrm{E}_u[q_u] \geq \alpha/\binom{d+1}{2}$. Let $2\alpha' = \alpha/\binom{d+1}{2}$. Call a vertex $u \in U$ 'good' if $q_u \geq \alpha'$. By averaging, at least α' fraction of vertices U are good. Let $u \in U$ be a 'good' vertex, i.e. l labels at least α' fraction of pairs $\{v_i, v_j\}$ consistently where v_i and v_j are neighbors of u. Again, by averaging, there must be a neighbor v' of u which is consistently labeled with at least $\alpha'/2$ fraction of neighbors of u. Now, extending the labeling l to u, by setting $l(u) = \pi_{uv'}(l(v'))$ will satisfy at least $\alpha'/2$ fraction of edges incident on u in $\mathcal{L}$. By labeling every 'good' vertex in a similar manner, we obtain a labeling l that satisfies at least $\alpha'^2/2$ fraction of edges of $\mathcal{L}$. Since $d+1$ is a fixed constant, for any $\alpha > 0$, choosing β to be small enough, we get a contradiction. So, there is no labeling of $\mathcal{L}'$ that weakly satisfies α fraction of the hyperedges.

4.6 Conclusion

For any constants $\delta, \varepsilon > 0$ and positive integer d, we show a $1 - 2^{-d} + \delta$ factor hardness of approximation for the polynomial reconstruction problem over $\mathbb{F}[2]$: given a set of point value pairs in $\mathbb{F}[2]^n$ with the guarantee that there exists a linear function that satisfies $1-\varepsilon$ fraction of the point-value pairs, to find a polynomial of degree at most d that satisfies as many pairs as possible.

Our result, however, is not tight and an optimal hardness factor of $\frac{1}{2} + \delta$ seems possible using the recent result on pseudo-random generators due to Viola [82]. As a hardness of learning result, even this conceivable construction would suffer from the limitation (as discussed in Section 2.8) that the error δ would be far from an inverse polynomial in the dimension of the instance. Thus, obtaining a polynomially small error δ in the soundness $\frac{1}{2} + \delta$ is an important open question.

CHAPTER V

SDP INTEGRALITY GAPS WITH LOCAL ℓ_1 EMBEDDABILITY

In this chapter we prove Theorem 1.6.6 regarding the integrality gap of the MAXIMUM CUT SDP relaxation augmented with the Sherali-Adams LP constraints. The proof Theorem 1.6.7, which is an analogous result for the SPARSEST CUT problem is very similar and is sketched in Section 5.7 of this chapter. We start by first giving a description of the SDP relaxation augmented with the Sherali-Adams LP.

SDP Augmented with Sherali-Adams LP

For a cut-problem such as MAXIMUM CUT, t-rounds of the Sherali-Adams LP hierarchy [75] (or $O(t)$-rounds if a somewhat different formulation is used, see [24]) amount to the following: on a graph $G(V, E)$, for every subset S of up to t vertices, there is a distribution $D(S)$ on $\{-1, 1\}^S$, thought of as a distribution over cuts on S. The distributions $\{D(S)\}_{S \subseteq V, |S| \leq t}$ are mutually consistent in the sense that if $T \subseteq S \subseteq V,\ |S| \leq t$, then $D(S)|_T = D(T)$, i.e. the marginal of $D(S)$ on the subset T is exactly equal to $D(T)$. The value of such a solution is average over all edges $(u, v) \in E$, of the probability $p_{u,v}$ that u and v are separated by a random cut on the set $S = \{u, v\}$ sampled according to the distribution $D(S)$. On the other hand, a basic SDP relaxation (one used by Goemans and Williamson [33]) amounts to assigning a unit vector $\mathbf{w}_u$ for every vertex $u \in V$ and the value of the solution is average over edges $(u, v) \in E$, of the quantity $\frac{1-\langle \mathbf{w}_u, \mathbf{w}_v \rangle}{2}$. We say that the SDP solution is consistent with the Sherali-Adams solution if $\forall u, v \in V,\ p_{u,v} = \frac{1-\langle \mathbf{w}_u, \mathbf{w}_v \rangle}{2}$. Finally, a (c, s)-integrality gap (or c/s-gap if concerned only with the ratio) for a LP/SDP relaxation is a graph along with a LP/SDP solution such that the relaxation has value at least c whereas the true (integral) optimum, i.e. the relative size of the maximum cut, is at most s. As is standard, existence of an integrality gap instance is taken as evidence that an algorithm based on such relaxation cannot yield an approximation guarantee better than c/s.

In the next section we give an overview of the main techniques involved in proving the results in this chapter.

5.1 Overview

Let us first restate, from Section 1.6.4, the results of this chapter. We have the following two theorems for the MAXIMUM CUT and SPARSEST CUT problems respectively.

Theorem. (1.6.6 restated) *Let $\varepsilon > 0$ be an arbitrarily small constant. For the* MAXIMUM CUT *problem on a graph of n vertices, the SDP relaxation augmented with $O((\log\log\log n)^{\frac{1}{6}})$ rounds of Sherali-Adams LP hierarchy has an integrality gap at least $\alpha_{GW}^{-1} - \varepsilon$.*

Theorem. (1.6.7 restated) *For the* SPARSEST CUT *problem on a graph of n vertices, the SDP relaxation augmented with $O((\log\log\log n)^{\frac{1}{6}})$ rounds of Sherali-Adams has an integrality gap at least $\Omega((\log\log\log n)^{\frac{1}{13}})$. Also, there is an n-point negative type metric such that every sub-metric on $O((\log\log\log n)^{\frac{1}{6}})$ points is isometrically ℓ_1-embeddable, but embedding the whole metric into ℓ_1 incurs distortion $\Omega((\log\log\log n)^{\frac{1}{13}})$.*

The constructions for the MAXIMUM CUT and the SPARSEST CUT integrality gaps are very similar (one only needs to change a certain *perturbation parameter*) and therefore, for the sake of exposition we shall focus only the MAXIMUM CUT integrality gap.

High level strategy

Our construction relies in large part on the work of Khot and Vishnoi [54] who gave SDP integrality gap examples for UNIQUE GAMES and cut-problems including MAXIMUM CUT. Their overall approach was to follow the reduction from UNIQUE GAMES to the target problem (say) MAXIMUM CUT. They first construct an integrality gap example for the UNIQUE GAMES SDP, i.e. an instance with low optimum (i.e. no good labeling) and a vector solution with high objective value. Using the reduction from [48], they convert the instance of UNIQUE GAMES with low optimum to an instance of MAXIMUM CUT, also with low optimum. The same reduction also transforms the vector solution for the UNIQUE GAMES SDP into a vector solution for the MAXIMUM CUT SDP. The transformation ensures that the

Maximum Cut SDP solution has a high objective value, thereby providing an integrality gap. In this work, we observe that there is also a natural way to construct a *good* solution to the Sherali-Adams LP relaxation for the Unique Games instance constructed in [54]. This solution can then be transformed into one for the Sherali-Adams LP relaxation for the Maximum Cut instance, via the same reduction as before. Again, the transformation ensures that the objective value of the Sherali-Adams solution remains high. Moreover, for any set of two vertices, the Sherali-Adams solution is *almost* consistent with the SDP vector solution. We then *massage* these solutions so that they are exactly consistent, yielding the integrality gap for Maximum Cut SDP augmented with super-constant rounds of Sherali-Adams LP. The next few paragraphs give an informal description of the construction.

Sherali-Adams solution (labeling) to Unique Games instance

We start with the Unique Games instance $\mathcal{U}$ constructed by Khot and Vishnoi [54]. Let $G(V, E)$ be its constraint graph and $[N]$ be the label set. The first step is to construct Sherali-Adams solution for $\mathcal{U}$. Specifically, we construct for every set $U \subseteq V$, $|U| \leq t$, a distribution $D(U)$ over labelings $\sigma : U \mapsto [N]$ such that:

- The distributions are mutually consistent, i.e. for any $W \subseteq U \subseteq V$, $|U| \leq t$, $D(U)|_W = D(W)$.
- The objective value of the solution is high, i.e. if $(u, v) \in E$ is a Unique Games constraint, then a random labeling $\sigma : \{u, v\} \mapsto [N]$ from $D(\{u, v\})$ satisfies the constraint with probability close to 1.

Towards this end, we look at the [54] example closely, and observe that one can define a metric $\rho(\cdot, \cdot)$ on the vertex set V such that any two vertices with an edge/constraint between them are very close w.r.t. ρ. Moreover for any set $U \subseteq V$, $|U| \leq t$ that has low diameter w.r.t. ρ, it is possible to assign a randomized labeling $\sigma : U \mapsto [N]$ that satisfies all the constraints inside U. The labeling has a very strong consistency property that we do not describe here. This property ensures that for any subset $W \subseteq U$ (it also has a low diameter), the randomized labeling $\tau : W \mapsto [N]$ is same as $(\sigma : U \mapsto [N])|_W$ in distribution. In other

words, we construct mutually consistent Sherali-Adams distributions $D(U)$ for all sets U having low ρ-diameter.

However, the Sherali-Adams relaxation requires us to define a randomized labeling $D(U)$ for *every* set of size at most t. Here is a natural idea: for an arbitrary set U, partition it (possibly in a randomized way) into sets of *low* ρ-diameter (call these clusters), and then label each cluster as earlier. Such partitioning schemes are well-known in the literature on metric embeddings. For us, the issue however is the consistency between sets. For $W \subseteq U$, we desire that the partition of W on its own is same as partition of W induced by a partition of U (in distribution if the partitioning scheme is randomized). At this point, we observe that the metric ρ can be chosen to be an ℓ_2 metric on points of a unit sphere. The sphere has unrestricted dimension, but if look only at a set $U \subseteq V$, $|U| \leq t$, then U can be thought of as embedded onto $(t-1)$-dimensional unit sphere $\mathbb{S}^{t-1}$ via a random orthogonal transformation. Now we partition $\mathbb{S}^{t-1}$ into clusters with low diameter using a well-known partitioning scheme and that automatically gives a partition of U into low diameter clusters. Since the partition of U depends only on its ℓ_2 geometry, it follows that if $W \subseteq U$, then partition of W is consistent in distribution with that induced from a partition of U!

A somewhat magical part is coming up with the ℓ_2 metric ρ. It turns out that the metric can be constructed from the SDP solution to the Unique Games instance. The solution consists (up to a normalization) of an orthonormal tuple $\{\mathbf{T}_{u,j}\}_{j\in[N]}$ for every vertex $u \in V$. Roughly speaking, desired metric ρ should capture the closeness between these tuples. Defining a single unit vector $\mathbf{T}_u$ from the tuple by $\mathbf{T}_u := \frac{1}{\sqrt{N}} \sum_{j\in[N]} \mathbf{T}_{u,j}^{\otimes 4}$, the ℓ_2 metric $\|\mathbf{T}_u - \mathbf{T}_v\|$ captures the closeness between tuples. This is the metric ρ that we desire.

Sherali-Adams solution to Maximum Cut

It is quite straightforward to translate the t-round Sherali-Adams solution for the Unique Games instance $\mathcal{U}$ into a t-round Sherali-Adams solution for the Maximum Cut instance $\mathcal{I}$. In the reduction of [54, 48], a Unique Games vertex is replaced by a N-dimensional boolean hypercube where the N labels correspond to the N dimensions of the hypercube.

Roughly speaking, the Sherali-Adams solution to Unique Games instance defines a labeling to its vertices. Each label corresponds to a dimension of a hypercube and the hypercube can be cut along that dimension. This yields Sherali-Adams solution for the Maximum Cut instance.

Approximately consistent SDP solution to Maximum Cut

In a similar way to [54], the vector solution for $\mathcal{U}$ can be transformed into one for $\mathcal{I}$ via a certain tensoring operation. We need to ensure that for the instance $\mathcal{I}$, the Sherali-Adams solution at the second level and the SDP vector solution are consistent, at least approximately. Unfortunately, we do not know whether this is true. We get around this problem in the following manner (which is possibly another place where some magic happens):

The Sherali-Adams solution for the Unique Games instance (and therefore the Maximum Cut instance) is parameterized by r, that specifies how *low* the diameter of the clusters is. On the other hand, the SDP solution for Maximum Cut instance is parameterized by an integer s, that specifies how *large* a tensor power is used. We appropriately choose a large number of pairs $\{(r_i, s_i)\}_{i=1}^{\Delta}$. For every choice of index i, we have a Sherali-Adams solution and the SDP solution parameterized by the diameter parameter r_i and the tensor-power parameter s_i. Finally, we define overall Sherali-Adams and SDP solutions to be the *combinations* of i^{th} solutions for $i \in \{1, \ldots, \Delta\}$. The crux of our argument is to show that for all but two values of $i \in [\Delta]$, the i^{th} Sherali-Adams and SDP solutions are almost consistent. Choosing Δ large, we see that the overall Sherali-Adams and SDP solutions are almost (i.e. approximately) consistent.

Correction step

Finally we massage the Sherali-Adams and the SDP solutions for Maximum Cut and ensure that the two are perfectly consistent with each other. The change in the LP/SDP objective value is negligible.

Organization of the chapter. In Section 5.2, we formally define the problems Unique Games, Maximum Cut and Sparsest Cut, describe the relaxations we consider, and

state our results. In Section 5.3, we describe the construction of *local* labelings to sets of Unique Games vertices with low diameter under the appropriate metric ρ. In Section 5.4, the Maximum Cut instance is derived from Unique Games instance via the same reduction as in [54]. Section 5.5 contains the construction of Sherali-Adams and SDP solutions to the Maximum Cut instance that are approximately consistent. In Section 5.6 the approximate solution is transformed to a feasible one and the value of the integrality gap is computed. The construction for Sparsest Cut is very similar to the one for Maximum Cut and we only sketch it in Section 5.7.

5.2 Preliminaries

We first formally define the Maximum Cut, Sparsest Cut, and Unique Games problems.

Definition 5.2.1. *(*Maximum Cut*) For a weighted graph $G = (V, E)$ with non-negative weights $\mathbf{wt}(e)$ for each edge $e \in E$, the goal is to find a cut that maximizes the weight of crossing edges, i.e. to maximize the following objective function,*

$$\max_{\emptyset \neq S \subseteq V} \sum_{e \in E(S,\overline{S})} \mathbf{wt}(e).$$

Definition 5.2.2. *(*non-uniform Sparsest Cut*) Given a graph $G = (V, E)$ with non-negative weights $\mathbf{wt}(e)$ and demands $\mathbf{dem}(e)$ for each edge, the goal is to find a cut to minimize the following,*

$$\min_{\emptyset \neq S \subseteq V} \frac{\sum_{e \in E(S,\overline{S})} \mathbf{wt}(e)}{\sum_{e \in E(S,\overline{S})} \mathbf{dem}(e)}.$$

Definition 5.2.3. *An instance of* Unique Games *$\mathcal{U}(G(V, E), [N], \{\pi_e\}_{e \in E})$ is a constraint satisfaction problem. For every edge $e = (u, v)$ in the graph, there is a bijection $\pi_e : [N] \mapsto [N]$ on the label set $[N]$. A labeling $\sigma : V \mapsto [N]$ satisfies an edge $e = (u, v) \in E$ iff $\pi_e(\sigma(u)) = \sigma(v)$. The goal is to find a labeling that satisfies maximum fraction of edges.*

The Unique Games Conjecture of Khot [47] states the following:

Conjecture 5.2.1. *For arbitrarily small constants $\varepsilon, \delta > 0$, there is a positive integer $N = N(\varepsilon, \delta)$ such that, given an instance $\mathcal{U}$ of* Unique Games *with label set $[N]$, it is*

NP-*hard to distinguish between the following two cases:*

- YES Case*: There is a labeling to the vertices of* $\mathcal{U}$ *that satisfies at least* $1-\varepsilon$ *fraction of the edges.*
- NO Case*: There is no labeling that satisfies even* δ *fraction of the edges of* $\mathcal{U}$*.*

Let $\mathcal{U}$ be the instance as described in Definition 5.2.3. Figure 1 gives a natural SDP relaxation `SDP-UG`. The relaxation is over the vector variables $\mathbf{x}_{u,i}$ for every vertex u of the graph G and label $i \in [N]$. Regarding the integrality gap of this relaxation, Khot and

$$\max \sum_{e=(u,v)\in E} \sum_{i\in[N]} \left\langle \mathbf{x}_{u,i}, \mathbf{x}_{v,\pi_e(i)} \right\rangle$$

Subject to,

$$\forall u \in V \qquad \sum_{i\in[N]} \|\mathbf{x}_{u,i}\|^2 = 1 \quad (1)$$

$$\forall u \in V,\ i,j \in [N],\ i \neq j \qquad \langle \mathbf{x}_{u,i}, \mathbf{x}_{u,j} \rangle = 0 \quad (2)$$

$$\forall u,v \in V,\ i,j \in [N] \qquad \langle \mathbf{x}_{u,i}, \mathbf{x}_{v,j} \rangle \geq 0 \quad (3)$$

Figure 1: Relaxation `SDP-UG` for Unique Games.

Vishnoi [54] proved the following Theorem. We will make use of their gap example.

Theorem 5.2.2. *There is a* Unique Games *instance* $\mathcal{U}_\eta(G(V,E),[N],\{\pi_e\}_{e\in E})$ *where* $\eta > 0$ *is a parameter, such that any labeling to* $\mathcal{U}_\eta$ *satisfies at most* $\frac{1}{N^\eta}$ *fraction of the edges, whereas there exists a solution to the relaxation* `SDP-UG` *with an objective value of at least* $1-4\eta$*.*

5.2.1 Relaxations for Maximum Cut and Sparsest Cut

The relaxation we consider for the Maximum Cut and the Sparsest Cut problems is a combination of a basic SDP and t rounds of the Sherali-Adams LP hierarchy. Let $G = (V, E, \mathbf{wt})$ be a weighted graph.

The relaxation for the MAXIMUM CUT problem, which we denote by SDP-MC(t), is given in Figure 2. The SDP component consists of a unit vector $\mathbf{w}_u$ for every vertex $u \in V$. The LP component consists of, for every set $S \subseteq V,\ |S| \leq t$, a distribution $D(S)$ over $\{-1,1\}$-assignments to S. The distribution is specified by the probabilities $\{x_{S,\sigma} \mid \sigma \in \{-1,1\}^S\}$ and it can be thought of as a distribution on cuts of S. We ensure the consistency between any sets $T \subseteq S,\ |S| \leq t$, i.e. the distribution of cuts on T is same as the one induced by a distribution of cuts on S. Finally, we ensure that the SDP solution is consistent with the LP solution for every set $S = \{u, v\}$ of size two. Specifically, let y_u and y_v be the marginals of the distribution $D(S)$ on $\{-1,1\}^S$ onto the co-ordinates u and v respectively. Constraint (4) states the consistency requirement:

$$\langle \mathbf{w}_u, \mathbf{w}_v \rangle = \mathrm{E}_{D(S)}[y_u y_v].$$

$$\max \sum_{e=\{u,v\} \in E} \mathbf{wt}(e) \frac{1 - \langle \mathbf{w}_u, \mathbf{w}_v \rangle}{2}$$

Subject to,

$$\forall u \in V \qquad \|u\|^2 = 1 \tag{1}$$

$$\forall S \subseteq V, \sigma \in \{-1,1\}^S \text{ s.t. } |S| \leq t \qquad 0 \leq x_{S,\sigma} \leq 1,\ \textstyle\sum_{\sigma \in \{-1,1\}^S} x_{S,\sigma} = 1 \tag{2}$$

$$\forall T \subseteq S \subseteq V, |S| \leq t, \tau \in \{-1,1\}^T \qquad \sum_{\substack{\sigma \in \{-1,1\}^S \\ \sigma|_T = \tau}} x_{S,\sigma} = x_{T,\tau} \tag{3}$$

$$\forall u, v \in V \qquad \sum_{\sigma \in \{-1,1\}^{\{u,v\}}} \sigma(u)\sigma(v) \cdot x_{\{u,v\},\sigma} = \langle \mathbf{w}_u, \mathbf{w}_v \rangle \tag{4}$$

Figure 2: Relaxation SDP-MC(t) for MAXIMUM CUT

For the SPARSEST CUT problem we have an additional parameter $\mathbf{dem}(e)$ for each edge e in the graph. In this case the objective function is the following.

$$\min \frac{\sum_{e=\{u,v\} \in E} \mathbf{wt}(e) \left(\frac{1 - \langle \mathbf{w}_u, \mathbf{w}_v \rangle}{2} \right)}{\sum_{e=\{u,v\} \in E} \mathbf{dem}(e) \left(\frac{1 - \langle \mathbf{w}_u, \mathbf{w}_v \rangle}{2} \right)}$$

We normalize the denominator to 1 and add this as a constraint. Figure 3 gives the relaxation SDP-SC(t) for the Sparsest Cut problem.

$$\min \sum_{e=\{u,v\}\in E} \mathbf{wt}(e)\frac{1-\langle \mathbf{w}_u, \mathbf{w}_v\rangle}{2}$$

Subject to,

$$\forall u \in V \qquad \|u\|^2 = 1 \tag{1}$$

$$\forall S \subseteq V, \sigma \in \{-1,1\}^S \text{ s.t. } |S| \le t \qquad 0 \le x_{S,\sigma} \le 1, \quad \textstyle\sum_{\sigma\in\{-1,1\}^S} x_{S,\sigma} = 1 \tag{2}$$

$$\forall T \subseteq S \subseteq V, |S| \le t, \tau \in \{-1,1\}^T \qquad \sum_{\substack{\sigma\in\{-1,1\}^S \\ \sigma|_T = \tau}} x_{S,\sigma} = x_{T,\tau} \tag{3}$$

$$\forall u, v \in V \qquad \sum_{\sigma\in\{-1,1\}^{\{u,v\}}} \sigma(u)\sigma(v)\cdot x_{\{u,v\},\sigma} = \langle \mathbf{w}_u, \mathbf{w}_v\rangle \tag{4}$$

$$\sum_{e=\{u,v\}\in E} \mathbf{dem}(e)\left(\frac{1-\langle \mathbf{w}_u, \mathbf{w}_v\rangle}{2}\right) = 1 \tag{5}$$

Figure 3: Relaxation SDP-SC(t) for Sparsest Cut

Local ℓ_1-Embeddability: We observe that Constraints (1)-(4) imply that the distance function $d(u,v) := \|\mathbf{w}_u - \mathbf{w}_v\|^2$ defines a metric such that any sub-metric on at most t points is isometrically embeddable into ℓ_1. Indeed, fix any set $S \subseteq V, |S| \le t$. Constraint (4) implies for any pair $u, v \in S$, $\langle \mathbf{w}_u, \mathbf{w}_v\rangle = \mathrm{E}_{D(S)}[y_u y_v]$, where y_u is the marginal of the distribution $D(S)$ onto u. Thus the mapping $u \mapsto y_u$ gives the isometric ℓ_1-embedding of the sub-metric $(S, d(\cdot,\cdot))$.

