# Approximation algorithms, Hardness, and PCPs

by

Devanathan Thiruvenkatachari

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
New York University
September 2019

Professor Subhash Khot

## Dedication

To Vee.

# Acknowledgements

I would like to thank Subhash for being an amazing mentor, guide and friend through this journey, for his patience in teaching me the concepts, for his words of encouragement in my low points, and for all the fun conversations about life, universe and everything. It has truly been an honor to work with him.

I would like to thank Professors Richard Cole, Prahladh Harsha, Swastik Kopparty and Oded Regev for taking the time to be on my committee.

I would also like to thank all the immensely talented people I have had the privilege to collaborate with during this period, including projects that didn't result in a publication: Amey Bhangale, Sasha Golovnev, Prahladh Harsha, Swastik Kopparty, Euiwoong Lee, Sushant Sachdeva, and Rishi Saket, and folks at Yahoo Research: David, Alina, Balazs, Chen-Yu, Chicheng. I gained a lot from every single one of them.

I am incredibly lucky to have had a lot of great teachers and mentors in high school and undergrad who taught me a lot about not just mathematics, but life in general. I wouldn't be here without them. I am deeply indebted to all of them, and in particular MNT, Jayalal, Shankar, Nari, and Sajin.

I would like to thank my friends Azam, Chaya, Harish, Huck, Noah, Sandro, Sasha, SidK, Siyao, and Shravas for all the great times at NYU, Aahlad, Aparna, Gunaa, Kiran, Kuchi, Manasa, and boss among others who have added color to my life, for the support and friendship in spite of the distance.

Finally, I would like to thank my family for supporting me and encouraging me to make bold choices all my life, my wife's family for becoming my family, and my wife Veena and our pups, Simba and Taz for the unconditional love.

# Abstract

This thesis is a collection of theoretical results on the topic of approximation algorithms and hardness of approximation. The results presented here use a combination of classical and modern techniques to achieve better approximation algorithms and hardness results for some pivotal NP-hard problems and their variants. We study CSPs from a multi-objective point of view, with the goal of simultaneous optimization of multiple instances over the same set of variables, with MAX-CUT as the central focus. We provide an approximation algorithm that is near optimal assuming the unique games conjecture. We also study PCPs and their role in hardness of approximation, and present a hardness result for 3-LIN in the sub-constant soundness regime. Lastly, dictatorship testing is a property testing problem with significant applications in proving hardness results, and we present an improvement on the soundness of the $k$-bit dictatorship test with perfect completeness.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Many fundamental optimization problems are known to be NP-hard, efficient polynomial time algorithms to solve these problems exactly don't exist unless P = NP. One approach to tackle these problems is to design algorithms that give suboptimal solutions, but with provable guarantees. Such algorithms, commonly referred to as approximation algorithms, run in time polynomial in input size, and return a valid solution which is bounded in terms of the optimal solution.

If the solution to a minimization problem given by an approximation algorithm is a multiplicative factor of $c$ away from the optimal solution in the worst case, the algorithm is called an $c$-approximation algorithm. For maximization problems, a $c$-approximation algorithm guarantees a solution of at least $1/c$ times the optimal value. In the realm of NP-hard problems, the best known approximation factor for various problems varies greatly, i.e. we know of problems for which the best known approximation algorithms give a guarantee of $(1 + \varepsilon)$ for all $\varepsilon > 0$, to problems for which we can only achieve approximation factor that degrades with the size of input. The question that immediately follows would be to know what's the best approximation we can hope to achieve. Therefore, a search for limits of approximability is an equally important question to fully understand a problem. Over the past two decades, using fairly involved reductions starting from the PCP theorem, the approximability of several important computational problems have been resolved.

## 1.1 SDP Hierarchies

Convex programming relaxations and rounding schemes are a powerful tool to design approximation algorithms. Most combinatorial optimization problems have a discrete solution space and allow for the problem to be modeled as an Integer program. Since it is NP-hard to solve integer programs exactly, the integral constraints are relaxed in order to get a program that can be solved in polynomial time. The resulting solution is then rounded to achieve a valid if not exact solution to the set of constraints, to give an approximation algorithm.