5.2.2 Our Results

We prove the following two theorems about the integrality gaps of the relaxations SDP-MC(t) and SDP-SC(t). The first theorem, which implies Theorem 1.6.6, is is proved in Sections 5.3 through 5.6, whereas for the proof of the second theorem, which implies Theorem 1.6.6, we

give a brief sketch in Section 5.7.

Theorem 5.2.3. *For all $\varepsilon > 0$, there is an instance $\mathcal{I}$ of* Maximum Cut *on (sufficiently large) n vertices, such that for $t = O((\log\log\log n)^{\frac{1}{6}})$,*

$$\frac{\mathsf{FRAC}(\mathcal{I})}{\mathsf{OPT}(\mathcal{I})} \geq \alpha_{GW}^{-1} - \varepsilon,$$

where $\mathsf{OPT}(\mathcal{I})$ is the optimum value of Maximum Cut *on $\mathcal{I}$, $\mathsf{FRAC}(\mathcal{I})$ is the optimal objective value of $\texttt{SDP-MC}(t)$ on $\mathcal{I}$, and α_{GW} is the Goemans-Williamson constant, i.e. $\alpha_{GW} = \min_{\rho\in[-1,1]} \frac{\arccos(\rho)/\pi}{(1-\rho)/2}$.*

Theorem 5.2.4. *There is an instance $\mathcal{I}$ of* Sparsest Cut *on (sufficiently large) n vertices, such that for $t = O((\log\log\log n)^{\frac{1}{6}})$,*

$$\frac{\mathsf{OPT}(\mathcal{I})}{\mathsf{FRAC}(\mathcal{I})} \geq \Omega((\log\log\log n)^{\frac{1}{13}}),$$

where $\mathsf{OPT}(\mathcal{I})$ is the optimum value of Sparsest Cut *on the instance $\mathcal{I}$, $\mathsf{FRAC}(\mathcal{I})$ is the optimal objective value of $\texttt{SDP-SC}(t)$ on $\mathcal{I}$.*

5.3 Unique Games *Instance*

In this section we state the relevant properties of the Unique Games instance and the corresponding SDP solution constructed by Khot and Vishnoi [54]. For parameters $\eta > 0$ and $N = 2^m$ for some $m \in \mathbb{Z}^+$, Khot and Vishnoi [54] construct the Unique Games instance $\mathcal{U}_\eta(G(V,E),[N],\{\pi_e\}_{e\in E})$ where the number of vertices $|V| = 2^N/N$. [1] The instance has no good labeling, i.e. has low optimum.

Lemma 5.3.1. *Any labeling to the vertices of the* Unique Games *instance $\mathcal{U}_\eta(G(V,E),[N],\{\pi_e\}_{e\in E})$ satisfies at most $\frac{2}{N^\eta}$ fraction of the edges.*

In construction of [54] the elements of $[N]$ are identified with the additive group $(\mathbb{F}[2]^m, \oplus)$. The authors construct a vector solution that consists of unit vectors $\mathbf{T}_{u,i}$ for every vertex $u \in V$ and label $i \in [N]$. These vectors (up to a normalization) form the solution to the Unique Games SDP relaxation $\texttt{SDP-UG}$. We highlight the important properties of the SDP solution below:

[1] For the sake of simplicity, we have slightly altered the presentation from [54].

Properties of the UNIQUE GAMES SDP Solution

- **(Orthonormality)** $\forall\ u \in V,\ \ \forall\ i \neq j \in [N]$,

$$\|\mathbf{T}_{u,i}\| = 1, \quad \langle \mathbf{T}_{u,i}, \mathbf{T}_{u,j} \rangle = 0. \tag{39}$$

- **(Non-negativity)** $\forall\ u, v \in V,\ \ \forall\ i, j \in [N]$,

$$\langle \mathbf{T}_{u,i}, \mathbf{T}_{v,j} \rangle \geq 0. \tag{40}$$

- **(Symmetry)** $\forall\ u, v \in V,\ \ \forall\ i, j, k \in [N]$,

$$\langle \mathbf{T}_{u,i}, \mathbf{T}_{v,j} \rangle = \langle \mathbf{T}_{u,k\oplus i}, \mathbf{T}_{v,k\oplus j} \rangle \tag{41}$$

  where '$\oplus$' is the group operation on $[N]$ as described above.

- **(High SDP Value)** For every edge $e = (v, w) \in E$,

$$\forall\ i \in [N], \quad \left\langle \mathbf{T}_{v,i}, \mathbf{T}_{w,\pi_e(i)} \right\rangle \geq 1 - 4\eta. \tag{42}$$

  In fact, there is $k_e \in [N]$ such that $\ \forall\ i \in [N],\ \ \pi_e(i) = k_e \oplus i$.

We now define for every vertex $u \in V$ a unit vector $\mathbf{T}_u$ as follows (it is a unit vector due to orthonormality condition (39)),

$$\forall u \in V \quad \mathbf{T}_u := \frac{1}{\sqrt{N}} \sum_{i \in [N]} \mathbf{T}_{u,i}^{\otimes 4}. \tag{43}$$

Our main idea is that the Euclidean distances between the vectors $\{\mathbf{T}_u\}_{u \in V}$ are a measure of the 'closeness' between the orthonormal tuples $\{\mathbf{T}_{u,i} \mid i \in [N]\}$. Specifically:

Lemma 5.3.2. *For every $u, v \in V$,*

$$\min_{i,j \in [N]} \|\mathbf{T}_{u,i} - \mathbf{T}_{v,j}\| \ \leq\ \|\mathbf{T}_u - \mathbf{T}_v\| \ \leq\ 2 \cdot \min_{i,j \in [N]} \|\mathbf{T}_{u,i} - \mathbf{T}_{v,j}\|. \tag{44}$$

Proof: Note that

$$\begin{aligned} 1 - \frac{1}{2}\|\mathbf{T}_u - \mathbf{T}_v\|^2 = \langle \mathbf{T}_u, \mathbf{T}_v \rangle &= \left\langle \frac{1}{\sqrt{N}} \sum_{i \in [N]} \mathbf{T}_{u,i}^{\otimes 4}, \frac{1}{\sqrt{N}} \sum_{j \in [N]} \mathbf{T}_{v,j}^{\otimes 4} \right\rangle \\ &= \frac{1}{N} \sum_{i \in [N]} \left(\sum_{j \in [N]} \langle \mathbf{T}_{u,i}, \mathbf{T}_{v,j} \rangle^4 \right). \end{aligned}$$

Due to symmetry (i.e. condition (41)), the inner sum above is the same for every index $i \in [N]$. Therefore fixing some $i_0 \in [N]$,

$$1 - \frac{1}{2}\|\mathbf{T}_u - \mathbf{T}_v\|^2 = \sum_{j\in[N]} \langle \mathbf{T}_{u,i_0}, \mathbf{T}_{v,j}\rangle^4 . \tag{45}$$

Since $\{\mathbf{T}_{v,j}\}_{j\in[N]}$ is an orthonormal set (and using non-negativity condition (40)), we have

$$\sum_{j\in[N]} \langle \mathbf{T}_{u,i_0}, \mathbf{T}_{v,j}\rangle^4 \leq \max_{j\in[N]} \langle \mathbf{T}_{u,i_0}, \mathbf{T}_{v,j}\rangle = 1 - \frac{1}{2}\min_{j\in[N]} \|\mathbf{T}_{u,i_0} - \mathbf{T}_{v,j}\|^2. \tag{46}$$

Combining (45) and (46), we get the left inequality in (44). On the other hand,

$$\begin{aligned}\sum_{j\in[N]} \langle \mathbf{T}_{u,i_0}, \mathbf{T}_{v,j}\rangle^4 &\geq \max_{j\in[N]} \langle \mathbf{T}_{u,i_0}, \mathbf{T}_{v,j}\rangle^4 = \left(1 - \frac{1}{2}\min_{j\in[N]} \|\mathbf{T}_{u,i_0} - \mathbf{T}_{v,j}\|^2\right)^4 \\ &\geq 1 - 2\min_{j\in[N]} \|\mathbf{T}_{u,i_0} - \mathbf{T}_{v,j}\|^2. \end{aligned} \tag{47}$$

Combining (45) and (47), and using symmetry, we get the right inequality in (44). ∎

5.3.1 Local Consistency

Lemma 5.3.3. *Suppose $u, v \in V$ are such that $\|\mathbf{T}_u - \mathbf{T}_v\| \leq \alpha \leq 0.1$. Then there is a unique $k_{u,v} \in [N]$ such that*

$$\forall i \in [N], \quad \|\mathbf{T}_{u,i} - \mathbf{T}_{v,k_{u,v}\oplus i}\| \leq \alpha. \tag{48}$$

Proof: Since $\|\mathbf{T}_u - \mathbf{T}_v\| \leq \alpha$, by Lemma 5.3.2, there exist $i_0, j_0 \in [N]$ such that $\|\mathbf{T}_{u,i_0} - \mathbf{T}_{v,j_0}\| \leq \alpha$. Defining $k_{u,v} = i_0 \oplus j_0$ and using symmetry, we satisfy the hypothesis of the lemma. For the uniqueness, suppose that $k_{u,v}, k'_{u,v}$ both satisfy the hypothesis of the lemma. Then for any i,

$$\|\mathbf{T}_{v,k_{u,v}\oplus i} - \mathbf{T}_{v,k'_{u,v}\oplus i}\| \leq \|\mathbf{T}_{v,k_{u,v}\oplus i} - \mathbf{T}_{u,i}\| + \|\mathbf{T}_{u,i} - \mathbf{T}_{v,k'_{u,v}\oplus i}\| \leq \alpha + \alpha = 2\alpha$$

Since $\{\mathbf{T}_{v,j} \mid j \in [N]\}$ is an orthonormal set, the distance between any two distinct vectors in this set is exactly $\sqrt{2}$. So one must have $k_{u,v} \oplus i = k'_{u,v} \oplus i$ and hence $k_{u,v} = k'_{u,v}$. ∎

Definition 5.3.1. *A set of vertices $V' \subseteq V$ is called* 0.1*-local if* $\forall\, u, v \in V'$, $\|\mathbf{T}_u - \mathbf{T}_u\| \leq 0.1$.

Lemma 5.3.3 states that whenever two vertices u and v are close (in terms of the distance $\|\mathbf{T}_u - \mathbf{T}_v\|$), there is a unique matching $i \mapsto k_{u,v} \oplus i$ such that the orthonormal tuples $\{\mathbf{T}_{u,i} \mid i \in [N]\}$ and $\{\mathbf{T}_{v,j} \mid j \in [N]\}$ are close via this matching. The next lemma shows that for a set V' that is 0.1 local, the matchings induced between every pair of vertices in V' are consistent with each other.

Lemma 5.3.4 (Local Consistency). *Suppose a set V' is 0.1-local and $u, v, w \in V'$. Let $k_{u,v}, k_{u,w}, k_{v,w} \in [N]$ be the elements given by Lemma 5.3.3, i.e. $\forall i \in [N]$,*

$$\|\mathbf{T}_{u,i} - \mathbf{T}_{v,k_{u,v}\oplus i}\| \leq 0.1, \quad \|\mathbf{T}_{u,i} - \mathbf{T}_{w,k_{u,w}\oplus i}\| \leq 0.1, \quad \|\mathbf{T}_{v,i} - \mathbf{T}_{w,k_{v,w}\oplus i}\| \leq 0.1.$$

Then $k_{v,w} = k_{u,v} \oplus k_{u,w}$.

Proof: By triangle inequality,

$$\|\mathbf{T}_{v,i} - \mathbf{T}_{w,k_{u,v}\oplus k_{u,w}\oplus i}\| \leq \|\mathbf{T}_{v,i} - \mathbf{T}_{u,k_{u,v}\oplus i}\| + \|\mathbf{T}_{u,k_{u,v}\oplus i} - \mathbf{T}_{w,k_{u,v}\oplus k_{u,w}\oplus i}\| \leq 0.1 + 0.1 = 0.2.$$

Since $\|\mathbf{T}_{v,i} - \mathbf{T}_{w,k_{v,w}\oplus i}\| \leq 0.1$, it follows that

$$\|\mathbf{T}_{w,k_{v,w}\oplus i} - \mathbf{T}_{w,k_{u,v}\oplus k_{u,w}\oplus i}\| \leq 0.3.$$

Now note that the set $\{\mathbf{T}_{w,j}\}_{j\in[N]}$ is orthonormal, so the distance between any two distinct vectors in this set is exactly $\sqrt{2}$. Therefore one must have $k_{v,w} \oplus i = k_{u,v} \oplus k_{u,w} \oplus i$, and hence $k_{v,w} = k_{u,v} \oplus k_{u,w}$. ∎

5.3.2 Construction of local labelings

Now we construct a (randomized) labeling $L_{V'}$ for any 0.1-local set $V' = \{u_1, \ldots, u_\ell\} \subseteq V$. Choose u_1 as the *pivot* vertex. We pick the label of u_1 to be a random $i \in [N]$ and let the label of every other vertex to be the *mate* of i via the induced matching between u_1 and that vertex. Thanks to Lemma 5.3.4, the labeling $L_{V'}$ does not depend on the choice of the pivot vertex. Formally, the labeling $L_{V'}$ is obtained as:

- Pick one vertex from V', say u_1.
- Choose the label of u_1 to be a random element $i \in [N]$.
- For $2 \leq p \leq \ell$, set the label of u_p to be $i \oplus k_{u_1,u_p}$.

5.3.3 Construction of labelings to arbitrary size-t sets

Let t be the universal parameter denoting the number of levels of Sherali-Adams relaxation our solution satisfies. We will now describe a procedure UG-Label which, given a parameter $r \leq 0.1$ and a subset $U \subseteq V$, $|U| \leq t$, outputs a (randomized) labeling to the vertices of U. Note that U need not be 0.1-local and is completely arbitrary. The idea is to first partition U into clusters such that each cluster is 0.1-local, and then each cluster is labeled according to (local) labeling procedure described in Section 5.3.2. The algorithm UG-Label outputs the partition of U as well, along with a labeling to U.

The following Theorem can be inferred from [36, Theorem 3.2] applied to the Euclidean unit sphere.

Theorem 5.3.5 ([36])**.** *Let $\mathbb{S}^{t-1}$ denote the $(t-1)$ dimensional unit sphere. For every $r > 0$ there is a randomized partition $\tilde{P}(r)$ of $\mathbb{S}^{t-1}$ into disjoint clusters such that,*

1. *For every cluster $\tilde{C} \in \tilde{P}(r)$, $\tilde{C} \subseteq \mathbb{S}^{t-1}$, $\texttt{diam}(\tilde{C}) \leq r$.*
2. *For any pair of points $x, y \in \mathbb{S}^{t-1}$ such that $\|x - y\| = \beta \leq \frac{r}{4}$,*

$$\Pr_{\tilde{P}(r)}\Big[x \text{ and } y \text{ fall into different clusters}\Big] \leq \frac{100\beta t}{r}.$$

Here is our randomized algorithm that outputs a labeling to an arbitrary set $U \subseteq V$ of size at most t, along with its partition into 0.1-local clusters.

Algorithm 5.3.1. UG-Label:

1. Embed the set of at most t unit vectors $\{\mathbf{T}_v \mid v \in U\}$ isometrically into the $(t-1)$-dimensional unit sphere $\mathbb{S}^{t-1}$ via a random orthogonal transformation.
2. Let $\tilde{P}(r)$ be the partition of $\mathbb{S}^{t-1}$ given by Theorem 5.3.5. This naturally induces a partition $P(r)$ of the set U via the above embedding.
3. Since every cluster $\tilde{C} \in \tilde{P}(r)$ has diameter at most 0.1, the corresponding cluster $C \in P(r)$ in the induced partition of U is 0.1-local (C is possibly empty).
4. Label every non-empty cluster $C \in P, C \subseteq U$, with L_C as in Section 5.3.2.

Consistency between sets: For a parameter $r \leq 0.1$, the algorithm UG-Label defines a distribution $D_{UG,r}(U)$ over labelings to the vertices of U, for every subset $U \subseteq V$ such that $|U| \leq t$. From the algorithm it is clear that the labeling to U depends only on the (geometric configuration of the) corresponding vectors $\{\mathbf{T}_u\}_{u \in U}$. It follows that for any two sets $W \subseteq U \subseteq V$ such that $|U| \leq t$, $D_{UG,r}(U)|_W = D_{UG,r}(W)$. Therefore these distributions define a solution to t rounds of Sherali-Adams relaxation for Unique Games.

5.4 Construction of Maximum Cut *Instance*

The Maximum Cut instance is essentially the same as constructed in [54]. We describe it in brief. Let $\rho \in (-1, 0)$ be a parameter[2] and denote the instance constructed as $\mathcal{I}_\rho(V^*, E^*)$. We start with the Unique Games instance $\mathcal{U}_\eta(G(V, E), [N], \{\pi_e\}_{e \in E})$ and replace each vertex $v \in V$ by a block of vertices $(v, \mathbf{x})$ where $\mathbf{x} \in \{-1, 1\}^N$. Thus each block is an N-dimensional boolean hypercube. Let $\mathbf{x} \xleftarrow{p} \{-1, 1\}^N$ denote a random string chosen from the p-biased distribution, i.e. every co-ordinate of $\mathbf{x}$ is chosen independently to be -1 with probability p and 1 with probability $1 - p$.

For every pair of edges $e = (v, w), e' = (v, w') \in E$, there are (all possible) weighted edges between the blocks $(w, \cdot)$ and $(w', \cdot)$ in the instance $\mathcal{I}_\rho(V^*, E^*)$. The weight of an edge e^* between $(w, \mathbf{x})$ and $(w', \mathbf{y})$ is defined as:

$$\mathbf{wt}(e^*) := \Pr_{\substack{\mathbf{z} \xleftarrow{1/2} \{-1,1\}^N \\ \boldsymbol{\mu} \xleftarrow{\frac{1-\rho}{2}} \{-1,1\}^N}} \left[(\mathbf{x} = \mathbf{z} \circ \pi_e^{-1}) \wedge (\mathbf{y} = \mathbf{z}\boldsymbol{\mu} \circ \pi_{e'}^{-1}) \right],$$

where $\mathbf{z} \circ \pi := (z_{\pi(1)}, \ldots, z_{\pi(N)})$. The following theorem is proved in [48, 54].

Theorem 5.4.1. *For any constants $\rho \in (-1, 0)$ and $\lambda > 0$, there is a constant $c(\rho, \lambda)$ such that the following holds: Let $\mathcal{U}_\eta(G(V, E), [N], \{\pi_e\}_{e \in E})$ be an instance of* Unique Games *with $OPT(\mathcal{U}_\eta) \leq c(\rho, \lambda)$, then the corresponding instance $\mathcal{I}_\rho$ as defined above satisfies the property that*

$$\mathsf{OPT}(\mathcal{I}_\rho) \leq \frac{1}{\pi} \arccos \rho + \lambda$$

[2]For the Maximum Cut problem, $\rho < 0$ will be chosen so that $\alpha_{GW} := \min_{\rho \in [-1,1]} \frac{2 \cdot \arccos(\rho)}{\pi(1-\rho)}$ is attained. For the Sparsest Cut problem, $\rho = 1 - \delta$ will be close to 1.

where $\mathsf{OPT}(\mathcal{I}_\rho)$ *is the normalized value of the maximum cut.*

5.5 Construction of Approximate Solution $\mathcal{A}$ to SDP-MC(t)

In this section we will describe the construction of an *approximate* solution for the relaxation SDP-MC(t) for the Maximum Cut instance $\mathcal{I}_\rho(V^*, E^*)$. The parameter t is a superconstant which we shall explicitly define later. Our solution will satisfy all constraints of SDP-MC(t) except for the Constraint (4) which will be satisfied only approximately. More precisely, the solution $\mathcal{A}$ has two components $\mathcal{A} = (D_\mathcal{A}(\cdot), G_\mathcal{A})$ where for every set $S \subseteq V^*$ of size at most t, $D_\mathcal{A}(S)$ is a distribution over $\{-1,1\}$-assignments over S and $G_\mathcal{A}$ is an assignment of unit vectors to V^*. The distributions $D_\mathcal{A}(S)$ satisfy the consistency property of the Sherali-Adams relaxation, i.e. for $T \subseteq S \subseteq V^*$, $|S| \leq t$, we have $D_\mathcal{A}(S) \mid_T = D_\mathcal{A}(T)$. Moreover, the vector solution $G_\mathcal{A}$ is approximately consistent with the Sherali-Adams solution at the second level, i.e. for any two vertices $a, b \in V^*$, if y_a, y_b are the marginals of the distribution $D_\mathcal{A}(\{a,b\})$ on either co-ordinate, then $\mathrm{E}[y_a y_b] \approx \langle G_\mathcal{A}(a), G_\mathcal{A}(b) \rangle$.

We first describe the distributions $D_\mathcal{A}(S)$. This is done in two stages. In the first stage, for a parameter $r \leq 0.1$, we construct distributions $D_{\mathcal{A},r}(S)$ and then in the second stage, we let $D_\mathcal{A}(S)$ to be the average of $D_{\mathcal{A},r_i}(S)$ for appropriately chosen sequence of parameters $\{r_i \mid i \in [\Delta]\}$. We will ensure that the distributions $D_{\mathcal{A},r_i}(S)$ (and therefore their average $D_{\mathcal{A},r}(S)$) satisfy the consistency property of the Sherali-Adams solution.

5.5.1 Construction of the Sherali-Adams solution $D_{\mathcal{A},r}(\cdot)$

Fix a parameter $r \leq 0.1$. For every set $S \subseteq V^*, |S| \leq t$, the distribution $D_{\mathcal{A},r}(S)$ is given by the following algorithm (i.e. the algorithm outputs a $\{-1,1\}$-assignment to S in a randomized manner):

1. Let $U \subseteq V$ be defined as $U := \{v \mid (v, \mathbf{x}) \in S\}$ (recall that V is the set of vertices of the Unique Games instance from which the Maximum Cut instance is derived). Clearly $|U| \leq t$.

2. Run UG-Label(U, r) to obtain a random labeling $\sigma : U \mapsto [N]$ and a partition

$P = P(r)$ of U.

3. For every cluster $C \in P$ choose a value $\omega_C \in \{-1, 1\}$ at random uniformly and independently.

4. For every vertex $(v, \mathbf{x}) \in S$ such that $v \in C$, assign it the value $\mathbf{x}(\sigma(v)) \cdot \omega_C$.

Observe that the distributions $D_{\mathcal{A},r}(\cdot)$ satisfy the consistency property of the Sherali-Adams relaxation. This is inherited from the consistency property of the UG-Label algorithm.

5.5.2 Construction of the Sherali-Adams solution $D_{\mathcal{A}}(\cdot)$.

Let $\Delta := t^4$ and for $i \in [\Delta]$, define a decreasing sequence of radii:

$$r_i = 2^{-it}. \tag{49}$$

For any set $S \subseteq V^*, |S| \leq t$, the following algorithm defines the distribution $D_{\mathcal{A}}(S)$ over $\{-1, 1\}$-assignments to S.

1. Choose a random index $i \in [\Delta]$.

2. Output a random $\{-1, 1\}$-assignment to S according to the distribution $D_{\mathcal{A},r_i}(S)$.

5.5.3 Construction of vector solution $G_{\mathcal{A}}$

Finally we construct the vector solution $G_{\mathcal{A}}$ and show that it is approximately consistent with the Sherali-Adams solution $D_{\mathcal{A}}$ at the second level. For $i \in [\Delta]$, define an increasing sequence of integers s_i as,

$$s_i = 8 \cdot 2^{2it}. \tag{50}$$

Roughly speaking, for every $i \in [\Delta]$, there will be a vector solution $G_{\mathcal{A},s_i}$ parameterized by integer s_i, that approximately agrees with the Sherali-Adams solution $D_{\mathcal{A},r_i}$. However, as it turns out, this is not necessarily true for *every* i, but for *most* $i \in [\Delta]$ (in fact for all but two values). The values of i for which the approximation fails may depend on the pair of vertices under consideration. We will then define the overall vector solution $G_{\mathcal{A}}$ to be

the combination (direct sum) of the solutions $G_{\mathcal{A},s_i}$ for $i \in \Delta$. Since $D_{\mathcal{A}}$ is an average of $D_{\mathcal{A},r_i}$, and $D_{\mathcal{A},r_i}$ approximately agrees with $G_{\mathcal{A},s_i}$ for most $i \in [\Delta]$, it would follow that $D_{\mathcal{A}}$ approximately agrees with $G_{\mathcal{A}}$.

Now we formally describe the construction. Let $(u, \mathbf{x}) \in V^*$ where $u \in V$ is a vertex of the UNIQUE GAMES instance and $\mathbf{x} \in \{-1, 1\}^N$.

For every $i \in [\Delta]$ we define the following unit vector:

Solution $G_{\mathcal{A},s_i}$:

$$\mathbf{G}^i_{(u,\mathbf{x})} := \frac{1}{\sqrt{N}} \sum_{k \in [N]} \mathbf{x}(k) \cdot \mathbf{T}^{\otimes s_i}_{u,k}. \tag{51}$$

Finally, we take direct sum of these vectors to construct the following unit vector:

Solution $G_{\mathcal{A}}$:

$$\mathbf{G}_{(u,\mathbf{x})} := \frac{1}{\sqrt{\Delta}} \left(\bigoplus_{i=1}^{\Delta} \mathbf{G}^i_{(u,\mathbf{x})} \right). \tag{52}$$

The following is the main theorem showing that the vector solution $G_{\mathcal{A}}$ approximately agrees with the Sherali-Adams solution $D_{\mathcal{A}}(\cdot)$ at the second level.

Theorem 5.5.1. *Let $(u, \mathbf{x})$ and $(v, \mathbf{y})$ be any two vertices of V^* where $u, v \in V$ and $\mathbf{x}, \mathbf{y} \in \{-1, 1\}^N$. Let $y_{(u,\mathbf{x})}$ and $y_{(v,\mathbf{y})}$ be the marginals of the $\{-1, 1\}$-assignment to the pair $S = \{(u, \mathbf{x}), (v, \mathbf{y})\}$, either under the distribution $D_{\mathcal{A}}(S)$ or under the distribution $D_{\mathcal{A},r_i}(S)$ (it will be clear from the context). Then,*

$$\left| \mathrm{E}_{D_{\mathcal{A}}} \left[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})} \right] - \left\langle \mathbf{G}_{(u,\mathbf{x})}, \mathbf{G}_{(v,\mathbf{y})} \right\rangle \right| \leq 2 \cdot 2^{-t/2} + \frac{2}{\Delta}. \tag{53}$$

Proof: Since $D_{\mathcal{A}}(\cdot)$ is an average of $D_{\mathcal{A},r_i}(\cdot)$, we have

$$\mathrm{E}_{D_{\mathcal{A}}} \left[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})} \right] = \mathrm{E}_{i \in [\Delta]} \left[\mathrm{E}_{D_{\mathcal{A},r_i}} \left[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})} \right] \right]. \tag{54}$$

Similarly, since the vector (52) is (up to normalization) direct sum of vectors in (51),

$$\left\langle \mathbf{G}_{(u,\mathbf{x})}, \mathbf{G}_{(v,\mathbf{y})} \right\rangle = \mathrm{E}_{i \in [\Delta]} \left[\left\langle \mathbf{G}^i_{(u,\mathbf{x})}, \mathbf{G}^i_{(v,\mathbf{y})} \right\rangle \right]. \tag{55}$$

We want to show that the left hand sides of (54) and (55) are close. We will achieve this by showing that for all but two values of $i \in [\Delta]$, after fixing i, the right hand sides of (54)

and (55) are close, i.e. within $2 \cdot 2^{-t/2}$ of each other. Towards this end, let $r_0 = \sqrt{2}$ and $r_{\Delta+1} = 0$, so that we have a decreasing sequence of radii

$$\sqrt{2} = r_0 > r_1 > \ldots r_\Delta > r_{\Delta+1} = 0.$$

Let $0 \leq p \leq \Delta$ be the unique index such that $r_p \geq \|\mathbf{T}_u - \mathbf{T}_v\| \geq r_{p+1}$. We will show that the right hand sides of (54) and (55) are close except possibly for $i = p, p+1$.

Case 1: $p + 2 \leq i \leq \Delta$.

In this case, we show that the right hand sides of (54) and (55) are essentially zero. First consider the right hand side of (54). The procedure UG-Label with parameter r_i produces clusters with diameter at most r_i and therefore always places u and v into different clusters since $\|\mathbf{T}_u - \mathbf{T}_v\| \geq r_{p+1} > r_i$. Therefore, it outputs labelings to u and v uniformly at random and independent of each other. Moreover, for any cluster C, the variable ω_C is uniformly distributed in $\{-1, 1\}$. Hence, in this case $y_{(u,\mathbf{x})}$ and $y_{(v,\mathbf{y})}$ are independent uniform $\{-1, 1\}$ random variables and therefore,

$$\mathrm{E}_{D_{\mathcal{A},r_i}}[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})}] = 0. \tag{56}$$

Now consider the right hand side of Equation (55). We bound it by e^{-2^t}.

$$\begin{aligned}
\left|\left\langle \mathbf{G}^i_{(u,\mathbf{x})}, \mathbf{G}^i_{(v,\mathbf{y})} \right\rangle\right| &= \left|\left\langle \frac{1}{\sqrt{N}} \sum_{j\in[N]} \mathbf{x}(j) \cdot \mathbf{T}^{\otimes s_i}_{u,j}, \frac{1}{\sqrt{N}} \sum_{\ell\in[N]} \mathbf{y}(\ell) \cdot \mathbf{T}^{\otimes s_i}_{v,\ell} \right\rangle\right| \\
&\leq \frac{1}{N} \sum_{j\in[N]} \left(\sum_{\ell\in[N]} \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle^{s_i} \right)
\end{aligned}$$

By symmetry, the inner sum is the same for every $j \in [N]$, so we may fix some $j_0 \in [N]$. Since $\{\mathbf{T}_{v,\ell} \mid \ell \in [N]\}$ is an orthonormal set

$$\sum_{\ell\in[N]} \langle \mathbf{T}_{u,j_0}, \mathbf{T}_{v,\ell} \rangle^{s_i} \leq \max_{\ell\in[N]} \langle \mathbf{T}_{u,j_0}, \mathbf{T}_{v,\ell} \rangle^{s_i-2} = \left(1 - \frac{1}{2} \min_{\ell\in[N]} \|\mathbf{T}_{u,j_0} - \mathbf{T}_{v,\ell}\|^2\right)^{s_i-2}.$$

The last term can be bounded (using Lemma 5.3.2),

$$\left(1 - \frac{1}{8} \|\mathbf{T}_u - \mathbf{T}_v\|^2\right)^{s_i-2} \leq \left(1 - \frac{1}{8} r^2_{p+1}\right)^{s_{p+2}-2} = \left(1 - \frac{1}{8} 2^{-2(p+1)t}\right)^{8\cdot 2^{2(p+2)t}-2} \leq e^{-2^t}.$$

Case 2: $1 \leq i \leq p-1$.

This case is more subtle. In this case $\|\mathbf{T}_u - \mathbf{T}_v\| \leq r_p \leq r_2 < 0.1$. By Lemma 5.3.3, there is a unique $k^* = k_{u,v}$ such that the orthonormal tuples $\{\mathbf{T}_{u,j} \mid j \in [N]\}$ and $\{\mathbf{T}_{v,\ell} \mid \ell \in [N]\}$ are close via the matching $j \mapsto k^* \oplus j$. In other words,

$$\forall j \in [N], \quad \|\mathbf{T}_{u,j} - \mathbf{T}_{v,k^* \oplus j}\| \leq r_p. \tag{57}$$

We will show that the right hand sides of Equations (54) and (55) are both close to $\frac{1}{N}\sum_{j\in[N]} \mathbf{x}(j) \cdot \mathbf{y}(k^* \oplus j)$.

Towards this end, first consider the right hand side of (54). Let Φ be the event that u and v are not separated into two clusters in the procedure UG-Label$(\{u, v\}, r_i)$. From Theorem 5.3.5 we have,

$$\Pr[\neg\Phi] \;\leq\; \frac{100 \cdot \|\mathbf{T}_u - \mathbf{T}_v\| \cdot t}{r_i} \;\leq \frac{100 \cdot r_p \cdot t}{r_{p-1}} = 100 \cdot 2^{-t} \cdot t \;\leq\; 2^{-t/2}. \tag{58}$$

In the event Φ, both u and v lie in the same cluster C. The procedure UG-Label picks $j \in [N]$ uniformly at random, assigns label $\sigma(u) = j$ and label $\sigma(v) = k^* \oplus j$. In the construction of $D_{\mathcal{A},r_i}$ (see Section 5.5.1), the vertex $(u, \mathbf{x})$ gets assigned $\mathbf{x}(\sigma(u)) \cdot \omega_C = \mathbf{x}(j) \cdot \omega_C$ and the vertex $(v, \mathbf{y})$ gets assigned $\mathbf{y}(\sigma(v)) \cdot \omega_C = \mathbf{y}(k^* \oplus j) \cdot \omega_C$. Therefore,

$$\mathrm{E}_{D_{\mathcal{A},r_i}}[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})} \mid \Phi] = \frac{1}{N} \sum_{j\in[N]} \mathbf{x}(j)\mathbf{y}(k^* \oplus j) \tag{59}$$

And using (58) we obtain,

$$\left| \mathrm{E}_{D_{\mathcal{A},r_i}}[y_{(u,\mathbf{x})} y_{(v,\mathbf{y})}] - \frac{1}{N} \sum_{j\in[N]} \mathbf{x}(j)\mathbf{y}(k^* \oplus j) \right| \;\leq\; 2^{-t/2}. \tag{60}$$

Now consider the right hand side of (55).

$$\begin{aligned}
\left\langle \mathbf{G}^i_{(u,\mathbf{x})}, \mathbf{G}^i_{(v,\mathbf{y})} \right\rangle &= \left\langle \frac{1}{\sqrt{N}} \sum_{j\in[N]} \mathbf{x}(j) \cdot \mathbf{T}_{u,j}^{\otimes s_i}, \frac{1}{\sqrt{N}} \sum_{\ell\in[N]} \mathbf{y}(\ell) \cdot \mathbf{T}_{v,\ell}^{\otimes s_i} \right\rangle \\
&= \left(\frac{1}{N} \sum_{j\in[N]} \mathbf{x}(j)\mathbf{y}(k^* \oplus j) \left\langle \mathbf{T}_{u,j}, \mathbf{T}_{v,k^*\oplus j} \right\rangle^{s_i} \right) \\
&\quad + \frac{1}{N} \sum_{\ell \neq k^* \oplus j} \mathbf{x}(j)\mathbf{y}(\ell) \left\langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \right\rangle^{s_i}.
\end{aligned}$$

We show that the second term is negligible and in the first term, $\langle \mathbf{T}_{u,j}, \mathbf{T}_{v,k^* \oplus j} \rangle^{s_i}$ is essentially equal to 1. This would imply that $\left\langle \mathbf{G}^i_{(u,\mathbf{x})}, \mathbf{G}^i_{(v,\mathbf{y})} \right\rangle$ is very close to $\frac{1}{N} \sum_{j \in [N]} \mathbf{x}(j)\mathbf{y}(k^* \oplus j)$ as desired. Indeed by Equation (57),

$$\begin{aligned} \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,k^* \oplus j} \rangle^{s_i} &= \left(1 - \frac{1}{2} \|\mathbf{T}_{u,j} - \mathbf{T}_{v,k^* \oplus j}\|^2\right)^{s_i} \geq \left(1 - \frac{1}{2} r_p^2\right)^{s_{p-1}} \\ &= \left(1 - \frac{1}{2} 2^{-2pt}\right)^{8 \cdot 2^{2(p-1)t}} \geq 1 - 4 \cdot 2^{-2t}. \end{aligned}$$

On the other hand, if $\ell \neq k^* \oplus j$, then

$$\|\mathbf{T}_{u,j} - \mathbf{T}_{v,\ell}\| \geq |\mathbf{T}_{v,k^* \oplus j} - \mathbf{T}_{v,\ell}\| - \|\mathbf{T}_{v,k^* \oplus j} - \mathbf{T}_{u,j}\| \geq \sqrt{2} - r_p \geq 1,$$

and hence by non-negativity (condition (40)) $0 \leq \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle \leq \frac{1}{2}$. This implies that

$$\left| \frac{1}{N} \sum_{\ell \neq k^* \oplus j} \mathbf{x}(j)\mathbf{y}(\ell) \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle^{s_i} \right| \leq \frac{1}{N} \sum_{j \in [N]} \left(\sum_{\ell \neq k^* \oplus j} \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle^{s_i} \right),$$

and for every $j \in [N]$, since $\{\mathbf{T}_{v,\ell} \mid \ell \neq k^* \oplus j\}$ is an orthonormal set,

$$\sum_{\ell \neq k^* \oplus j} \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle^{s_i} \leq \max_{\ell \neq k^* \oplus j} \langle \mathbf{T}_{u,j}, \mathbf{T}_{v,\ell} \rangle^{s_i - 2} \leq \left(\frac{1}{2}\right)^{s_1 - 2} \leq 2^{-2^t}.$$

Combining everything, we see that the right hand sides of (54) and (55) are within $2 \cdot 2^{-t/2}$ of each other.

Completing the proof of Theorem 5.5.1

Now we can complete the proof of Theorem 5.5.1. We have shown that the right hand sides of (54) and (55) are within $2 \cdot 2^{-t/2}$ of each other for all $i \in \{p+2, \ldots, \Delta\} \cup \{1, \ldots, p-1\}$, i.e. for every $i \in [\Delta]$ except possibly $i = p, p+1$. Clearly, the expressions on (54) and (55) are within $2 \cdot 2^{-t/2} + \frac{2}{\Delta}$ of each other. ∎

We have shown in this section an approximate solution $\mathcal{A}$ to $\texttt{SDP-MC}(t)$ on the instance $\mathcal{I}_\rho$. The solution satisfies all constraints except Constraint (4) of the relaxation, which is only approximately satisfied, up to an error of $2 \cdot 2^{-t/2} + \frac{2}{\Delta}$ that can be made sufficiently small with the choice of t, Δ. In the next section we show how to eliminate this error and obtain the final solution to the relaxation.

5.6 *Final solution $\mathcal{F}$ to* SDP-MC(t)

This section describes the construction of our final feasible solution $\mathcal{F}$ to SDP-MC(t). In the next subsection we first prove a crucial theorem which shows that given an approximate solution of a certain kind to the relaxation SDP-MC(t), it is possible to derive from it a feasible solution to the relaxation with only a negligible loss in the objective value.

5.6.1 Deriving a feasible solution from an approximate solution

The following is the generic theorem we shall require for our construction.

Theorem 5.6.1. *Let $t \in \mathbb{Z}^+$ be any (large enough) parameter and let $\mathcal{I}(V^{\mathcal{I}}, E^{\mathcal{I}})$ be an instance of* Maximum Cut. *Suppose there is a (possibly infeasible) solution $\mathcal{A}$ to the relaxation* SDP-MC(t) *where $\mathcal{A}$ consists of a collection of distributions $\{D_{\mathcal{A}}(S)\}_{S \subseteq V^{\mathcal{I}}, |S| \leq t}$ on $\{-1, 1\}$-assignments to sets of vertices S with size at most t, and a vector solution $G_{\mathcal{A}}$ consisting of unit vectors $\{\mathbf{G}_a\}_{a \in V^{\mathcal{I}}}$. Suppose that for $T \subseteq S \subseteq V^{\mathcal{I}}$, $|S| \leq t$, the distributions $D_{\mathcal{A}}(S)$ and $D_{\mathcal{A}}(T)$ are consistent. Further, for any two vertices $a, b \in V^{\mathcal{I}}$, if y_a and y_b are the marginals of the $\{-1, 1\}$-assignments to $\{a, b\}$ given by the distribution $D_{\mathcal{A}}(\{a, b\})$, then*

$$\left| \mathrm{E}_{D_{\mathcal{A}}}[y_a y_b] - \langle \mathbf{G}_a, \mathbf{G}_b \rangle \right| \leq \frac{1}{t^2}. \tag{61}$$

Then there exists a feasible solution $\mathcal{F}$ to the relaxation SDP-MC(t), *consisting of a collection of distributions $\{D_{\mathcal{F}}(S)\}_{S \subseteq V^{\mathcal{I}}, |S| \leq t}$ and a vector solution $H_{\mathcal{F}}$ of unit vectors $\{\mathbf{H}_a\}_{a \in V^{\mathcal{I}}}$ such that for any two vertices $a, b \in V^{\mathcal{I}}$,*

$$\langle \mathbf{H}_a, \mathbf{H}_b \rangle = \left(1 - O\left(\frac{1}{t}\right)\right) \langle \mathbf{G}_a, \mathbf{G}_b \rangle \tag{62}$$

Proof: We start by defining a collection of distributions $D_{\mathcal{F}}(S)$.

Construction of $D_{\mathcal{F}}(S)$: For every pair of distinct vertices $a, b \in V^{\mathcal{I}}$, we construct a "correcting" distribution $\Gamma_{\{a,b\}}$ over $\{-1, 1\}$-assignments to the set $\{a, b\}$. We will explicitly define these distributions later. We note for now that the marginals of $\Gamma_{a,b}$ on either co-ordinate is uniform.