One such relaxation is the Semidefinite Programming relaxation. Here, the variables are relaxed to have vector values, and the goal is to optimize a linear objective function. A Semidefinite Program is an optimization problem of the form

$$\begin{array}{ll} \textbf{Minimize} & C \cdot X \\ \text{s. t} & A_i \cdot X = b_i, i = 1, ..., m \\ & X \succcurlyeq 0 \end{array}$$

The above relaxation can be viewed as a basic SDP relaxation. For a number of combinatorial optimization problems, the basic SDP relaxation yields optimal approximation algorithms. On the other hand, for some other problems, adding more constraints to the relaxation and gives better approximation guarantees. One systematic way to add contraints to SDPs was defined by Parrilo and Lasserre. They obtain a sequence of increasingly powerful relaxations, termed as the Lasserre Hierarchy of convex relaxations. One of the basic ingredients underlying mathematical programming relaxation hierarchies for combinatorial optimization problems is the idea of expanding the search space, from the discrete space of pure assignments to the continuous space of distributions over assignments. The $r$-th level hierarchy typically has $n^{O(r)}$ additional constraints and can be solved in time $n^{O(r)}$.

We give a more detailed explanation of the Hierarchy and how we use it to achieve a better approximation algorithm for simultaneous MAX-CUT in Chapter 3.

## 1.2 Probabilistically checkable proofs

Probabilistically checkable proofs [AS98, ALM+98] offer a robust classification of NP and is a central technique in hardness of approximation results. A PCP system for a language consists of a verifier that runs in polynomial time, and has oracle access to a ”proof”. Given an input, the verifier makes a sequence of queries to access various proof locations and decides on the membership of the input. The verifier is required to satisfy completeness and soundness, the former is the requirement that if the input belongs to the language, the verifier will always accept for some proof string. Soundness states that if the input does not belong to the language, irrespective of the proof, the verifier accepts the input with probability at most $\frac{1}{2}$. The complexity class $PCP(r, q)$ consists of languages captured by a PCP system that uses at most $r$ bits of randomness and queries the proof in at most $q$ locations. The PCP theorem states that NP is exactly the set of languages which have a PCP verifier that asks a constant number of queries using a logarithmic (in the size of input) number of coin tosses.

**Theorem 1.2.1 (PCP Theorem)** $NP = PCP(\log n, O(1))$

The PCP theorem has lead to many inapproximability results for various optimization problems.

### 1.2.1 Label Cover

**Definition 1.2.2 (LABEL COVER)** *An instance of* LABEL COVER *contains a regular bipartite multi- graph $G = (A, B, E)$ and two finite sets $\Sigma_A$ and $\Sigma_B$, where $|\Sigma_A| \geqslant |\Sigma_B|$. Every vertex in $A$ is supposed to get a label in $\Sigma_A$, and every vertex in $B$ is supposed to get a label in $\Sigma_B$. For each edge $e \in E$ there is a projection $\pi_e : \Sigma_A \to \Sigma_B$. Given a labeling to the vertices of the graph, i.e., functions $\phi_A : A \to \Sigma_A$ and $\phi_B : B \to \Sigma_B$, an edge $e = (a, b) \in E$ is said to be "satisfied" if $\pi_e(\phi_A(a)) = \phi_B(b)$. For $1 \geqslant c > s > 1$,* GAP LABEL COVER$(c, s)$ *is the problem if distinguishing whether the given instance of* LABEL COVER *is at least $c$-satisfiable or at most $s$-satisfiable.*

The PCP Theorem is equivalent to the following inapproximability of LABEL COVER.