Let $S \subseteq V^{\mathcal{I}}$ be such that $|S| \leq t$. The distribution $D_{\mathcal{F}}(S)$ on $\{-1,1\}$-assignments to the vertices of S is given by the following randomized procedure:

1. Let $W := S \cup \{z_1, \ldots, z_{t-|S|}\}$ where $z_1, \ldots, z_{t-|S|}$ are dummy vertices. Thus $|W| = t$.
2. From the set W, select uniformly at random a pair of distinct vertices $I = \{w_1, w_2\}$.
3. Using the distribution $D_{\mathcal{A}}(S \setminus I)$, sample a $\{-1,1\}$-assignment γ to vertices of $S \setminus I$.
4. If $I \cap S = \emptyset$, we are done.
5. If $I \cap S = \{a\}$, then the vertex a is assigned a value from $\{-1,1\}$ uniformly at random.
6. If $I \cap S = \{a,b\}$, then the assignment to set $\{a,b\}$ is sampled from the distribution $\Gamma_{\{a,b\}}$.

A case analysis shows that for $T \subseteq S, |S| \leq t$, the distributions $D_{\mathcal{F}}(S)$ and $D_{\mathcal{F}}(T)$ are consistent. One uses the fact that in Step (3), the assignment γ is sampled from $D_{\mathcal{A}}(S \setminus I)$, and these distributions are mutually consistent. Moreover, the marginals of $\Gamma_{a,b}$ are uniform. We skip a formal proof.

Before we define the corresponding vector solution we analyze the distributions $D_{\mathcal{F}}(S)$ corresponding to sets of size two. This analysis will be useful later in the proof.

Analyzing $D_{\mathcal{F}}(S)$ for $|S| = 2$: Let $S = \{a,b\} \subseteq V^{\mathcal{I}}$. Let y_a and y_b denote the marginals of the distribution $D_{\mathcal{F}}(S)$ (or $D_{\mathcal{A}}(S)$ or Γ_S depending on the context). For $J \subseteq S$, let E_J denote the event that $S \cap I = J$, where I is as chosen in Step (2) of the construction of $D_{\mathcal{F}}(S)$. The following are easy to see:

$$\Pr[E_\emptyset] = 1 - \frac{2t-3}{\binom{t}{2}}. \tag{63}$$

$$\Pr\left[E_{\{a\}}\right] = \Pr\left[E_{\{b\}}\right] = \frac{t-2}{\binom{t}{2}}. \tag{64}$$

$$\Pr[E_S] = \frac{1}{\binom{t}{2}}. \tag{65}$$

We also have,

$$\begin{aligned}\mathrm{E}_{D_{\mathcal{F}}(S)}\left[y_a y_b\right] &= \mathrm{E}_{D_{\mathcal{F}}(S)}\left[y_a y_b \mid E_{\emptyset}\right] \cdot \Pr\left[E_{\emptyset}\right] + \mathrm{E}_{D_{\mathcal{F}}(S)}\left[y_a y_b \mid E_{\{a\}}\right] \cdot \Pr\left[E_{\{a\}}\right] \\ &\quad + \mathrm{E}_{D_{\mathcal{F}}(S)}\left[y_a y_b \mid E_{\{b\}}\right] \cdot \Pr\left[E_{\{b\}}\right] + \mathrm{E}_{D_{\mathcal{F}}(S)}\left[y_a y_b \mid E_S\right] \cdot \Pr\left[E_S\right].\end{aligned}$$

If the event $E_{\{a\}}$ occurs then y_a is chosen uniformly at random from $\{-1, 1\}$ independent of y_b and therefore $\mathrm{E}_{D_{\mathcal{F}}(S)}[y_a y_b \mid E_{\{a\}}] = 0$. Similarly, $\mathrm{E}_{D_{\mathcal{F}}(S)}[y_a y_b \mid E_{\{b\}}] = 0$. Moreover, given event $E_{\emptyset}$, $D_{\mathcal{F}}(S)$ is identical to $D_{\mathcal{A}}(S)$. Similarly, given the event E_S, $D_{\mathcal{F}}(S)$ is identical to Γ_S. Therefore,

$$\mathrm{E}_{D_{\mathcal{F}}(S)}[y_a y_b] = \left(1 - \frac{2t-3}{\binom{t}{2}}\right) \mathrm{E}_{D_{\mathcal{A}}(S)}[y_a y_b] + \frac{1}{\binom{t}{2}} \mathrm{E}_{\Gamma_S}[y_a y_b]. \tag{66}$$

Construction of vector solution $H_{\mathcal{F}}$: We now construct the final vector solution $H_{\mathcal{F}}$ as follows.

1. Let $\zeta := 1 - \left(1 - \frac{2(2t-3)}{t(t-1)}\right)$.
2. Construct a pairwise orthonormal set of vectors $\{\mathbf{h}_a\}_{a \in V^{\mathcal{I}}}$ such that for every vertex $a \in V^{\mathcal{I}}$, the vector $\mathbf{h}_a$ is orthogonal to the set of vectors $\{\mathbf{G}_b\}_{b \in V^{\mathcal{I}}}$ comprising the solution $G_{\mathcal{A}}$.
3. In the vector solution $H_{\mathcal{F}}$, for any vertex $a \in V^{\mathcal{I}}$, define the unit vector

$$\mathbf{H}_a := \left(\sqrt{1-\zeta}\right) \mathbf{G}_a + \left(\sqrt{\zeta}\right) \mathbf{h}_a. \tag{67}$$

For any two vertices $a, b \in V^{\mathcal{I}}$, we have that $\mathbf{h}_a \perp \mathbf{h}_b$ and $\mathbf{h}_a$ and $\mathbf{h}_b$ are each orthogonal to both $\mathbf{G}_a$ and $\mathbf{G}_b$. Therefore,

$$\begin{aligned}\langle \mathbf{H}_a, \mathbf{H}_b \rangle &= \left\langle \left(\sqrt{1-\zeta}\right) \mathbf{G}_a + \left(\sqrt{\zeta}\right) \mathbf{h}_a, \left(\sqrt{1-\zeta}\right) \mathbf{G}_b + \left(\sqrt{\zeta}\right) \mathbf{h}_b \right\rangle \\ &= (1-\zeta) \langle \mathbf{G}_a, \mathbf{G}_b \rangle \\ &= \left(1 - O\left(\frac{1}{t}\right)\right) \langle \mathbf{G}_a, \mathbf{G}_b \rangle\end{aligned}$$

which satisfies the desired condition of equation (62).

Finally, we show that there is a way to define the distributions $\Gamma_{\{a,b\}}$ so that the solution $\mathcal{F}$ satisfies the Constraint (4) of the relaxation $\texttt{SDP-MC}(t)$.

Lemma 5.6.2. *For every two distinct vertices $a, b \in V^{\mathcal{I}}$, there is a distribution Γ_S (where $S = \{a, b\}$) such that,*

$$\langle \mathbf{H}_a, \mathbf{H}_b \rangle = \mathrm{E}_{D_{\mathcal{F}}(S)}[y_a y_b]. \tag{68}$$

Proof: From the construction of $H_{\mathcal{F}}$ we have,

$$\begin{aligned} \langle \mathbf{H}_a, \mathbf{H}_b \rangle &= \left\langle \left(\sqrt{1-\zeta}\right) \mathbf{G}_a + \left(\sqrt{\zeta}\right) \mathbf{h}_a, \left(\sqrt{1-\zeta}\right) \mathbf{G}_b + \left(\sqrt{\zeta}\right) \mathbf{h}_b \right\rangle \\ &= (1-\zeta) \langle \mathbf{G}_a, \mathbf{G}_b \rangle . \end{aligned} \tag{69}$$

Equation (66) and substituting the value of ζ in it gives us,

$$\mathrm{E}_{D_{\mathcal{F}}(S)}[y_a y_b] = (1-\zeta)\mathrm{E}_{D_{\mathcal{A}}(S)}[y_a y_b] + \frac{1}{\binom{t}{2}} \mathrm{E}_{\Gamma_S}[y_a y_b].$$

Since we desire that Equation (68) holds, it suffices to set

$$(1-\zeta) \langle \mathbf{G}_a, \mathbf{G}_b \rangle = (1-\zeta)\mathrm{E}_{D_{\mathcal{A}}(S)}[y_a y_b] + \frac{1}{\binom{t}{2}} \mathrm{E}_{\Gamma_S}[y_a y_b], \quad \text{i.e.}$$

$$\mathrm{E}_{\Gamma_S}[y_a y_b] = (1-\zeta)\binom{t}{2} \left(\langle \mathbf{G}_a, \mathbf{G}_b \rangle - \mathrm{E}_{D_{\mathcal{A}}(S)}[y_a y_b] \right).$$

Due to the bound (61), the right hand side above is in $[-1, 1]$. Therefore we can define Γ_S appropriately, with the additional property that its marginal on either co-ordinates is uniform. This completes the proof of Lemma 5.6.2 as well as Theorem 5.6.1. ∎

∎

Applying the above Theorem to the (possibly infeasible) solution $\mathcal{A}$ constructed in Section 5.5, we obtain a feasible solution $\mathcal{F}$ to the relaxation $\texttt{SDP-MC}(t)$ for the instance $\mathcal{I}_\rho$ of Maximum Cut. The solution $\mathcal{F}$ consists of a collection of distributions $\{D_{\mathcal{F}}(S)\}_{S \subseteq V^*, |S| \leq t}$ and a vector solution $H_{\mathcal{F}}$ with unit vectors $\{\mathbf{H}_{(u,\mathbf{x})}\}_{(u,\mathbf{x}) \in V^*}$. The theorem guarantees that for any two vertices $(u, \mathbf{x}), (v, \mathbf{y}) \in V^*$,

$$\left\langle \mathbf{H}_{(u,\mathbf{x})}, \mathbf{H}_{(v,\mathbf{y})} \right\rangle = \left(1 - O\left(\frac{1}{t}\right)\right) \left\langle \mathbf{G}_{(u,\mathbf{x})}, \mathbf{G}_{(v,\mathbf{y})} \right\rangle . \tag{70}$$

5.6.2 Computation of the Integrality Gap

We start by setting the parameters $\eta = (\log N)^{-0.98}$ and $t = (\log\log N)^{\frac{1}{6}}$. The optimum of the UNIQUE GAMES instance $\mathcal{U}_\eta$ is at most $\frac{2}{N^\eta} \leq 2^{-(\log N)^{0.01}}$. The size of $\mathcal{U}_\eta$ is $|V| = 2^N/N$ whereas the size of the MAXIMUM CUT instance $\mathcal{I}_\rho$ is $n := |V| \cdot 2^N = 2^{2N}/N$. The value of $\rho \in (-1, 0)$ is chosen so that $\alpha_{GW}^{-1} := \max_{\rho\in[-1,1]} \frac{\pi(1-\rho)}{2\cdot\arccos(\rho)}$ is attained.

We shall first show the following. Fix a vertex $v \in V$. Let $e(v, w)$ and $e'(v, w') \in E(v)$ any two edges incident on v. Let $\mathbf{x} \in_{1/2} \{-1, 1\}^N$, and $\boldsymbol{\mu} \in_{\frac{1-\rho}{2}} \{-1, 1\}^N$. Then, with probability at least $1 - \eta$,

$$\left\langle \mathbf{H}_{(w,\mathbf{x}\circ\pi_e^{-1})}, \mathbf{H}_{(w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1})} \right\rangle = \rho \pm O\left(\frac{1}{t}\right) \tag{71}$$

Using Chernoff bound we can make the following observation.

Observation 5.6.3. *The following event takes place with probability at least* $1 - \eta$*,*

$$\mathrm{E}_{i\in_R[N]}[\boldsymbol{\mu}(i)] \in [\rho - \eta, \rho + \eta]\,.$$

Using the **High SDP Value** property (condition (42)) of the UNIQUE GAMES SDP solution, we have that for any $\ell \in [N]$,

$$\langle \mathbf{T}_{v,\ell}, \mathbf{T}_{w,k_e\oplus\ell} \rangle \geq 1 - 4\eta, \qquad \left\langle \mathbf{T}_{v,\ell}, \mathbf{T}_{w',k_{e'}\oplus\ell} \right\rangle \geq 1 - 4\eta.$$

From the above and using the triangle inequality, we have,

$$\|\mathbf{T}_{w,k_e\oplus\ell} - \mathbf{T}_{w',k_{e'}\oplus\ell}\| \leq 4\sqrt{2\eta}$$

Using the **Symmetry** property (condition (41)) the above can be restated as follows. For all $\ell \in [N]$,

$$\|\mathbf{T}_{w,\ell} - \mathbf{T}_{w',(k_e\oplus k_{e'})\oplus\ell}\| \leq 4\sqrt{2\eta}. \tag{72}$$

Combining the above with Lemma 5.3.2 we obtain,

$$\|\mathbf{T}_w - \mathbf{T}_{w'}\| \leq 8\sqrt{2\eta}. \tag{73}$$

Combining Equations (72) and (73) with Lemma 5.3.3 we obtain $k_{w,w'} = k_e \oplus k_{e'}$.

From our choice of the parameters $\sqrt{\eta} \ll r_\Delta := 2^{-t^5}$, and thus,

$$r_\Delta > \|\mathbf{T}_w - \mathbf{T}_{w'}\| \geq r_{\Delta+1} = 0, \tag{74}$$

where r_i ($i \in \{0, \ldots, \Delta+1\}$) are as defined in Section 5.5. Moreover, combining Equation (74) with Case 2 of the proof of Theorem 5.5.1, we obtain that for all indices i such that $1 \leq i \leq \Delta - 1$,

$$\left\langle \mathbf{G}^i_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{G}^i_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle = \frac{1}{N} \sum_{j\in[N]} \left(\mathbf{x}\circ\pi_e^{-1}(j)\right)\left(\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}(k_{w,w'}\oplus j)\right) \pm 2^{-t}$$

Since $k_{w,w'} = k_e \oplus k_{e'}$, we can rewrite the above as,

$$\left\langle \mathbf{G}^i_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{G}^i_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle = \frac{1}{N} \sum_{j\in[N]} \left(\mathbf{x}\circ\pi_e^{-1}(k_e\oplus j)\right)\left(\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}(k_{e'}\oplus j)\right) \pm 2^{-t}$$

From the **High SDP Value** property we have that $k_e(j) = \pi_e(j)$ and $k_{e'}(j) = \pi_{e'}(j)$. Substituting in the above equation we get that for all indices $1 \leq i \leq \Delta - 1$,

$$\begin{aligned}
\left\langle \mathbf{G}^i_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{G}^i_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle &= \frac{1}{N} \sum_{j\in[N]} \left(\mathbf{x}\circ\pi_e^{-1}(\pi_e(j))\right)\left(\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}(\pi_{e'}(j))\right) \pm 2^{-t} \\
&= \frac{1}{N} \sum_{j\in[N]} \mathbf{x}(j)\mathbf{x}\boldsymbol{\mu}(j) \pm 2^{-t} \\
&= \frac{1}{N} \sum_{j\in[N]} \boldsymbol{\mu}(j) \pm 2^{-t}
\end{aligned}$$

Since the above holds for all $i \in \{1, \ldots, \Delta-1\}$, by Equation (55) we have,

$$\left\langle \mathbf{G}_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{G}_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle = \frac{1}{N} \sum_{j\in[N]} \boldsymbol{\mu}(j) \pm O\left(\frac{1}{\Delta} + 2^{-t}\right) \tag{75}$$

Combining the above with Observation 5.6.3, we obtain that with probability at least $1-\eta$,

$$\left\langle \mathbf{G}_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{G}_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle = \rho \pm O\left(\frac{1}{\Delta}\right), \tag{76}$$

and from Equation (70),

$$\left\langle \mathbf{H}_{w,\mathbf{x}\circ\pi_e^{-1}}, \mathbf{H}_{w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1}} \right\rangle = \rho \pm O\left(\frac{1}{t}\right)$$

which proves the condition given by Equation (71). Since Equation (71) holds for all $v, e = (v,w)$ and $e' = (v,w')$ this implies that the normalized objective value of $\texttt{SDP-MC}(t)$

on $\mathcal{I}_\rho$ is,

$$\mathsf{FRAC}(\mathcal{I}_\rho) \;\geq\; \mathrm{E}_{\mathbf{x},\boldsymbol{\mu}}\left[\frac{1-\left\langle \mathbf{H}_{(w,\mathbf{x}\circ\pi_e^{-1})}, \mathbf{H}_{(w',\mathbf{x}\boldsymbol{\mu}\circ\pi_{e'}^{-1})}\right\rangle}{2}\right] \;\geq\; \frac{1-\rho}{2} - O\left(\frac{1}{t}\right) - O(\eta).$$

Applying Theorem 5.4.1, by choosing the optimum of $\mathcal{U}_\eta$ (at most $2^{-(\log N)^{0.01}}$) low enough, we see that the (normalized) value of the best cut in $\mathcal{I}_\rho$ is,

$$\mathsf{OPT}(\mathcal{I}_\rho) \leq \frac{1}{\pi}\arccos\rho + \frac{\varepsilon}{4}, \tag{77}$$

where $\varepsilon > 0$ is the constant in Theorem 5.2.3. Therefore, the Integrality Gap of $\texttt{SDP-MC}(t)$ is,

$$\frac{\mathsf{FRAC}(\mathcal{I}_\rho)}{\mathsf{OPT}(\mathcal{I}_\rho)} \;\geq\; \frac{\pi(1-\rho)}{2\cdot\arccos\rho} - O\left(\frac{1}{t}\right) - O(\eta) - \frac{\varepsilon}{2} \;\geq\; \alpha_{GW}^{-1} - \varepsilon.$$

This proves Theorem 5.2.3 and therefore Theorem 1.6.6 (note that $1/t$ and η are sub-constant).

5.7 Integrality gap for Sparsest Cut

We give a brief overview of the construction of the integrality gap example for the Sparsest Cut relaxation $\texttt{SDP-SC}(t)$. As in the construction of Khot and Vishnoi [54], we actually construct an integrality gap example for a similar relaxation for the non-uniform Balanced Separator problem. For this the only change we need to make to the construction for Maximum Cut is the setting of the parameter ρ. We choose set ρ to be $1-\delta$, where $\delta = 1/t$. It was shown in [54] that the instance of Balanced Separator thus obtained has an optimum of $\Omega(\delta^c)$ (any exponent $c > \frac{1}{2}$ works, say $c = \frac{7}{13}$), provided that the soundness of the Unique Games instance is at most $2^{-O(1/\delta^2)}$. On the other hand, the SDP value is at most $O(\delta + 1/t) = O(\delta)$. This gives us an integrality gap of $\Omega((1/\delta)^{1-c})$ which, on substituting the value of the chosen parameters, is $\Omega((\log\log\log n)^{\frac{1}{13}})$.

5.8 Conclusion

We construct integrality gap examples for the Maximum Cut and the Sparsest Cut problems for the standard SDP relaxation augmented with $O((\log\log\log n)^{\frac{1}{6}})$ rounds of

the Sherali-Adams LP relaxation. For MAXIMUM CUT we obtain a gap of $\alpha_{GW}^{-1} - \varepsilon$, for arbitrarily small constant $\varepsilon > 0$, and a gap of $\Omega((\log\log\log n)^{\frac{1}{13}})$ for the SPARSEST CUT problems. The parameters of our construction are weaker than in a contemporaneous work of Raghavendra and Steurer [69] which nevertheless uses techniques similar to our construction.

A natural and important problem is to construct such integrality gap examples for seemingly stronger SDP relaxations, especially the Lassere hierarchy. As mentioned in Section 1.4, a t-round Lassere relaxation includes the basic SDP augmented with t-rounds of Sherali-Adams relaxation and it is conceivable that techniques from our construction may be useful in obtaining integrality gaps for the Lassere hierarchy. Lastly, these questions are intimately related to the Unique Games Conjecture, the resolution of which is one of most important problems in PCP theory.

CHAPTER VI

INTEGRALITY GAP FOR UNIFORM SPARSEST CUT

In this chapter we prove Theorem 1.6.8 giving an integrality gap for the BALANCED SEPARATOR and UNIFORM SPARSEST CUT problems. In the next section we give an overview of the proof of this result.

6.1 Overview

We begin by restating the main theorem we intend to prove.

Theorem. (1.6.8 restated) *The standard SDP relaxations of* UNIFORM SPARSEST CUT *and* BALANCED SEPARATOR *with the triangle inequality constraints, on an n-vertex graph, have an integrality gap of at least* $\Omega(\log \log n)$.

As mentioned in Section 1.6.5, we construct an $\Omega(\log \log n)$ integrality gap for the BALANCED SEPARATOR problem which implies the same integrality gap for UNIFORM SPARSEST CUT as well. The next few paragraphs give an informal description of the construction illustrating the main ideas and techniques involved.

We first highlight how the construction in Theorem 6.2.1 differs from the one in [54] (and also [57]). In these previous works, the vertex set V is partitioned into sets $V_1, V_2, \ldots, V_l$ of roughly equal size. It is not true that their graph does not have "small" balanced cuts. For instance, there are small balanced cuts that place, for every $1 \leq j \leq l$, the entire set V_j on either side of the cut. This issue is handled by introducing the non-uniform version of the BALANCED SEPARATOR problem. They define a "piece-wise balanced cut" as a cut that cuts "many" sets V_i in a balanced manner. Then they show that in their graph, there is no small piece-wise balanced cut. For the non-uniform version of the BALANCED SEPARATOR problem, it suffices to worry only about the piece-wise balanced cuts.

We, however, need to construct a graph $G(V, E)$ that has no small balanced cuts. Here is a simple approach: Start with a hypercube $\mathcal{F} = \{-1, 1\}^N$ and a suitable group action

on the N co-ordinates. The group naturally acts on the set of hypercube vertices and partitions it into "orbits". We merge all vertices that fall into the same orbit. Call the resulting multi-graph $G(V, E)$. Note that the hypercube has small balanced cuts, namely, the dimensionality cuts which cut $1/N$ fraction of the edges. However, if the group is reasonable (e.g. transitive), then $G(V, E)$ does not have small balanced cuts. A balanced cut in the graph $G(V, E)$ corresponds to a balanced boolean function on the hypercube that is invariant on each orbit. Kahn, Kalai and Linial's [42] result says that a balanced function must have a co-ordinate with "influence" at least $\Omega(\log N/N)$, and if the function is invariant under a transitive group action, all co-ordinates have the same influence. Thus, the sum of all influences is $\Omega(\log N)$. This is same as saying that every balanced cut in $G(V, E)$ must cut $\Omega(\log N/N)$ fraction of edges. Note that this lower bound is $\Omega(\log N)$ factor larger than the size of the dimensionality cuts in the hypercube.

Now we outline the SDP solution. We want to assign one unit vector to each orbit. Here is the basic idea: An orbit consists of N elements of $\mathcal{F}$. Pick one element in the orbit as a representative and call it $\mathbf{x}_1$. Thus, all elements in the orbit are given by

$$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$$

where $\mathbf{x}_j$ is $(j-1)^{th}$ cyclic shift of $\mathbf{x}_1$. We view $\mathbf{x}_i$ as vectors with ± 1 co-ordinates and norm $\sqrt{N}$. Roughly speaking, the SDP solution assigns the following vector to the orbit:

$$\mathbf{V} := \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_j^{\otimes \gamma}$$

where γ is some suitable chosen positive integer. Couple of observations: (1) The vector $\mathbf{V}$ does not depend on the choice of the representative $\mathbf{x}_1$. (2) For a typical orbit, the vectors $\mathbf{x}_j, 1 \leq j \leq N$ are almost orthogonal and therefore $\mathbf{V}$ has norm close to 1.

This is only the basic idea and the actual SDP solution we construct (as well as the notation) is somewhat different (see Section 6.3.2).

Organization of the chapter. The formal statements of our integrality gap construction appears in Sections 6.2.2. The formal description of the construction of the integrality gap instance for Balanced Separator appears in Section 6.3. It has two parts: First, showing

that the constructed graph has no small balanced cuts, and second, the constructed SDP solution satisfies the SDP constraints. The first part involves a simple application of the Kahn, Kalai and Linial Theorem [42], and is presented in Section 6.4.1. Construction of the SDP solution is rather technical and all the proofs are deferred to Section 6.4.

6.2 Preliminaries

In this section, we describe the SDP relaxations for Uniform Sparsest Cut, Balanced Separator and Minimum Linear Arrangement problems and give formal statements of our integrality gap constructions.

6.2.1 Uniform Sparsest Cut

Definition 6.2.1 (Uniform Sparsest Cut)**.** *Given a multi-graph $G(V, E)$, the goal is to find a cut $(S, \overline{S})$ that minimizes the following objective called the sparsity of the cut,*

$$\min_{\emptyset \neq S \subsetneq V} \frac{\sum_{e \in E(S,\overline{S})}}{|S||\overline{S}|}.$$

Figure 4 is an SDP relaxation, `SDP-SC` for Uniform Sparsest Cut. To see that this is indeed a relaxation, for any cut $(S, \overline{S})$, consider the following vector assignment: Fix a unit vector $\mathbf{w}$. If $i \in S$, let $\mathbf{v}_i := \mathbf{w}/\sqrt{|S||\overline{S}|}$ and if $i \in \overline{S}$, let $\mathbf{v}_i := -\mathbf{w}/\sqrt{|S||\overline{S}|}$. It is easy to check that this gives a valid SDP solution, and its objective value is equal to the sparsity of the cut.

6.2.2 Balanced Separator

Definition 6.2.2 (Balanced Separator)**.** *For a multi-graph $G = (V, E)$, and a balance parameter $b \in (0, 1/2]$ (to be thought of as a fixed constant), the goal is to find a cut $(S, \overline{S})$ that minimizes $\sum_{e \in E(S,\overline{S})} 1$, subject to $\min\{|S|, |\overline{S}|\} \geq b \cdot |V|$. The cuts that satisfy $\min |S|, |\overline{S}| \geq b|V|$ are called $(b, 1-b)$* **balanced cuts**.

Figure 5 is an SDP relaxation, `SDP-BS`(b), of Balanced Separator with parameter b. To see that this is indeed a relaxation, fix a unit vector $\mathbf{w}$ and let $\mathbf{v}_i := \mathbf{w}$ or $\mathbf{v}_i := -\mathbf{w}$ depending on which side of the cut vertex i belongs to.

$$\min \frac{1}{4} \sum_{e\{i,j\}} \|\mathbf{v}_i - \mathbf{v}_j\|^2$$

Subject to,

$$\forall i,j \in V \qquad \|\mathbf{v}_i\| = \|\mathbf{v}_j\| \qquad (1)$$

$$\forall\ i,j,k \in V \quad \|\mathbf{v}_i - \mathbf{v}_j\|^2 + \|\mathbf{v}_j - \mathbf{v}_k\|^2 \geq \|\mathbf{v}_i - \mathbf{v}_k\|^2 \qquad (2)$$

$$\tfrac{1}{4}\textstyle\sum_{i<j} \|\mathbf{v}_i - \mathbf{v}_j\|^2 = 1 \qquad (3)$$

Figure 4: Relaxation `SDP-USC` for Uniform Sparsest Cut.

The result of Arora, Rao and Vazirani [7] established that the integrality gap of `SDP-BS`(b) (for some constant b) is at-most $O(\sqrt{\log n})$. They further conjectured that the integrality gap is $O(1)$ (for any constant b). We disprove this conjecture by constructing $\Omega(\log\log n)$ integrality gap instance for `SDP-BS`($\frac{1}{3}$) which – by a well known fact (refer, for example, to [54]) – also implies the same gap for the Uniform Sparsest Cut SDP relaxation `SDP-USC`. The following theorem gives a $\Omega(\log\log n)$ integrality gap for the relaxation `SDP-BS`($\frac{1}{3}$) thereby proving Theorem 1.6.8.

$$\min \frac{1}{4} \sum_{e\{i,j\}} \|\mathbf{v}_i - \mathbf{v}_j\|^2$$

Subject to,

$$\forall i \in V \qquad \|\mathbf{v}_i\|^2 = 1 \qquad (1)$$

$$\forall\ i,j,k \in V \quad \|\mathbf{v}_i - \mathbf{v}_j\|^2 + \|\mathbf{v}_j - \mathbf{v}_k\|^2 \geq \|\mathbf{v}_i - \mathbf{v}_k\|^2 \qquad (2)$$

$$\tfrac{1}{4}\textstyle\sum_{i<j} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq b \cdot (1-b) \cdot |V|^2 \qquad (3)$$

Figure 5: Relaxation `SDP-BS`(b) for b-Balanced Separator

Theorem 6.2.1 ($\Omega(\log\log n)$ Integrality Gap Instance for Balanced Separator)**.** *There*

are absolute constants $c_1, c_2 > 0$ such that, for every large enough n, there exists a multi-graph $G(V, E)$ on n vertices, and a vector assignment $i \mapsto \mathbf{v}_i$ for every $i \in V$ such that

1. *Every $(\frac{1}{3}, \frac{2}{3})$ balanced cut must contain at-least $c_1|E|\frac{\log\log n}{\log n}$ edges.*

2. *The vector assignment is a valid solution to the relaxation* SDP-BS($\frac{1}{3}$) *for the* Balanced Separator *problem on G with an objective value,*

$$\frac{1}{4}\sum_{e\{i,j\}} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \leq c_2|E|\frac{1}{\log n}.$$

6.3 Integrality Gap Instance for Balanced Separator

In this section, we present the construction stated in Theorem 6.2.1. Section 6.4.1 proves that the graph $G(V, E)$ has no small balanced cuts.

6.3.1 Constructing The Graph $G(V, E)$

Let N be an integer, which we assume to be prime for the rest of the chapter. [1] Consider the hypercube $\mathcal{F} = \{-1, 1\}^N$. Let $\sigma : \mathcal{F} \mapsto \mathcal{F}$ be the *rotation* operator defined as follows. For an element $\mathbf{u} = (u_1, \ldots, u_N) \in \mathcal{F}$,

$$\sigma((u_1, u_2, \ldots, u_{N-1}, u_N)) := (u_N, u_1, u_2, \ldots, u_{N-1}).$$

Define σ^i recursively as follows: $\sigma^1 := \sigma$, and for all $1 < i \leq N$, $\sigma^i := \sigma \circ \sigma^{i-1}$. This corresponds to applying the σ operator i times. The set of rotations $\mathcal{H} := \{\sigma^i\}_{i=1}^N$ forms a group under composition. Hence, $\mathcal{H}$ partitions $\mathcal{F}$ into *orbits* $\{\mathcal{O}_1, \ldots, \mathcal{O}_m\}$, for some m. Since N is a prime, all but two orbits have N elements each and hence $m = 2 + (2^N - 2)/N$. We recall that for $\mathbf{u}, \mathbf{v} \in \mathcal{F}$, the inner product $\langle \mathbf{u}, \mathbf{v} \rangle := \sum_{i=1}^N u_i v_i$. We now identify certain orbits which have a particularly nice structure: the elements in it are *nearly orthogonal*.

Definition 6.3.1 (Nearly Orthogonal Orbit)**.** *An orbit $\mathcal{O} \in \{\mathcal{O}_1, \ldots, \mathcal{O}_m\}$ is said to be nearly orthogonal if it has N elements and for all $\mathbf{u} \neq \mathbf{v} \in \mathcal{O}$*

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq 8\sqrt{N \log N}.$$

[1] This assumption is not strictly necessary, but makes some of the proofs easier.

Without loss of generality, let $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n\}$ be the set of orbits each of which is nearly orthogonal. The following lemma is a simple consequence of Chernoff Bounds. The proof appears in Section 6.4.2.

Lemma 6.3.1 (Most Orbits are Nearly Orthogonal)**.** *For every large enough N, the number n of nearly orthogonal orbits satisfies $m \geq n \geq (1 - 4/N^2)m$.*

The vertices of $G(V, E)$**.** There is a vertex (call it $\mathcal{O}$) for every orbit $\mathcal{O}$ which is nearly orthogonal, i.e., for every orbit in the set $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n\}$. Henceforth, we will only refer to nearly orthogonal orbits. We use the notation $\mathcal{O} < \mathcal{O}'$ if orbit $\mathcal{O}$ appears before the orbit $\mathcal{O}'$ in the above canonical ordering.

The edges of $G(V, E)$**.** Let $\Delta(\cdot, \cdot)$ denote the Hamming distance. If there are $\mathbf{u} \in \mathcal{O},\ \mathbf{v} \in \mathcal{O}'$ with $\Delta(\mathbf{u}, \mathbf{v}) = 1$, add an edge between $\mathcal{O}$ and $\mathcal{O}'$. Note that if $\Delta(\mathbf{u}, \mathbf{v}) = 1$, then $\Delta(\sigma^j(\mathbf{u}), \sigma^j(\mathbf{v})) = 1$ for every $1 \leq j \leq N$, and hence, there are exactly N edges between $\mathcal{O}$ and $\mathcal{O}'$. Thus, edges in the multi-graph $G(V, E)$ are in one-to-one correspondence with edges in the hypercube $\{-1, 1\}^N$, except for the edges incident on $\{-1, 1\}^N \backslash \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$. Since almost all orbits are orthogonal, $|E| = (1 - O(1/N^2))N \cdot 2^{N-1}$, where $N \cdot 2^{N-1}$ is the number of edges of the hypercube.

The following theorem, proved in Section 6.4.1, establishes that this graph has no small balanced cut. This is a consequence of a Fourier analytic result due to Kahn, Kalai and Linial [42].

Theorem 6.3.2. *There is an absolute constant $c > 0$, such that every $(\frac{1}{3}, \frac{2}{3})$ balanced cut in the graph $G(V, E)$ cuts at-least $c\frac{\log \log n}{\log n}$ fraction of the edges.*

6.3.2 The SDP Solution

We now show how to associate vectors $\mathcal{O} \mapsto \mathbf{V}_{\mathcal{O}}$ to vertices of $G(V, E)$ so that the conditions of Theorem 6.2.1 are satisfied. Fix integers $r = 2^{12}, s = 10$ and $t = 2^{10^6} + 1$. Write any orbit $\mathcal{O}$, as

$$\mathcal{O} = \{\mathbf{V}_{\mathcal{O},1}, \mathbf{V}_{\mathcal{O},2}, \ldots, \mathbf{V}_{\mathcal{O},N}\}$$

where $\mathbf{V}_{\mathcal{O},1}$ is fixed (arbitrarily) as the representative element of the orbit and $\mathbf{V}_{\mathcal{O},j} = \sigma^{j-1}(\mathbf{V}_{\mathcal{O},1})$ for $1 \leq j \leq N$.

Note that the set of vectors $\{\frac{1}{\sqrt{N}}\mathbf{V}_{\mathcal{O},j}\}_{j=1}^{N}$ is a nearly orthogonal set of vectors, each with unit norm (every pairwise dot-product is bounded by $O(\sqrt{\log N/N})$). We take high enough tensor powers of these vectors so that they become even more near-orthogonal. In particular, the vectors

$$\mathbf{T}_{\mathcal{O},j} := \left(\frac{1}{\sqrt{N}}\mathbf{V}_{\mathcal{O},j}\right)^{\otimes r} \qquad 1 \leq j \leq N$$

are unit vectors with pairwise dot-products bounded by $(O(\sqrt{\log N/N}))^r \leq 1/N^{r/3}$. Now we apply Gram-Schmidt orthogonalization process to these vectors and obtain vectors $\mathbf{W}_{\mathcal{O},j}, 1 \leq j \leq N$. Since we apply Gram-Schmidt process on vectors that are already nearly orthogonal, the resulting vectors are very close to the original ones. To be precise, $\|\mathbf{W}_{\mathcal{O},j} - \mathbf{T}_{\mathcal{O},j}\| \leq 1/N^{r/10}$ (Lemma 6.4.6).

We are ready to assign a vector $\mathbf{V}_{\mathcal{O}}$ to the orbit $\mathcal{O}$. Consider the representative element for the orbit $\mathbf{V}_{\mathcal{O},1}$ and define

$$\mathbf{V}_{\mathcal{O}} := \left(\frac{1}{\sqrt{N}}\sum_{j=1}^{N}(\mathbf{W}_{\mathcal{O},j})^{\otimes 4s}\right)^{\otimes t}.$$

Note that $\mathbf{V}_{\mathcal{O}}$ is a unit vector because of orthonormality of vectors $\mathbf{W}_{\mathcal{O},j}, 1 \leq j \leq N$. The following Lemmas 6.3.3, 6.3.4 and 6.3.5 along with Theorem 6.3.2 establish Theorem 6.2.1. The proofs of the lemmas appear in Sections 6.4.3, 6.4.4 and 6.4.5 respectively. The most technical part is proving the triangle inequality. The proof is tedious and proceeds along similar lines as in [54]. We need to keep track of (the negligible) error terms introduced by the Gram-Schmidt orthogonalization process.

Lemma 6.3.3 (Low Objective Value)**.** *There is a fixed constant $c > 0$ such that*

$$\frac{1}{4}\sum_{\substack{e\in E,\\ \mathcal{O},\mathcal{O}' \text{ are endpoints of } e}} \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 \leq c \cdot |E|\frac{1}{\log n}.$$

Lemma 6.3.4 (Well-Separatedness)**.**

$$\frac{1}{4}\sum_{\mathcal{O}<\mathcal{O}'\in V} \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 \geq \frac{2n^2}{9}.$$

Lemma 6.3.5 (Triangle Inequality)**.**

$$\forall\ \mathcal{O}, \mathcal{O}', \mathcal{O}'' \in V,\ \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 + \|\mathbf{V}_{\mathcal{O}'} - \mathbf{V}_{\mathcal{O}''}\|^2 \geq \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}''}\|^2\ .$$

6.4 Proofs

6.4.1 The Instance Has No Small Balanced Cuts

In this section we prove Theorem 6.3.2. Our proof relies on the following Fourier analytic result due to Kahn, Kalai and Linial [42]. First, we need a notion of the *influence*: Let $f : \{-1,1\}^K \to \{-1,1\}$ be a boolean function. Let $\mathbf{e_j} \in \{-1,1\}^K$ be the vector containing -1 at the j-th position and 1 at all other positions. Define $\mathrm{Inf}_j(f) := \Pr_{\mathbf{x}\in_R\{-1,1\}^K}[f(\mathbf{x} \cdot \mathbf{e_j}) \neq f(\mathbf{x})]$. In words, viewing f as a cut in the hypercube, influence of j^{th} co-ordinate equals the fraction of edges along the j^{th} dimension which are cut.

Theorem 6.4.1. *[42] If f is a $(\frac{1}{3}, \frac{2}{3})$ balanced boolean function on $\{-1,1\}^K$, then there is a $j \in [K]$ such that*

$$\mathrm{Inf}_j(f) \geq c\frac{\log K}{K}$$

for some absolute constant $c > 0$.

Proof: [of Theorem 6.3.2] Let $C \subseteq V$ be a $(\frac{1}{3}, \frac{2}{3})$ balanced cut in the instance graph. We need to lower bound the size of this cut. A cut C is viewed as a boolean function $C : V \mapsto \{-1,1\}$. This naturally induces a cut $C' : \{-1,1\}^N \mapsto \{-1,1\}$ as follows: for $\mathbf{u} \in \mathcal{O}_i$, $1 \leq i \leq n$, let $C'(\mathbf{u}) := C(\mathcal{O}_i)$. Without loss of generality assume that C' takes the value -1 more often. For all points $\mathbf{u} \in \{-1,1\}^N \backslash \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$, let $C'(\mathbf{u}) = 1$. This, along with Lemma 6.3.1 ensures that C' is also a $(\frac{1}{3}, \frac{2}{3})$ balanced cut of $\{-1,1\}^N$. Note that C' is a boolean function invariant on each orbit.

For $1 \leq i \leq 2N$, let E_i denote the set of edges of dimension i in C'. Formally, $E_i := \{\{\mathbf{x}, \mathbf{x} \cdot \mathbf{e}_i\} : \mathbf{x} \in \{-1,1\}^N\} \cap C'$. Note that all the E_i's are mutually disjoint. Hence, $|C'| \geq \sum_{i=1}^{2N} |E_i|$. By Theorem 6.4.1, there is a $1 \leq j \leq 2N$, such that $|E_j| = \Omega\left(\frac{2^N \log N}{N}\right)$. Without loss of generality, let $1 \leq j \leq N$. Since the cut C' is invariant on each orbit, the dimensions $\{1, \ldots, N\}$ should all have the same influence on C', and hence, $|E_i| = |E_j|$ for $1 \leq i \leq N$. Hence, $|C'| \geq \Omega\left(2^N \log N\right)$.

Finally, we observe from the construction of $G(V, E)$ in Section 6.3.1 that the edge set E includes all but $O(2^N/N)$ edges of the hypercube $\{-1, 1\}^{2N}$. Thus

$$|C| \geq |C'| - O(2^N/N) = \Omega\left(2^N \log N\right) = \Omega(|E|\frac{\log\log n}{\log n})$$

concluding the proof. ∎

6.4.2 Most Orbits are Orthogonal

In this section we give the proof of Lemma 6.3.1. Recall that an orbit $\mathcal{O}$ is nearly orthogonal if for all $\mathbf{u}, \mathbf{v} \in \mathcal{O}$,

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq 8\sqrt{N \log N}.$$

The following version of Chernoff Bound would be needed for the proof.

Theorem 6.4.2. *If $X_1, X_2, \ldots, X_N$ are independent random variables where each $X_i \in_{1/2} \{-1, 1\}$, then for any $\lambda > 0$*

$$\Pr\left[|X_1 + X_2 + \cdots + X_N| \geq \lambda\sqrt{N}\right] \leq 2\exp(-\lambda^2/4).$$

Note that we have chosen N to be a large odd prime number greater than 3. This ensures that every orbit is of size N, except the ones containing $\mathbf{1}$ and $-\mathbf{1}$. In particular we prove the following lemma which implies Lemma 6.3.1.