**Theorem 1.2.3** *There exists a constant $c < 1$ such that given an instance of* LABEL COVER*, it is NP-hard to distinguish between two cases*

- *There exists an assignment satisfying all the edges.*

- *No assignment satisfies more than $c$ fraction of constraints.*

## 1.3 Overview of our Results

In this section, we present a brief overview of results that will be a part of the thesis.[1]

### 1.3.1 Simultaneous Approximation of Maxcut

Multiobjective optimization is an area of optimizing over more than one objective function where all the objective functions share the same solution space. In a joint work with Bhangale, Khot, Kopparty, and Sachdeva [BKK+16], we consider the well known optimization problem, MAX-CUT, from a multiobjective point of view, which we call *Simultaneous* MAX-CUT. The *Simultaneous* MAX-CUT problem is defined as follows. We are given multiple graphs $\{G_1, ..., G_k\}$ on the same vertex-set $V$ and their edge weights are given by functions $E_1, ..., E_k$, mapping each pair of vertices to the weight. We call each graph an instance. The goal is to partition $V$ into 2 sets such that across all the graphs, the minimum cut-weight is maximized. Our notion of an approximate solution to this problem is a very natural notion, given constants $c_1, c_2, ..., c_k$ such that there is an ”optimal” partition that has cut weight of $c_i$ in graph $G_i$, an $\alpha$ approximate algorithm would achieve a partition that would cut at least $\alpha \cdot c_i$ in graph $G_i$.

Since Simultaneous MAX-CUT is a generalization of MAX-CUT, we cannot

[1] The original papers, jointly written with co-authors, described our work in the best way I know of. Therefore, parts of this section, while rephrased, are similar to the constituent papers.

hope to achieve a better approximation algorithm than the best known algorithm for MAX-CUT which achieves a 0.878 approximation factor[GW95]. We also know that for MAX-CUT, this approximation factor is tight assuming the Unique Games Conjecture. We improved the approximation ratio of simultaneous Max-CUT problem from 1/2 [BKS15] to very close to [GW95], in fact our approximation factor matches theirs up to 3 decimal places, although it was achieved by computer assisted techniques. This improved algorithm uses the Lasserre Hierarchy.

### 1.3.2 Improved Hardness for 3LIN via Linear Label Cover

An instance of 3-LIN constitutes a system of linear equations such that there are at most 3 variables in each equation. We prove that for every constant $c$ and $\varepsilon = (\log n)^{-c}$, there is no polynomial time algorithm that when given an instance of 3-LIN with $n$ variables where an $(1-\varepsilon)$-fraction of the clauses are satisfiable, finds an assignment that satisfies atleast $(\frac{1}{2}+\varepsilon)$-fraction of clauses unless $\mathbf{NP} \subseteq \mathbf{BPP}$. The previous best hardness using a *polynomial time* reduction achieves $\varepsilon = (\log \log n)^{-c}$, which is obtained by the LABEL COVER hardness of Moshkovitz and Raz [MR08] followed by the reduction from LABEL COVER to 3-LIN of Håstad [Hås01].

Our main idea is to prove a hardness result for LABEL COVER similar to Moshkovitz and Raz where each projection has a *linear* structure. This linear structure of LABEL COVER allows us to use Hadamard codes instead of long codes, making the reduction more efficient. For the hardness of LINEAR LABEL COVER, we follow the work of Dinur and Harsha [DH13] that simplified the construction of Moshkovitz and Raz [MR08], and observe that running their reduction from a hardness of the problem LIN(of unbounded arity) instead of the more standard problem of solving quadratic equations ensures the linearity of the resultant LABEL COVER.

### 1.3.3 k-bit dictatorship test with perfect completeness

Dictatorship tests are central in proving many hardness results for constraint satisfaction problems. It falls under the category of property testing for Boolean functions, where given query access to a boolean function, we have to decide if the function satisfies a property or if it is "far" from it. The objective is to minimize the number of queries required to achieve this. The quality of a test is determined by two important factors - completeness and soundness. Completeness is the probability with which the test accepts if the function satisfies the property, and soundness is the probability with which the test (erroneously) accepts a function that is "far" from the property. A test with perfect completeness is one that always accepts a function that satisfies the property.

A boolean function is called a dictator if it depends on exactly one variable, i.e

$$f(x_1, x_2, ..., x_n) = x_i$$

for some $i \in [n]$. In joint work with Bhangale and Khot, we give a randomized dictatorship test with perfect completeness which is restricted to make only $k$ queries to $f$, with an improved soundness. The soundness probability we achieve is at most $\frac{2k+1}{2^k} + O(\varepsilon)$. The previous work [TY15] required the queried bits to satisfy pairwise independence condition, we improve on it and design a test which lacks pairwise independence condition but still proves the required soundness guarantee.

## 1.4 Organization

In chapter 2 we introduce some preliminaries and notations that we use in the thesis. In Chapter 3 we describe the near optimal approximation algorithm simultaneous MAX-CUT. In Chapter 4, we prove an improved hardness result for 3-LIN in the subconstant soundness regime, and in Chapter 5 we present an improved k-bit dictatorship test with perfect completeness.

# Chapter 2

# Preliminaries

## 2.1 Information Theory

In this section, we define and state some facts about entropy and mutual information between random variables.

**Definition 2.1.1 (Entropy)** *Let $X$ be a random variable taking values in $[q]$ then, entropy of $X$ is defined as:*

$$H(X) := \sum_{i \in [q]} \Pr[X = i] \log \frac{1}{\Pr[X = i]}.$$

**Definition 2.1.2 (Conditional Entropy)** *Let $X$, $Y$ be jointly distributed random variables taking values in $[q]$ then, the conditional entropy of $X$ conditioned on $Y$ is defined as:*

$$H(X|Y) = E_{i \in [q]} H(X|Y = i).$$

The following observations can be made about entropy of a collection of random variables.

Entropy of a collection of random variables cannot exceed the sum of their entropies.

**Fact 2.1.3** $H(X_1, X_1, \ldots, X_n) \leqslant \sum_{i=1}^{n} H(X_i)$.

Entropy never decreases on adding more random variables to the collection.

**Fact 2.1.4** $H(X_1, X_2|Y) \geqslant H(X_1|Y)$.

Conditioning can only decrease the entropy.

**Fact 2.1.5** $H(X|Y) - H(X|Y, Z) \geqslant 0$.

**Definition 2.1.6 (Mutual Information)** *Let X, Y be jointly distributed random variable taking values in $[q]$ then, the mutual information between $X$ and $Y$ is defined as:*

$$I(X;Y) := \sum_{i,j \in [q]} \Pr[X = i, Y = j] \log \frac{\Pr[X = i, Y = j]}{\Pr[X = i] \Pr[Y = j]}.$$

**Theorem 2.1.7** *(Data Processing Inequality) If $X, Y, W, Z$ are random variables such that $X$ is fully-determined by $W$ and $Y$ is fully-determined by $Z$, then*

$$I(X, Y) \leqslant I(W, Z).$$

## 2.2 Analysis of Boolean Function over Probability Spaces

For a positive integer $k$, we will denote the set $\{1, 2, \ldots, k\}$ by $[k]$. For a distribution $\mu$, let $\mu^{\otimes n}$ denotes the $n$-wise product distribution.

For a function $f : \{0,1\}^n \to \mathbf{R}$, the *Fourier decomposition* of $f$ is given by

$$f(x) = \sum_{T \subseteq [n]} \widehat{f}(T)\chi_T(x) \text{ where } \chi_T(x) := \prod_{i \in T}(-1)^{x_i} \text{ and } \widehat{f}(T) := \mathop{\mathbf{E}}_{x \in \{0,1\}^n} f(x)\chi_T(x).$$

The *Efron-Stein decomposition* is a generalization of the Fourier decomposition to product distributions of arbitrary probability spaces.