Lemma 6.4.3.

$$\Pr_{\mathbf{x} \in_{1/2}\{-1,1\}^N}\left[\exists\ 1 \leq l \leq N, |\langle \mathbf{x}, \sigma^l(\mathbf{x}) \rangle| \geq 8\sqrt{N \log N}\right] \leq 4/N^3.$$

Proof: Let $\mathbf{x} := (x_1 x_2 \ldots x_N)$. Then

$$\begin{aligned}
\langle \mathbf{x}, \sigma(\mathbf{x}) \rangle &= x_1x_2 + x_2x_3 + \cdots + x_{N-1}x_N + x_Nx_1 \\
&= (x_1x_2 + x_3x_4 + \cdots + x_{2i-1}x_{2i} + \cdots + x_{N-2}x_{N-1} + x_Nx_1) \\
&\quad +(x_2x_3 + x_4x_5 + \cdots + x_{2i}x_{2i+1} + \cdots + x_{N-1}x_N) \\
&=: X + Y
\end{aligned}$$

where we let $X := (x_1x_2 + x_3x_4 + \cdots + x_{2i-1}x_{2i} + \cdots + x_{N-2}x_{N-1} + x_Nx_1)$ and $Y := (x_2x_3 + x_4x_5 + \cdots + x_{2i}x_{2i+1} + \cdots + x_{N-1}x_N)$. For a randomly chosen $\mathbf{x}$, X is the sum of

$(N+1)/2$ independent random variables, where each variable is 1 with probability $1/2$, and -1 with probability $1/2$. Similarly, Y is the sum of $(N-1)/2$ such independent random variables. We now analyze the probability that $|\langle \mathbf{x}, \sigma(\mathbf{x})\rangle|$ is greater than $8\sqrt{N \log N}$. All the probabilities are over $\mathbf{x}$ chosen uniformly at random from $\{-1,1\}^N$.

$$\begin{aligned}
\Pr_{\mathbf{x}}[|\langle \mathbf{x}, \sigma(\mathbf{x})\rangle| > 8\sqrt{N \log N}] &= \Pr_{\mathbf{x}}[|X+Y| > 8\sqrt{N \log N}] \\
&\leq \Pr_{\mathbf{x}}[|X| > 4\sqrt{N \log N}] \\
&\quad + \Pr_{\mathbf{x}}[|Y| > 4\sqrt{N \log N}] \\
&\leq 4/N^4.
\end{aligned}$$

The last inequality follows from Theorem 6.4.2. Exactly the same analysis is true for σ^l instead of σ. The lemma follows by taking the union bound over all l's and it implies Lemma 6.3.1 ∎

Before proceeding to the next parts of this section, we state some lemmas that shall be useful later in the section.

Lemma 6.4.4. *If the unit vectors $\mathbf{W}, \mathbf{W}', \mathbf{T}, \mathbf{T}'$ are such that $\|\mathbf{W}-\mathbf{T}\|^2, \|\mathbf{W}'-\mathbf{T}'\|^2 \leq O(1/N)$, and $\langle \mathbf{T}, \mathbf{T}'\rangle \geq 1 - O(1/N)$, then $\langle \mathbf{W}, \mathbf{W}'\rangle \geq 1 - O(1/N)$.*

Proof: It is easy to check that for $a, b, c, d \geq 0$, if $a \leq b+c+d$, then $a^2 \leq 3(b^2+c^2+d^2)$. Therefore, by the triangle inequality on the l_2 norm of the vectors, it follows that

$$\|\mathbf{W}-\mathbf{W}'\|^2 \leq 3(\|\mathbf{W}-\mathbf{T}\|^2 + \|\mathbf{T}-\mathbf{T}'\|^2 + \|\mathbf{T}'-\mathbf{W}'\|^2).$$

Since each term in the RHS is $O(1/N)$, we get that $1 - \langle \mathbf{W}, \mathbf{W}'\rangle = \frac{1}{2}\|\mathbf{W}-\mathbf{W}\|^2 \leq O(1/N)$. ∎

Lemma 6.4.5. *For any two orbits $\mathcal{O}$, $\mathcal{O}'$, for any given j, $k \in [N]$,*

$$\langle \mathbf{T}_{\mathcal{O},j}, \mathbf{T}_{\mathcal{O}',k}\rangle = \langle \mathbf{T}_{\mathcal{O},i+j}, \mathbf{T}_{\mathcal{O}',i+k}\rangle$$

for all $i \in [N]$.

Proof: This follows from the fact that $\mathbf{V}_{\mathcal{O},j+k} = \sigma^k(\mathbf{V}_{\mathcal{O},j})$, and therefore $\langle \mathbf{V}_{\mathcal{O},j}, \mathbf{V}_{\mathcal{O}',k}\rangle = \langle \mathbf{V}_{\mathcal{O},i+j}, \mathbf{V}_{\mathcal{O}',i+k}\rangle$ for all $i \in [N]$. ∎

Lemma 6.4.6. *For any orbit $\mathcal{O}$, $\|\mathbf{W}_{\mathcal{O},i} - \mathbf{T}_{\mathcal{O},i}\| \leq 1/N^{r/10}$ for all $i \in [N]$.*

Proof: This follows from the fact that for any orbit $\mathcal{O}$, for $i \neq j$, $|\langle \mathbf{T}_{\mathcal{O},i},\ \mathbf{T}_{\mathcal{O},j}\rangle| \leq 1/N^{r/3}$. Therefore, applying the Gram-Schmidt orthogonalization process on the N vectors in an orbit does not change the norm of the vectors by more that $1/N^{r/10}$.

A proof of this fact is presented in Section 6.5. ∎

Lemma 6.4.7. *Given any two orbits $\mathcal{O}$ and $\mathcal{O}'$, for any $i,\ j \in [N]$, $|\langle \mathbf{W}_{\mathcal{O},i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}',j}^{\otimes 2}\rangle - \langle \mathbf{T}_{\mathcal{O},i}^{\otimes 2}, \mathbf{T}_{\mathcal{O}',j}^{\otimes 2}\rangle| \leq 1/N^{r/10-1}$ for large enough N.*

Proof: We have the following,

$$\begin{aligned} |\langle \mathbf{W}_{\mathcal{O},i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}',j}^{\otimes 2}\rangle - \langle \mathbf{T}_{\mathcal{O},i}^{\otimes 2}, \mathbf{T}_{\mathcal{O}',j}^{\otimes 2}\rangle| &\leq \left(|\langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j}\rangle - \langle \mathbf{T}_{\mathcal{O},i}, \mathbf{T}_{\mathcal{O}',j}\rangle|\right) \cdot \left(|\langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j}\rangle \right. \\ &\quad \left. + \langle \mathbf{T}_{\mathcal{O},i}, \mathbf{T}_{\mathcal{O}',j}\rangle|\right) \\ &\leq 2|\langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j}\rangle - \langle \mathbf{T}_{\mathcal{O},i}, \mathbf{T}_{\mathcal{O}',j}\rangle| \end{aligned} \tag{78}$$

Now can write $|\langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j}\rangle - \langle \mathbf{T}_{\mathcal{O},i}, \mathbf{T}_{\mathcal{O}',j}\rangle|$ as

$$|\langle \mathbf{T}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j} - \mathbf{T}_{\mathcal{O}',j}\rangle + \langle \mathbf{T}_{\mathcal{O}',j}, \mathbf{W}_{\mathcal{O},i} - \mathbf{T}_{\mathcal{O},i}\rangle + \langle \mathbf{W}_{\mathcal{O}',j} - \mathbf{T}_{\mathcal{O}',j},\ \mathbf{W}_{\mathcal{O},i} - \mathbf{T}_{\mathcal{O},i}\rangle|$$

and apply Lemma 6.4.6 to get the required bound. ∎

6.4.3 Low SDP Optimum

In this section we show that the SDP solution that we constructed has a low optimum (Lemma 6.3.3). We show that if there is an edge between $\mathcal{O}$ and $\mathcal{O}'$, then $\|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 \leq O(1/N)$. Thus

$$\sum_{\substack{e \in E, \\ \mathcal{O},\mathcal{O}' \text{ are endpoints of } e}} \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 \leq O\left(\frac{|E|}{\log n}\right).$$

Recall that there is an edge between $\mathcal{O}$ and $\mathcal{O}'$ if for some k, $\Delta(\mathbf{V}_{\mathcal{O},1}, \mathbf{V}_{\mathcal{O}',k}) = 1$.

Let $\mathbf{u}_i := (\mathbf{W}_{\mathcal{O},i})^{\otimes 4s}$, $\mathbf{v}_i := \left(\mathbf{W}_{\mathcal{O}',i+k-1}\right)^{\otimes 4s}$. Again, $\{\mathbf{u}_i\}_i$ and $\{\mathbf{v}_i\}_i$ are sets of orthonormal vectors. Now by definition, $\mathbf{V}_{\mathcal{O}} = \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{u}_i\right)^{\otimes t}$ and $\mathbf{V}_{\mathcal{O}'} = \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{v}_i\right)^{\otimes t}$. Hence,

$$\langle \mathbf{V}_{\mathcal{O}}, \mathbf{V}_{\mathcal{O}'}\rangle^{1/t} = \left\langle \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{u}_i, \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{v}_i \right\rangle = \frac{1}{N}\left(\sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{v}_i\rangle\right) + \frac{1}{N}\left(\sum_{i \neq j} \langle \mathbf{u}_i, \mathbf{v}_j\rangle\right) \tag{79}$$

We now show that $\langle \mathbf{u}_i, \mathbf{v}_i \rangle = 1 - O(1/N)$ for all i. The fact that $\Delta(\mathbf{V}_{\mathcal{O},1}, \mathbf{V}_{\mathcal{O}',k}) = 1$ implies that $\left\langle \mathbf{V}_{\mathcal{O},i}, \mathbf{V}_{\mathcal{O}',i+k-1} \right\rangle = N - 2$. This implies that $\left\langle \mathbf{T}_{\mathcal{O},i}, \mathbf{T}_{\mathcal{O}',i+k-1} \right\rangle \geq 1 - 2r/N$. Hence, from Lemmas 6.4.6 and 6.4.4 we have

$$\left\langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',i+k-1} \right\rangle \geq 1 - O(1/N) \tag{80}$$

and thus,

$$\langle \mathbf{u}_i, \mathbf{v}_i \rangle \geq 1 - O(1/N). \tag{81}$$

Since $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \geq 0$ for any $i, j \in [N]$, combining the above with Equation (79) we obtain,

$$\langle \mathbf{V}_{\mathcal{O}}, \mathbf{V}_{\mathcal{O}'} \rangle^{1/t} \geq 1 - O(1/N),$$

and hence,

$$\| \mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'} \|^2 = 2(1 - \langle \mathbf{V}_{\mathcal{O}}, \mathbf{V}_{\mathcal{O}'} \rangle) \leq O\left(\frac{1}{N}\right),$$

which gives us the desired bound.

6.4.4 The SDP Solution is "Well-Separated"

In this section we prove Lemma 6.3.4. Let us fix a nearly orthogonal orbit $\mathcal{O}$, and let $\mathbf{V}_{\mathcal{O},1}$ be an element of the orbit. Let $\mathcal{O}'$ be a randomly chosen orbit from $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\} \setminus \{\mathcal{O}\}$. Using the Chernoff bound (Theorem 6.4.2), along with Lemma 6.3.1 it can be deduced that,

$$\Pr_{\mathcal{O}'} \left[\left| \left\langle \mathbf{V}_{\mathcal{O},1}, \mathbf{V}_{\mathcal{O}',j} \right\rangle \right| \leq 16\sqrt{N \log N},\ \forall j \in [N] \right] \geq 1 - 8/N^2.$$

Since the vectors in each orbit are rotations, the above implies that,

$$\Pr_{\mathcal{O}'} \left[\left| \left\langle \mathbf{V}_{\mathcal{O},i}, \mathbf{V}_{\mathcal{O}',j} \right\rangle \right| \leq 16\sqrt{N \log N},\ \forall i, j \in [N] \right] \geq 1 - 8/N^2.$$

This further implies that,

$$\Pr_{\mathcal{O}'} \left[\left| \left\langle \mathbf{T}^{\otimes 2}_{\mathcal{O},i}, \mathbf{T}^{\otimes 2}_{\mathcal{O}',j} \right\rangle \right| \leq 1/N^{r/3},\ \forall i, j \in [N] \right] \geq 1 - 8/N^2.$$

Using Lemma 6.4.7 we obtain,

$$\Pr_{\mathcal{O}'} \left[\left| \left\langle \mathbf{W}^{\otimes 2}_{\mathcal{O},i}, \mathbf{W}^{\otimes 2}_{\mathcal{O}',j} \right\rangle \right| \leq 1/N^{r/11},\ \forall i, j \in [N] \right] \geq 1 - 8/N^2. \tag{82}$$

Therefore, with probability at least $1 - 8/N^2$ over the choice of $\mathcal{O}' \in \{\mathcal{O}_1, \ldots, \mathcal{O}_n\} \setminus \{\mathcal{O}\}$,

$$\begin{aligned}\langle \mathbf{V}_{\mathcal{O}}, \mathbf{V}_{\mathcal{O}'} \rangle &= \left(\frac{1}{N} \sum_{i,j \in [N]} \langle \mathbf{W}_{\mathcal{O},i}, \mathbf{W}_{\mathcal{O}',j} \rangle^{4s} \right)^t \\ &\leq \left(\frac{1}{N^{2sr/11-1}} \right)^t \\ &\leq \frac{1}{N^4}\end{aligned}$$

for our choice of parameters r, s and t. Using the above analysis the following is easy to see,

$$\sum_{\mathcal{O} < \mathcal{O}'} \|\mathbf{V}_{\mathcal{O}} - \mathbf{V}_{\mathcal{O}'}\|^2 \geq (2 - 2/N^4)(1 - 8/N^2) \geq 1, \tag{83}$$

which proves the Well-Separatedness property of Lemma 6.3.4.

6.4.5 The Triangle Inequality

Consider any three orbits $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$. We will prove that the vectors $\mathbf{V}_{\mathcal{O}_1}$, $\mathbf{V}_{\mathcal{O}_2}$ and $\mathbf{V}_{\mathcal{O}_3}$ satisfy the triangle inequality. Recall the definition of these vectors.

$$\begin{aligned}\mathbf{V}_{\mathcal{O}_1} &= \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 4s} \right)^{\otimes t} =: \mathbf{U}_1^{\otimes t}, \\ \mathbf{V}_{\mathcal{O}_2} &= \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{W}_{\mathcal{O}_2,i}^{\otimes 4s} \right)^{\otimes t} =: \mathbf{U}_2^{\otimes t}, \\ \mathbf{V}_{\mathcal{O}_3} &= \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{W}_{\mathcal{O}_3,i}^{\otimes 4s} \right)^{\otimes t} =: \mathbf{U}_3^{\otimes t}.\end{aligned}$$

In this notation, we need to show that

$$1 + \langle \mathbf{U}_2^{\otimes t}, \mathbf{U}_3^{\otimes t} \rangle \geq \langle \mathbf{U}_1^{\otimes t}, \mathbf{U}_2^{\otimes t} \rangle + \langle \mathbf{U}_1^{\otimes t}, \mathbf{U}_3^{\otimes t} \rangle.$$

We can assume that at-least one of the dot-products has magnitude at-least $1/3$, otherwise the inequality trivially holds. Assume, w.l.o.g., that $|\langle \mathbf{U}_1^{\otimes t}, \mathbf{U}_3^{\otimes t} \rangle| \geq 1/3$. This implies that $|\langle \mathbf{U}_1, \mathbf{U}_3 \rangle|^t \geq 1/3$, and therefore, $|\langle \mathbf{U}_1, \mathbf{U}_3 \rangle| = 1 - \eta'$, for some $\eta' = O(1/t)$. Hence,

$$\max_{1 \leq i,j \leq N} |\langle \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,j}^{\otimes 2} \rangle| = 1 - \eta$$

for some $\eta \leq 2^{-100s}$. We may relabel, if necessary, and assume that $\langle \mathbf{W}_{\mathcal{O}_1,1}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,1}^{\otimes 2} \rangle = 1 - \eta$.

Note that we need to show that

$$1 + \langle \mathbf{U}_2, \mathbf{U}_3 \rangle^t \geq \langle \mathbf{U}_1, \mathbf{U}_2 \rangle^t + \langle \mathbf{U}_1, \mathbf{U}_3 \rangle^t.$$

By Lemma 6.4.9 it suffices to show that

$$1 + \langle \mathbf{U}_2, \mathbf{U}_3 \rangle \geq \langle \mathbf{U}_1, \mathbf{U}_2 \rangle + \langle \mathbf{U}_1, \mathbf{U}_3 \rangle.$$

We may assume that no two orbits $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ are the same, otherwise the triangle inequality is trivially satisfied. Therefore for any i, j,

$$\langle \mathbf{T}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{T}_{\mathcal{O}_2,j}^{\otimes 2} \rangle,\ \langle \mathbf{T}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{T}_{\mathcal{O}_3,j}^{\otimes 2} \rangle,\ \langle \mathbf{T}_{\mathcal{O}_2,i}^{\otimes 2}, \mathbf{T}_{\mathcal{O}_3,j}^{\otimes 2} \rangle \leq \left(1 - \frac{1}{N}\right)^{\frac{r}{2}}.$$

Since r is large enough, applying Lemma 6.4.7 we obtain that

$$\langle \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_2,j}^{\otimes 2} \rangle,\ \langle \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,j}^{\otimes 2} \rangle,\ \langle \mathbf{W}_{\mathcal{O}_2,i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,j}^{\otimes 2} \rangle \leq 1 - \frac{1}{N}.$$

As noted before, we may assume that $\langle \mathbf{W}_{\mathcal{O}_1,1}, \mathbf{W}_{\mathcal{O}_3,1} \rangle = 1 - \eta$, and hence, by Lemma 6.4.5 and Lemma 6.4.7, for large enough N,

$$1 - \eta - \frac{1}{N^{r/10-1}} \leq \langle \mathbf{W}_{\mathcal{O}_1,1}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,1}^{\otimes 2} \rangle, \langle \mathbf{W}_{\mathcal{O}_1,2}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,2}^{\otimes 2} \rangle, \cdots, \langle \mathbf{W}_{\mathcal{O}_1,N}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_3,N}^{\otimes 2} \rangle \leq 1 - \eta.$$

Let $\alpha := \max_{1 \leq i,j \leq N} \langle \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_2,j}^{\otimes 2} \rangle$. We may assume, w.l.o.g., that the maximum is achieved for $\mathbf{W}_{\mathcal{O}_1,1},\ \mathbf{W}_{\mathcal{O}_2,1}$ and therefore,

$$\alpha - \frac{1}{N^{r/10-1}} \leq \langle \mathbf{W}_{\mathcal{O}_1,1}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_2,1}^{\otimes 2} \rangle, \langle \mathbf{W}_{\mathcal{O}_1,2}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_2,2}^{\otimes 2} \rangle, \cdots, \langle \mathbf{W}_{\mathcal{O}_1,N}^{\otimes 2}, \mathbf{W}_{\mathcal{O}_2,N}^{\otimes 2} \rangle \leq \alpha.$$

Now, letting $\mathbf{w}_i := \mathbf{W}_{\mathcal{O}_1,i}^{\otimes 2},\ \mathbf{u}_i := \mathbf{W}_{\mathcal{O}_2,i}^{\otimes 2}$, and $\mathbf{v}_i := \mathbf{W}_{\mathcal{O}_3,i}^{\otimes 2}$ the desired inequality follows from Lemma 6.4.11 where Lemma 6.4.8 is used to make sure that the part of the hypothesis which requires that the set $\{\mathbf{W}_{\mathcal{O},i}^{\otimes 2}\}_{\mathcal{O},i}$ satisfies the triangle inequality.

The rest of the this section consists of some lemmas to complete the proof of triangle inequality as illustrated in the above argument.

Lemma 6.4.8. *For any $\mathcal{O}, \mathcal{O}', \mathcal{O}''$ and $i, j, k \in [N]$, $\mathbf{W}_{\mathcal{O},i}^{\otimes 2},\ \mathbf{W}_{\mathcal{O}',j}^{\otimes 2},\ \mathbf{W}_{\mathcal{O}'',k}^{\otimes 2}$ satisfy the triangle inequality.*

Proof: Consider the vectors

$$\mathbf{T}_{\mathcal{O},i} = \left(\frac{1}{\sqrt{N}}\mathbf{V}_{\mathcal{O},i}\right)^{\otimes r}$$
$$\mathbf{T}_{\mathcal{O}',j} = \left(\frac{1}{\sqrt{N}}\mathbf{V}_{\mathcal{O}',j}\right)^{\otimes r}$$
$$\mathbf{T}_{\mathcal{O}'',k} = \left(\frac{1}{\sqrt{N}}\mathbf{V}_{\mathcal{O}'',k}\right)^{\otimes r}.$$

We may assume that no two of these are the same, otherwise the corresponding orthogonalized vectors would also be the same, and therefore the triangle inequality would be trivially valid. Applying Lemma 6.4.10 to the set $U := \{\mathbf{V}_{\mathcal{O},i},\ \mathbf{V}_{\mathcal{O}',j},\ \mathbf{V}_{\mathcal{O}'',k}, -\mathbf{V}_{\mathcal{O},i},\ -\mathbf{V}_{\mathcal{O}',j},\ -\mathbf{V}_{\mathcal{O}'',k}\}$ and $D := N$, and taking the exponent to be $2r$ (r is a power of 2), we obtain

$$N^{2r} + \langle N^r\mathbf{T}^{\otimes 2}_{\mathcal{O}',j},\ N^r\mathbf{T}^{\otimes 2}_{\mathcal{O}'',k}\rangle \ \geq\ \langle N^r\mathbf{T}^{\otimes 2}_{\mathcal{O},i},\ N^r\mathbf{T}^{\otimes 2}_{\mathcal{O}',j}\rangle + \langle N^r\mathbf{T}^{\otimes 2}_{\mathcal{O},i},\ N^r\mathbf{T}^{\otimes 2}_{\mathcal{O}'',k}\rangle + N^{2r-2}.$$

Therefore $1 + \langle \mathbf{T}^{\otimes 2}_{\mathcal{O}',j},\ \mathbf{T}^{\otimes 2}_{\mathcal{O}'',k}\rangle \geq \langle \mathbf{T}^{\otimes 2}_{\mathcal{O},i},\ \mathbf{T}^{\otimes 2}_{\mathcal{O}',j}\rangle + \langle \mathbf{T}^{\otimes 2}_{\mathcal{O},i},\ \mathbf{T}^{\otimes 2}_{\mathcal{O}'',k}\rangle + \frac{1}{N^2}$. Now using Lemma 6.4.7 and the fact that r is a large number, for large enough N we get the desired triangle inequality,

$$1 + \langle \mathbf{W}^{\otimes 2}_{\mathcal{O}',j},\ \mathbf{W}^{\otimes 2}_{\mathcal{O}'',k}\rangle \geq \langle \mathbf{W}^{\otimes 2}_{\mathcal{O},i},\ \mathbf{W}^{\otimes 2}_{\mathcal{O}',j}\rangle + \langle \mathbf{W}^{\otimes 2}_{\mathcal{O},i},\ \mathbf{W}^{\otimes 2}_{\mathcal{O}'',k}\rangle.$$

■

The following lemma is proved in [54], and we reproduce it without proof.

Lemma 6.4.9. *[54] Let $a, b, c \in [-1, 1]$ such that $1 + a \geq b + c$. Then, $1 + a^t \geq b^t + c^t$ for every odd integer $t \geq 1$.*

Lemma 6.4.10. *Let U be a set of vectors in $\mathbb{Z}^m$ that satisfy the following properties:*

1. *$\mathbf{u} \in U \Rightarrow -\mathbf{u} \in U$.*
2. *There is a number D such that $||u||^2 \leq D$ for all $\mathbf{u} \in U$.*
3. *For every $\mathbf{u},\ \mathbf{v},\ \mathbf{w} \in U$,*

$$D + \langle \mathbf{u}, \mathbf{v}\rangle \geq \langle \mathbf{u}, \mathbf{w}\rangle + \langle \mathbf{v}, \mathbf{w}\rangle$$

Then, given any three vectors $\mathbf{u},\ \mathbf{v},\ \mathbf{w} \in U$ *such that* $|\langle \mathbf{u}, \mathbf{v}\rangle|, |\langle \mathbf{u}, \mathbf{w}\rangle|, |\langle \mathbf{v}, \mathbf{w}\rangle| < D$,

$$D^{2^l} + \langle \mathbf{u}^{\otimes 2^l}, \mathbf{v}^{\otimes 2^l}\rangle \geq \langle \mathbf{u}^{\otimes 2^l}, \mathbf{w}^{\otimes 2^l}\rangle + \langle \mathbf{v}^{\otimes 2^l}, \mathbf{w}^{\otimes 2^l}\rangle + D^{2^l-2}$$

for all $l \geq 1$.

Proof: The proof is by induction on l. Let $\mathbf{u}, \mathbf{v}, \mathbf{w}$ be any three vectors in U such that they satisfy the condition that $|\langle \mathbf{u}, \mathbf{v}\rangle|, |\langle \mathbf{u}, \mathbf{w}\rangle|, |\langle \mathbf{v}, \mathbf{w}\rangle| < D$. Let $x := \langle \mathbf{u}, \mathbf{v}\rangle,\ y := \langle \mathbf{u}, \mathbf{w}\rangle,\ z := \langle \mathbf{v}, \mathbf{w}\rangle$.

Base Case: $l = 1$. We need to prove that, $D^2 + x^2 \geq y^2 + z^2 + 1$. It is sufficient to prove this when $|x| \leq |y|$ and $|x| \leq |z|$. Hence, we may assume that $|z| \geq |y| \geq |x|$. Moreover, we may assume that $z \geq 0$. For if $z < 0$, then we argue about the vector $-\mathbf{w}$ instead of $\mathbf{w}$ to get $z \geq 0$. Consider two cases based on the sign of y:

1. $y \geq 0$. We have $D + y \geq z + x$. Since $y \geq 0$, we get that $y \geq x$. By hypothesis we know that $D > z$. Since all the numbers are integers we have that

$$D + y - 1 \geq z + x. \tag{84}$$

   We also have that $D + x \geq z + y$, which implies that

$$D - y \geq z - x. \tag{85}$$

   Since $z + x \geq 0$ and $z - x \geq 0$, we can multiply inequalities (84), (85), and using the fact that $D > y$ we get, $D^2 - y^2 - D + y \geq z^2 - x^2$ which implies that $D^2 + x^2 \geq y^2 + z^2 + D - y \geq y^2 + z^2 + 1$.

2. $y < 0$. In this case $-y \geq -x$. Using this and the fact that $D > z$ we deduce the following inequalities from (85):

$$D + y \geq z + x$$

$$D - y - 1 \geq z - x.$$

   Since $z + x \geq 0$ and $D + y > 0$, multiplying the above two inequalities, we get $D^2 - y^2 - D - y \geq z^2 - x^2$, which implies that $D^2 + x^2 \geq y^2 + z^2 + D + y \geq y^2 + z^2 + 1$.

INDUCTIVE STEP: Assume that the lemma holds for some $k \geq 1$. We need to prove it for $k+1$. As before, we may assume that $|x| \leq |y| \leq |z|$, and hence, it is sufficient to show that $D^{2^{k+1}} + x^{2^{k+1}} \geq y^{2^{k+1}} + z^{2^{k+1}} + D^{2^{k+1}-2}$. By the induction hypothesis we have the following inequalities:

$$D^{2^k} + x^{2^k} \geq z^{2^k} + y^{2^k} + D^{2^k-2} \Rightarrow D^{2^k} - y^{2^k} - D^{2^k-2} \geq z^{2^k} - x^{2^k} \tag{86}$$

$$D^{2^k} + y^{2^k} \geq z^{2^k} + x^{2^k} \tag{87}$$

Observing that right hand sides of inequalities (86) and (87) are non-negative, we multiply both of them to get $D^{2^{k+1}} - y^{2^{k+1}} - D^{2^{k+1}-2} - y^{2^k} D^{2^k-2} \geq z^{2^{k+1}} - x^{2^{k+1}}$. This implies that $D^{2^{k+1}} + x^{2^{k+1}} \geq y^{2^{k+1}} + z^{2^{k+1}} + D^{2^{k+1}-2}$ as desired. ∎

6.4.5.1 Main Lemma

Lemma 6.4.11. *Let $\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{v}_i\}_{i=1}^N$ and $\{\mathbf{w}_i\}_{i=1}^N$ be three sets of vectors, each set being an orthonormal set. Let $s = 10$. For some $\gamma \geq 0$, suppose these vectors satisfy:*

1. **Mild Separation:** *Dot-product of any two vectors is at most $1-\gamma$ in absolute value.*
2. **Triangle Inequality:** *Any three vectors satisfy the triangle inequality.*
3. **Matching Property and Proper Indexing :** *Let $\mu := \max_{1 \leq i,j \leq N} |\langle \mathbf{v}_i, \mathbf{w}_j \rangle|$ and $\lambda := \max_{1 \leq i,j \leq N} |\langle \mathbf{u}_i, \mathbf{w}_j \rangle|$. Then*

$$\mu - \gamma \leq \langle \mathbf{v}_i, \mathbf{w}_i \rangle \leq \mu \quad \forall 1 \leq i \leq N$$

$$\lambda - \gamma \leq \langle \mathbf{u}_i, \mathbf{w}_i \rangle \leq \lambda \quad \forall 1 \leq i \leq N$$

4. **Closeness:** *Either μ or λ is at least equal to $1 - 2^{-100s}$.*

Let $s_i, t_i, r_i \in \{-1, 1\}$ for $1 \leq i \leq N$ and define

$$\mathbf{u} := \frac{1}{\sqrt{N}} \sum_{i=1}^N s_i \mathbf{u}_i^{\otimes 2s}, \quad \mathbf{v} := \frac{1}{\sqrt{N}} \sum_{i=1}^N t_i \mathbf{v}_i^{\otimes 2s}, \quad \mathbf{w} := \frac{1}{\sqrt{N}} \sum_{i=1}^N r_i \mathbf{w}_i^{\otimes 2s}.$$

Then $\mathbf{u}, \mathbf{v}, \mathbf{w}$ satisfy triangle inequality: $1 + \langle \mathbf{u}, \mathbf{v} \rangle \geq \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$.

Proof: Assume w.l.o.g. that $\lambda \leq \mu$. By the Mild Separation property $\mu \leq 1-\gamma$. Define η' such that $\mu - \gamma = 1 - \eta'$. Few observations are in order:

- $1-\eta' = \mu - \gamma \leq 1-\gamma-\gamma = 1-2\gamma$ and hence, $\eta' \geq 2\gamma$. Also, $\mu = 1-\eta'+\gamma \leq 1-\eta'/2$.
- Using the Matching Property we get that

$$1-\eta' \leq \langle \mathbf{v}_i, \mathbf{w}_i \rangle \leq 1-\eta'/2 \quad \forall 1 \leq i \leq N \quad .$$

- $1-\eta' = \mu - \gamma \geq \mu - \eta'/2$. Since $\mu \geq 1 - 2^{-100s}$, we have $\eta' \leq 2^{-99s}$.

We need to show that

$$N + \sum_{i,j=1}^{N} s_i t_j \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s} \geq \sum_{i,j=1}^{N} s_i r_j \langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} + \sum_{i,j=1}^{N} t_i r_j \langle \mathbf{v}_i, \mathbf{w}_j \rangle^{2s}.$$

It suffices to show that for every $1 \leq j \leq N$,

$$1 + \sum_{i=1}^{N} s_i t_j \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s} \geq \sum_{i=1}^{N} s_i r_j \langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} + t_j r_j \langle \mathbf{v}_j, \mathbf{w}_j \rangle^{2s} + \sum_{1 \leq i \leq N, i \neq j} \langle \mathbf{v}_i, \mathbf{w}_j \rangle^{2s}. \quad (88)$$

Fix j henceforth. Write $\langle \mathbf{v}_j, \mathbf{w}_j \rangle = 1 - \eta$ for some $\eta'/2 \leq \eta \leq \eta'$. Thus, $\eta \leq 2^{-40s}$. Note that $\lambda \leq \mu \leq 1 - \eta'/2 \leq 1 - \eta/2$. We consider three cases depending on the value of λ:

$$(1)\ \lambda \leq \eta \quad (2)\ \eta \leq \lambda \leq 1 - \sqrt{\eta} \quad (3)\ 1 - \sqrt{\eta} \leq \lambda \leq 1 - \eta/2$$

(Case 1) $\lambda \leq \eta$ **:** Since $\langle \mathbf{v}_j, \mathbf{w}_j \rangle = 1 - \eta$, and $\sum_{1 \leq i \leq N} \langle \mathbf{v}_i, \mathbf{w}_j \rangle^2 \leq 1$, we have

$$\sum_{1 \leq i \leq N; i \neq j} \langle \mathbf{v}_i, \mathbf{w}_j \rangle^{2s} \leq (2\eta - \eta^2)^s.$$

Also, $\sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} \leq \lambda^{2s-2} \leq \eta^{2s-2}$. Moreover, for any $1 \leq i \leq N$, by the triangle inequality, $1 \pm \langle \mathbf{u}_i, \mathbf{v}_j \rangle \geq \langle \mathbf{v}_j, \mathbf{w}_j \rangle \pm \langle \mathbf{u}_i, \mathbf{w}_j \rangle \geq 1 - \eta - \lambda \geq 1 - 2\eta$, and therefore, $|\langle \mathbf{u}_i, \mathbf{v}_j \rangle| \leq 2\eta$. Therefore, $\sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s} \leq (2\eta)^{2s-2}$. Thus, it suffices to prove that

$$1 \geq (2\eta)^{2s-2} + \eta^{2s-2} + (1-\eta)^{2s} + (2\eta - \eta^2)^s.$$

This is true when $\eta \leq 2^{-40s}$.

(Case 2) $\eta \leq \lambda \leq 1 - \sqrt{\eta}$ **:** We will show that

$$1 + \sum_{i=1}^{N} s_i t_j \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s} \geq \sum_{i=1}^{N} s_i r_j \langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} + t_j r_j \langle \mathbf{v}_j, \mathbf{w}_j \rangle^{2s} + (2\eta - \eta^2)^s. \quad (89)$$

(Subcase i) $t_j \neq r_j$ **:** In this case it suffices to show that

$$1 + (1-\eta)^{2s} \geq \sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s} + \sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} + (2\eta - \eta^2)^s.$$

Again, as before, we have that for every $1 \leq i \leq N$, $|\langle \mathbf{u}_i, \mathbf{w}_j \rangle| \leq \lambda \leq 1 - \sqrt{\eta}$, and $|\langle \mathbf{u}_i, \mathbf{v}_j \rangle| \leq \lambda + \eta \leq 1 - \sqrt{\eta} + \eta$. Thus, it suffices to prove that

$$1 + (1-\eta)^{2s} \geq (1 - \sqrt{\eta} + \eta)^{2s-2} + (1 - \sqrt{\eta})^{2s-2} + (2\eta - \eta^2)^s.$$

This also holds when $\eta \leq 2^{-40s}$.

(Subcase ii) $t_j = r_j$ **:** We need to prove (89). It suffices to show that

$$1 - (1-\eta)^{2s} - (2\eta - \eta^2)^s \;\geq\; \sum_{i=1}^{N} |\langle \mathbf{u}_i, \mathbf{w}_j \rangle^{2s} - \langle \mathbf{u}_i, \mathbf{v}_j \rangle^{2s}| = \sum_{i=1}^{N} |\theta_i^{2s} - \mu_i^{2s}|$$

where $\theta_i := |\langle \mathbf{u}_i, \mathbf{w}_j \rangle|$, $\mu_i := |\langle \mathbf{u}_i, \mathbf{v}_j \rangle|$. Clearly,

$$|\theta_i - \mu_i| \leq |\langle \mathbf{u}_i, \mathbf{v}_j \rangle - \langle \mathbf{u}_i, \mathbf{w}_j \rangle| \leq 1 - \langle \mathbf{v}_j, \mathbf{w}_j \rangle = \eta.$$

Here, we used the assumption that $(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_j)$ satisfy the triangle inequality. Note also that $\max_{1 \leq i \leq N} \theta_i \leq \lambda$ and $\sum_{i=1}^{N} \theta_i^2 \leq 1$. Let $J := \{i \mid \theta_i \leq \eta\}$ and $I := \{i \mid \theta_i \geq \eta\}$. We have,

$$\begin{aligned} \sum_{i=1}^{N} |\theta_i^{2s} - \mu_i^{2s}| \;&\leq\; \sum_{i \in J} (\theta_i^{2s} + \mu_i^{2s}) + \sum_{i \in I} ((\theta_i + \eta)^{2s} - \theta_i^{2s}) \\ &\leq\; (\eta)^{2s-2} + (2\eta)^{2s-2} + \sum_{i \in I} ((\theta_i + \eta)^{2s} - \theta_i^{2s}). \end{aligned}$$

Lemma 6.4.12 implies that the summation on the last line above is bounded by

$$\sum_{l=1}^{2s-2} \binom{2s}{l} \lambda^{2s-l-2} \eta^l + (2s+1)\eta^{2s-2}.$$

Thus, it suffices to show that

$$1 - (1-\eta)^{2s} - (2\eta - \eta^2)^s \;\geq\; \sum_{l=1}^{2s-2} \binom{2s}{l} \lambda^{2s-l-2} \eta^l + (4\eta)^{2s-2}.$$

This is true if

$$2s\eta - \sum_{l=2}^{2s} \binom{2s}{l} \eta^l - (2\eta - \eta^2)^s \geq 2s\lambda^{2s-3}\eta + \sum_{l=2}^{2s} \binom{2s}{l} \eta^l + (4\eta)^{2s-2}.$$

This is true if $2s\eta(1-\lambda^{2s-3}) \geq \eta^2(2^{2s}+2^{2s}+1+4^{2s})$. This is true if $2s\eta\sqrt{\eta} \geq \eta^2 \cdot 4^{2s+1}$, which holds when $\eta \leq 2^{-40s}$. Note that we used the fact that $\lambda \leq 1-\sqrt{\eta}$.

(Case 3) $1-\sqrt{\eta} \leq \lambda \leq 1-\eta/2$ **:** We have $\langle \mathbf{v}_j, \mathbf{w}_j \rangle = 1-\eta$ and

$$1-\eta/2 \geq \lambda \geq \langle \mathbf{u}_j, \mathbf{w}_j \rangle \geq \lambda - \gamma \geq \lambda - \eta'/2 \geq 1-\sqrt{\eta}-\eta$$

Thus $\langle \mathbf{u}_j, \mathbf{w}_j \rangle = 1-\zeta$ for some ζ that satisfies $\eta/2 \leq \zeta \leq \sqrt{\eta}+\eta$. Write $\langle \mathbf{u}_j, \mathbf{v}_j \rangle = 1-\delta$, and by the triangle inequality

$$\eta \leq \zeta + \delta, \quad \delta \leq \eta + \zeta, \quad \zeta \leq \eta + \delta.$$

Thus, to prove (88), it suffices to show that

$$1 + s_j t_j \langle \mathbf{u}_j, \mathbf{v}_j \rangle^{2s} \geq s_j r_j \langle \mathbf{u}_j, \mathbf{w}_j \rangle^{2s} + t_j r_j \langle \mathbf{v}_j, \mathbf{w}_j \rangle^{2s} + (2\eta - \eta^2)^s + (2\zeta - \zeta^2)^s + (2\delta - \delta^2)^s.$$

Depending on signs s_j, t_j, r_j, this reduces to proving one of the three cases:

$$1 + (1-\delta)^{2s} \geq (1-\zeta)^{2s} + (1-\eta)^{2s} + (2\eta - \eta^2)^s + (2\zeta - \zeta^2)^s + (2\delta - \delta^2)^s.$$

$$1 + (1-\eta)^{2s} \geq (1-\zeta)^{2s} + (1-\delta)^{2s} + (2\eta - \eta^2)^s + (2\zeta - \zeta^2)^s + (2\delta - \delta^2)^s.$$

$$1 + (1-\zeta)^{2s} \geq (1-\eta)^{2s} + (1-\delta)^{2s} + (2\eta - \eta^2)^s + (2\zeta - \zeta^2)^s + (2\delta - \delta^2)^s.$$

We will prove the first case, and the remaining two are proved in a similar fashion. We have that $1 + (1-\delta)^{2s} - (1-\zeta)^{2s} - (1-\eta)^{2s}$

$$\begin{aligned}
&\geq 1 + (1-(\zeta+\eta))^{2s} - (1-\zeta)^{2s} - (1-\eta)^{2s} \\
&\geq 2s(2s-1)\cdot\zeta\eta - \sum_{\substack{3\leq i+j\leq 2s \\ i\geq 1, j\geq 1}} \binom{2s}{i+j}\binom{i+j}{i}\zeta^i\eta^j \\
&\geq 2s(2s-1)\zeta\eta - \ 2^{8s}\zeta\eta\cdot\max\{\zeta,\eta,\delta\} \\
&\geq \min\{\zeta\eta, \eta\delta, \zeta\delta\},
\end{aligned}$$

provided that $2^{8s}\max\{\zeta,\eta,\delta\} \leq 1$. Thus, it suffices to have

$$\min\{\zeta\eta, \eta\delta, \zeta\delta\} \geq (2\eta - \eta^2)^s + (2\zeta - \zeta^2)^s + (2\delta - \delta^2)^s.$$

This is clearly true if ζ, η, δ are within a quadratic factor of each other, and $\eta \leq 2^{-40s}$. On the contrary if $\delta < \eta^2$, since we already have $\delta \leq \eta + \zeta$ from the triangle inequality, it reduces to Case 2 by setting η to δ and setting λ to $1 - \eta$.

■

Lemma 6.4.12. *Let η, λ and $\{\theta_i\}_{i=1}^N$ be non-negative reals, such that $\sum_{i=1}^N \theta_i^2 \leq 1+\delta, (0.1 > \delta > 0)$ and for all i, $\eta \leq \theta_i \leq \lambda$. Then*

$$\sum_{i=1}^{N}((\theta_i + \eta)^{2s} - \theta_i^{2s}) \leq \sum_{l=1}^{2s-2} \binom{2s}{l} \lambda^{2s-l-2}\eta^l + (2s+1)\eta^{2s-2}.$$

Proof: Clearly, $N \leq 2/\eta^2$.

$$\begin{aligned}
\sum_{i=1}^{N}(\theta_i + \eta)^{2s} - \theta_i^{2s} &= \sum_{i=1}^{N}\sum_{l=1}^{2s} \binom{2s}{l} \theta_i^{2s-l}\eta^l \\
&= \sum_{l=1}^{2s-2} \binom{2s}{l} \sum_{i=1}^{N} \theta_i^{2s-l}\eta^l + 2s \cdot \left(\sum_{i=1}^{N} \theta_i\right) \eta^{2s-1} + N\eta^{2s} \\
&\leq \sum_{l=1}^{2s-2} \binom{2s}{l} \lambda^{2s-l-2}\eta^l + 2s \cdot \sqrt{N}\eta^{2s-1} + N\eta^{2s} \\
&\leq \sum_{l=1}^{2s-2} \binom{2s}{l} \lambda^{2s-l-2}\eta^l + (2s+2)\eta^{2s-2}.
\end{aligned}$$

■

6.5 Gram-Schmidt Orthogonalization

In this section we prove that, given a set of almost orthogonal unit vectors, the process of orthogonalizing them using the Gram-Schmidt orthogonalization process does not substantially change any of the vectors. Suppose $\{\mathbf{v}_i\}_{i=1}^N$ is a set of unit vectors that are almost orthogonal, i.e., for all $i \neq j$, $|\langle \mathbf{v}_i, \mathbf{v}_j \rangle| \leq \varepsilon$. Moreover, for every $i \in [N]$ $\|\mathbf{v}_i\| = 1$. Let $\{\mathbf{b}_i\}_{i=1}^N$ be defined as (Gram-Schmidt orthogonalization),

$$\mathbf{b}_1 := \mathbf{v}_1, \tag{90}$$

$$\forall\ i \geq 2,\ \mathbf{b}_i := \frac{1}{n_i}\left(\mathbf{v}_i - \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle \mathbf{b}_j\right), \tag{91}$$

$$n_i \text{ is so that } \langle \mathbf{b}_i, \mathbf{b}_i \rangle = 1. \tag{92}$$

It is easy to see that for all $i \neq j, \langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$. Moreover, $n_1 = 1$, and $n_i \leq 1$ for all $i \in [N]$. Define the following quantities in a recursive manner:

$$\varepsilon_1 := \varepsilon \tag{93}$$

$$s_0 := 0 \tag{94}$$

$$\forall\ i \in [N],\ s_i := \sum_{j=1}^{i} \varepsilon_j^2 \tag{95}$$

$$\forall\ i \geq 2,\ \varepsilon_i := \frac{(\varepsilon + s_{i-1})}{(1 - s_{i-1})} \tag{96}$$

We may assume that ε is very small as a function of N, so that ε_i and s_i are bounded far from 1 for all $i \in [N]$. We first prove the following lemma.

Lemma 6.5.1. *The following conditions hold true for all $i \in [N]$:*

1. $1 - n_i^2 \leq s_{i-1}$.
2. $\forall\ i' > i, i' \in [N]\ \ |\langle \mathbf{v}_{i'}, \mathbf{b}_i \rangle| \leq \varepsilon_i$.
3. $\|\mathbf{v}_i - \mathbf{b}_i\|^2 \leq 2 s_{i-1}$.

Proof: The proof proceeds by induction on i, where the hypothesis, for a particular i, consists of the conditions $1, 2$ and 3 of the statement of the lemma.

Base case, $i = 1$, is easy since $\mathbf{b}_1 = \mathbf{v}_1, n_1 = 0$ and $s_0 = 0$, and $|\langle \mathbf{v}_{i'}, \mathbf{b}_1 \rangle| = |\langle \mathbf{v}_{i'}, \mathbf{v}_1 \rangle| \leq \varepsilon = \varepsilon_1$ for all $1 < i' \leq n$.

Suppose that the hypothesis holds for all $j : 1 \leq j < i$. We shall prove that it holds for i as well. We prove each of the three conditions of the lemma separately as below.

1. We prove property 1 of the lemma as follows. From Equation (90) we have,

$$\mathbf{v}_i = n_i \mathbf{b}_i + \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle \mathbf{b}_j.$$

Taking inner product with v_i on both sides we obtain,

$$\langle \mathbf{v}_i, \mathbf{v}_i \rangle = n_i^2 + \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle^2 .$$

Since v_i is a unit vector the above can be rearranged as,

$$\begin{aligned} 1 - n_i^2 &= \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle^2 \\ &\leq s_{i-1}, \end{aligned}$$

where the last inequality follows from Equation (95) and condition 2 of the lemma applied to $j \in \{1, \ldots, i-1\}$. This proves condition 1 of the lemma for i.

2. We now prove property 2 for i. For all $i' > i$, we obtain after rearranging Equation (91) and taking inner product on both sides by $\mathbf{v}_{i'}$,

$$\begin{aligned} n_i \langle \mathbf{v}_{i'}, \mathbf{b}_i \rangle &= \langle \mathbf{v}_{i'}, \mathbf{v}_i \rangle - \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle \langle \mathbf{v}_{i'}, \mathbf{b}_j \rangle \\ \implies n_i |\langle \mathbf{v}_{i'}, \mathbf{b}_i \rangle| &= |\langle \mathbf{v}_{i'}, \mathbf{v}_i \rangle| + \sum_{j=1}^{i-1} |\langle \mathbf{v}_i, \mathbf{b}_j \rangle| |\langle \mathbf{v}_{i'}, \mathbf{b}_j \rangle|. \end{aligned} \tag{97}$$

From the definition of almost orthogonality, $|\langle \mathbf{v}_i, \mathbf{v}_{i'} \rangle| \leq \varepsilon$. Moreover, applying condition 2 of the lemma gives us,

$$|\langle \mathbf{v}_i, \mathbf{b}_j \rangle| |\langle \mathbf{v}_{i'}, \mathbf{b}_j \rangle| \leq \varepsilon_j^2,$$

Therefore, we can rewrite Equation (97) as,

$$n_i |\langle \mathbf{v}_{i'}, \mathbf{b}_i \rangle| \leq \varepsilon + s_{i-1}. \tag{98}$$

Also,

$$\frac{1}{n_i} \leq \frac{1}{n_i^2} \leq \frac{1}{1 - s_{i-1}}.$$

Combining the above with Equation (98),

$$|\langle \mathbf{v}_{i'}, \mathbf{b}_i \rangle| \leq (\varepsilon + s_{i-1})/(1 - s_{i-1}) = \varepsilon_i.$$

which proves condition 2 of the lemma for i.

3. For proving condition 3 of the lemma, we rewrite Equation (91) to obtain,

$$\mathbf{v}_i - \mathbf{b}_i = (n_i - 1)\mathbf{b}_i - \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle \, \mathbf{b}_j.$$

Since, $\mathbf{b}_j$s are orthonormal, taking the ℓ_2^2 norm on both sides we get,

$$\|\mathbf{v}_i - \mathbf{b}_i\|^2 \;=\; (n_i - 1)^2 + \sum_{j=1}^{i-1} \langle \mathbf{v}_i, \mathbf{b}_j \rangle^2 \,. \tag{99}$$

Applying property 1, and since $0 < n_i \leq 1$ we have $s_{i-1} \geq 1 - n_i^2 \geq 1 - n_i \geq (1 - n_i)^2$. Moreover, applying condition 2 for $j = \{1, \ldots, i-1\}$ enables us to bound the second term in the above equation by s_{i-1} as well. Therefore we can rewrite Equation (99) as,

$$\|\mathbf{v}_i - \mathbf{b}_i\|^2 \leq 2s_{i-1},$$

which proves condition 3 for i.

This completes the proof of Lemma 6.5.1. ∎

We now prove a final lemma to bound the total errors, given that ε is small enough. This lemma, along with Lemma 6.5.1 implies Lemma 6.4.6.

Lemma 6.5.2. *Suppose that $\varepsilon \leq 1/N^2$ (for large enough N). Then for all $i \in [N]$,*

1. $\varepsilon_i \leq 4\varepsilon$.
2. $s_i \leq 16i\varepsilon^2$.

Proof: The proof is via induction on i. Clearly, the base case for $i = 1$ is true.

Assume the conditions of the lemma to hold for $i-1$ for some $1 < i \leq N$. Now we have,

$$\varepsilon_i = \frac{(\varepsilon + s_{i-1})}{(1 - s_{i-1})}.$$

Since s_{i-1} is $O(1/N^3)$, for large enough N the above can be bounded as,

$$\varepsilon_i \leq 4\varepsilon,$$

which proves condition 1. Moreover,

$$s_i = s_{i-1} + \varepsilon_i^2 \leq 16(i-1)\varepsilon^2 + 16\varepsilon^2 = 16i\varepsilon^2,$$

which proves condition 2. This completes the inductive proof of the lemma. ∎

6.6 Conclusion

We show a $\Omega(\log \log n)$ integrality gap for the Balanced Separator and Uniform Sparsest Cut SDP relaxations with triangle inequalities. As mentioned in Section 1.6.5, recently Raghavendra and Steurer [69] have built upon this construction and shown an integrality gap of $\Omega(\log \log^{\gamma} n)$ for the SDP relaxation augmented with k-gonal inequalities for $k = O(2^{\log \log^{\delta} n})$, for some constants $\delta, \gamma > 0$.

An important open problem is to bridge the gap between the above lower bounds and the best known upper bound of $O(\sqrt{\log n})$ [7]. Constructing similar integrality gaps for the Lassere hierarchy also remains a challenging open problem.

REFERENCES

[1] Alekhnovich, M., Braverman, M., Feldman, V., Klivans, A., and Pitassi, T., "The complexity of properly learning simple concept classes," *J. Comput. Syst. Sci.*, vol. 74, no. 1, pp. 16–34, 2008.

[2] Allender, E., Hellerstein, L., McCabe, P., Pitassi, T., and Saks, M., "Minimizing DNF formulas and AC^0_d circuits given a truth table," in *Proc. IEEE CCC*, pp. 237–251, 2006.

[3] Ambühl, C., Mastrolilli, M., and Svensson, O., "Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constrained scheduling," in *Proc.* 48^{th} *IEEE FOCS*, pp. 329–337, 2007.

[4] Applebaum, B., Barak, B., and Xiao, D., "On basing lower-bounds for learning on worst case assumptions," in *Proc.* 49^{th} *IEEE FOCS*, pp. 211–220, 2008.

[5] Arora, S., Lee, J. R., and Naor, A., "Euclidean distortion and the sparsest cut," *J. AMS*, vol. 21, no. 1, pp. 1–21, 2008.

[6] Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M., "Proof verification and the hardness of approximation problems," *J. ACM*, vol. 45, no. 3, pp. 501–555, 1998.

[7] Arora, S., Rao, S., and Vazirani, U., "Expander flows, geometric embeddings and graph partitioning," in *Proc.* 36^{th} *ACM STOC*, pp. 222–231, 2004.

[8] Arora, S. and Safra, S., "Probabilistic checking of proofs: A new characterization of NP," *J. ACM*, vol. 45, no. 1, pp. 70–122, 1998.

[9] Arriaga, R. and Vempala, S., "An algorithmic theory of learning: Robust concepts and random projection," in *Proc.* 40^{th} *IEEE FOCS*, pp. 616–623, 1999.

[10] Benjamin, I., Kalai, G., and Schramm, O., "Noise sensitivity of boolean functions and applications to percolation," *Inst. Hautes Études Sci. Publ. Math.*, no. 90, pp. 5–43, 1999.

[11] Blum, A., Frieze, A., Kannan, R., and Vempala, S., "A polynomial-time algorithm for learning noisy linear threshold functions," in *Proc.* 37^{th} *IEEE FOCS*, pp. 330–338, 1996.

[12] Blum, A., Kalai, A., and Wasserman, H., "Noise-tolerant learning, the parity problem, and the statistical query model," *J. of the ACM*, vol. 50(4), pp. 506–519, 2003.

[13] Blum, A. and Kannan, R., "Learning an intersection of a constant number of halfspaces over a uniform distribution," *J. Comput. Syst. Sci.*, vol. 54, no. 2, pp. 371–380, 1997.

[14] Blum, A. and Rivest, R., "Training a 3-node neural network is NP-complete," in *Proc. Machine Learning: From Theory to Applications*, pp. 9–28, 1993.

[15] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M., "Learnability and the Vapnik Chervonenkis dimension," *J. ACM*, vol. 36, no. 4, pp. 929–965, 1989.

[16] Bourgain, J., "On the distribution of the Fourier spectrum of boolean functions.," *Israel Journal of Mathematics*, vol. 131, pp. 269–276, 2002.

[17] Charikar, M., Makarychev, K., and Makarychev, Y., "Integrality gaps for Sherali-Adams relaxations," in *Proc.* 41^{st} *ACM STOC*, 2009.

[18] Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., and Sivakumar, D., "On the hardness of approximating multicut and sparsest-cut," *Computational Complexity*, vol. 15, no. 2, pp. 94–114, 2006.

[19] Cheeger, J. and Kleiner, B., "Differentiating maps into l^1 and the geometry of bv functions," Preprint, 2006.

[20] Cheeger, J. and Kleiner, B., "On the differentiation of Lipschitz maps from metric measure spaces to Banach spaces," Preprint, 2006.

[21] Cheeger, J., Kleiner, B., and Naor, A., "Quantitative bounds on the rate of central collapse of lipschitz maps on the heisenberg group," In preparation, 2006.

[22] Coudert, O. and Sasao, T., *Two level logic minimization.* Kluwer Academic Publishers, 2001.

[23] Czort, S., "The complexity of minimizing disjunctive normal form formulas. Master's Thesis, University of Aarhus," 1999.

[24] de la Vega, W. F. and Kenyon-Mathieu, C., "Linear programming relaxations of maxcut," in *ACM SODA*, pp. 53–61, 2007.

[25] Devanur, N., Khot, S., Saket, R., and Vishnoi, N., "Integrality gaps for sparsest cut and minimum linear arrangement problems," in *Proc.* 38^{th} *ACM STOC*, pp. 537–546, 2006.

[26] Engebretsen, L., "Lower bounds for non-boolean constraint satisfaction," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 7, no. 42, 2000.

[27] Feige, U., "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[28] Feige, U. and Schechtman, G., "On the optimality of the random hyperplane rounding technique for max cut," *Random Struct. Algorithms*, vol. 20, no. 3, pp. 403–440, 2002.

[29] Feldman, V., "Hardness of approximate two-level logic minimization and PAC learning with membership queries," in *Proc.* 38^{th} *ACM STOC*, pp. 363–372, 2006.

[30] Feldman, V., "Optimal hardness results for maximizing agreements with monomials," in *Proc. IEEE CCC*, pp. 226–236, 2006.

[31] Feldman, V., Gopalan, P., Khot, S., and Ponnuswami, A., "New results for learning noisy parities and halfspaces," in *Proc.* 47^{th} *IEEE FOCS*, pp. 563–574, 2006.

[32] Feldman, V., Guruswami, V., Raghavendra, P., and Wu, Y., "Agnostic learning of monomials by halfspaces is hard," Manuscript, 2009.

[33] Goemans, M. X. and Williamson, D. P., "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[34] Goldreich, O., Rubinfeld, R., and Sudan, M., "Learning polynomials with queries: The highly noisy case.," *SIAM J. Discrete Math.*, vol. 13, no. 4, pp. 535–570, 2000.

[35] Gopalan, P., Khot, S., and Saket, R., "Hardness of reconstructing multivariate polynomials over finite fields," in *Proc.* 48^{th} *IEEE FOCS*, pp. 349–359, 2007.

[36] Gupta, A., Krauthgamer, R., and Lee, J. R., "Bounded geometries, fractals, and low-distortion embeddings," in *Proc.* 44^{th} *IEEE FOCS*, 2003.

[37] Guruswami, V. and Raghavendra, P., "Hardness of learning halfspaces with noise," in *Proc.* 47^{th} *IEEE FOCS*, pp. 543–552, 2006.

[38] Håstad, J., "Some optimal inapproximability results," *J. ACM*, vol. 48, no. 4, pp. 798–859, 2001.

[39] Haussler, D., "Decision theoretic generalizations of the PAC model for neural net and other learning applications," *Inf. Comput.*, vol. 100, no. 1, pp. 78–150, 1992.

[40] Holmerin, J. and Khot, S., "A new PCP outer verifier with applications to homogeneous linear equations and max-bisection," in *Proc.* 36^{th} *ACM STOC*, pp. 11–20, 2004.

[41] Jackson, J., "An efficient membership-query algorithm for learning DNF with respect to the uniform distribution," *J. Comput. Syst. Sci.*, vol. 55, no. 3, pp. 414–440, 1997.

[42] Kahn, J., Kalai, G., and Linial, N., "The influence of variables on boolean functions," in *Proc.* 29^{th} *IEEE FOCS*, no. 29, pp. 68–80, 1988.

[43] Kalai, A., Klivans, A., Mansour, Y., and Servedio, R., "Agnostically learning halfspaces," in *Proc.* 46^{th} *IEEE FOCS*, pp. 11–20, 2005.

[44] Kearns, M. J., Schapire, R. E., and Sellie, L., "Toward efficient agnostic learning," *Machine Learning*, vol. 17, no. 2-3, pp. 115–141, 1994.

[45] Khot, S., "Improved inaproximability results for maxclique, chromatic number and approximate graph coloring," in *Proc.* 42^{nd} *IEEE FOCS*, pp. 600–609, 2001.

[46] Khot, S., "Hardness results for coloring 3-colorable 3-uniform hypergraphs," in *Proc.* 43^{rd} *IEEE FOCS*, pp. 23–32, 2002.

[47] Khot, S., "On the power of unique 2-prover 1-round games," in *Proc.* 34^{th} *ACM STOC*, pp. 767–775, 2002.

[48] Khot, S., Kindler, G., Mossel, E., and O'Donnell, R., "Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?," *SIAM J. Comput.*, vol. 37, no. 1, pp. 319–357, 2007.

[49] Khot, S. and Ponnuswami, A., "Better inapproximability results for maxclique, chromatic number and min-3Lin-deletion," in *ICALP*, pp. 226–237, 2006.

[50] Khot, S. and Saket, R., "A 3-query non-adaptive PCP with perfect completeness," in *Proc. IEEE CCC*, pp. 159–169, 2006.

[51] Khot, S. and Saket, R., "Hardness of minimizing and learning DNF expressions," in *Proc.* 49^{th} *IEEE FOCS*, pp. 231–240, 2008.

[52] Khot, S. and Saket, R., "On hardness of learning intersection of two halfspaces," in *Proc.* 40^{th} *ACM STOC*, pp. 345–354, 2008.

[53] Khot, S. and Saket, R., "SDP integrality gaps with local ℓ_1-emmbeddability," Submitted, 2009.

[54] Khot, S. and Vishnoi, N., "The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1," in *Proc.* 46^{th} *IEEE FOCS*, pp. 53–62, 2005.

[55] Klivans, A., O'Donnell, R., and Servedio, R., "Learning intersections and thresholds of halfspaces," in *Proc.* 43^{rd} *IEEE FOCS*, pp. 177–186, 2002.

[56] Klivans, A. and Servedio, R., "Learning DNF in time $2^{\tilde{O}(n^{1/3})}$," *J. Comput. Syst. Sci.*, vol. 68, no. 2, pp. 303–318, 2004.

[57] Krauthgamer, R. and Rabani, Y., "Improved lower bounds for embeddings into l_1," in *ACM SODA*, pp. 1010–1017, 2006.

[58] Lee, J. R. and Naor, A., "l_p metrics on the Heisenberg group and the Goemans-Linial conjecture," in *Proc.* 47^{th} *IEEE FOCS*, pp. 99–108, 2006.

[59] Levin, A. and Shashua, A., "Principal component analysis over continuous subspaces and intersection of half-spaces," in *Proc. ECCCV(3)*, pp. 635–650, 2002.

[60] Lund, C. and Yannakakis, M., "On the hardness of approximating minimization problems," *J. ACM*, vol. 41, no. 5, pp. 960–981, 1994.

[61] Masek, W., "Some NP-complete set covering problems. Unpublished," 1979.

[62] McCluskey, E., "Minimization of boolean functions," *Bell Sys. Tech. Jour.*, vol. 35, pp. 1417–1444, 1956.

[63] Murphy, D., "Nearest neighbor pattern classification perceptrons," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1595–1598, 1990.

[64] Nock, R., Jappy, P., and Sallantin, J., "Generalized graph colorability and compressibility of boolean formulae," in *Proc. ISAAC*, pp. 237–246, 1998.

[65] Pitt, L. and Valiant, L., "Computational limitations of learning from examples," *J. ACM*, vol. 35, no. 4, 1988.

[66] Quine, W., "The problem of simplifying truth functions," *American Mathematical Monthly*, vol. 59, pp. 521–531, 1952.

[67] Quine, W., "A way to simplify truth functions," *American Mathematical Monthly*, vol. 62, pp. 627–631, 1956.

[68] Raghavendra, P., "Optimal algorithms and inapproximability results for every CSP?," in *Proc.* 40^{th} *ACM STOC*, pp. 245–254, 2008.

[69] Raghavendra, P. and Steurer, D., "Integrality gaps for strong SDP relaxations of Unique Games," Manuscript, 2009.

[70] Rao, A., "Parallel repetition in projection games and a concentration bound," in *Proc.* 40^{th} *ACM STOC*, 2008.

[71] Raz, R., "A parallel repetition theorem," *SIAM J. Comput.*, vol. 27, no. 3, pp. 763–803, 1998.

[72] Rückert, U., Richter, L., and Kramer, S., "Quantitative association rules based on half-spaces: An optimization approach," in *Proc. ICDM*, pp. 507–510, 2004.

[73] Samorodnitsky, A. and Trevisan, L., "A PCP characterization of NP with optimal amortized query complexity," in *Proc.* 32^{nd} *ACM STOC*, pp. 191–199, 2000.

[74] Schapire, R., "The strength of weak learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990.

[75] Sherali, H. D. and Adams, W. P., "A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems," *Discrete Applied Mathematics*, vol. 52, no. 1, pp. 83–106, 1994.

[76] Shmoys, D. B., *Approximation Algorithms for NP-hard Problems*, ch. Approximation algorithms for Cut problems and their application to divide-and-conquer, pp. 192–235. PWS, 1997.

[77] Sudan, M. and Trevisan, L., "Probabilistically checkable proofs with low amortized query complexity," in *Proc.* 39^{th} *IEEE FOCS*, pp. 18–27, 1998.

[78] Ta-Shma, A., "A note on PCP vs MIP," *Information Processing Letters*, vol. 58, no. 3, pp. 475–484, 1997.

[79] Valiant, L., "A theory of the learnable," in *Proc.* 16^{th} *ACM STOC*, pp. 436–445, 1984.

[80] Vapnik, V. and Chervonenkis, A., "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[81] Vempala, S., "A random sampling based algorithm for learning the intersection of half-spaces," in *Proc.* 38^{th} *IEEE FOCS*, pp. 508–513, 1997.

[82] Viola, E., "The sum of d small-bias generators fools polynomials of degree d," in *Proc. IEEE CCC*, pp. 124–127, 2008.