**Definition 2.2.1** *Let $(\Omega, \mu)$ be a probability space and $(\Omega^n, \mu^{\otimes n})$ be the corresponding product space. For a function $f : \Omega^n \to \mathbf{R}$, the Efron-Stein decomposition of $f$ with respect to the product space is given by*

$$f(x_1, \cdots, x_n) = \sum_{\beta \subseteq [n]} f_\beta(x),$$

*where $f_\beta$ depends only on $x_i$ for $i \in \beta$ and for all $\beta' \not\supseteq \beta$ , $a \in \Omega^{\beta'}$,*

$$\mathop{\mathbf{E}}_{x \in \mu^{\otimes n}} [f_\beta(x) \mid x_{\beta'} = a] = 0$$

Let $\|f\|_p := \mathbf{E}_{x \in \mu^{\otimes n}}[|f(x)|^p]^{1/p}$ for $1 \leqslant p < \infty$ and $\|f\|_\infty := \max_{x \in \Omega^{\otimes n}} |f(x)|$ .

**Definition 2.2.2** *For a multilinear polynomial $f : \mathbf{R}^n \to \mathbf{R}$ and any $D \in [n]$ define*

$$f^{\leqslant D} := \sum_{T \subseteq [n], |T| \leqslant D} \hat{f}(T)\chi_T$$

*i.e. $f^{\leqslant D}$ is degree $D$ part of $f$. Also define $f^{>D} := f - f^{\leqslant D}$.*

**Definition 2.2.3** *For $i \in [n]$, the influence of the $i$th coordinate on $f$ is defined as* *follows.*

$$\mathsf{Inf}_i[f] := \mathop{\mathbf{E}}_{x_1,\cdots,x_{i-1},x_{i+1},\cdots,x_n} \mathsf{Var}_{x_i}[f(x_1,\cdots,x_n)] = \sum_{\beta:i\in\beta} \|f_\beta\|_2^2.$$

*For an integer $d$, the degree $d$ influence is defined as*

$$\mathsf{Inf}_i^{\leqslant d}[f] := \sum_{\beta:i\in\beta,|\beta|\leqslant d} \|f_\beta\|_2^2.$$

It is easy to see that for Boolean functions, the sum of all the degree $d$ influences is at most $d$. A dictator is a function which depends on one variable. Thus, the degree $1$ influence of any dictator function is $1$ for some $i \in [n]$. We call a function *far* from any dictator if for every $i \in [n]$, the degree $d$ influence is very small for some large $d$. This motivates the following definition.

**Definition 2.2.4 ($(d,\tau)$-quasirandom function)** *A multilinear function $f : \mathbf{R}^n \to \mathbf{R}$ is said to be $(d,\tau)$-quasirandom if for every $i \in [n]$ it holds that*

$$\sum_{i\in S\subseteq[n],|S|\leqslant d} \hat{f}(S)^2 \leqslant \tau$$

We recall the Bonami-Beckner operator on Boolean functions.

**Definition 2.2.5** *For $\gamma \in [0,1]$, the Bonami-Beckner operator $T_{1-\gamma}$ is a linear operator mapping functions $f : \{0,1\}^n \to \mathbf{R}$ to functions $T_{1-\gamma}f : \{0,1\}^n \to \mathbf{R}$ as $T_{1-\gamma}f(x) = \mathbf{E}_y[f(y)]$ where $y$ is sampled by setting $y_i = x_i$ with probability $1-\gamma$ and $y_i$ to be uniformly random bit with probability $\gamma$ for each $i \in [n]$ independently.*

We have the following relation between the fourier decomposition of $T_{1-\gamma}f$ and $f$.

**Fact 2.2.6** $T_{1-\gamma} f = \sum_{T \subseteq [n]} (1-\gamma)^{|T|} \hat{f}(T) \chi_T$.

# Chapter 3

# Simultaneous Max-Cut

## 3.1 Introduction

In this paper, we give near-optimal approximation algorithms for the simultaneous MAX-CUT problem. Here we are given a collection of weighted graphs $G_1, G_2, \ldots, G_k$ on the same vertex set $V$ of size $n$. Our goal is to find a partition of the vertex set $V$ into two parts, such that in *every* graph, the total weight of edges going between the two parts is large. The $k = 1$ case is the classical MAX-CUT problem, and the approximability of this problem has been extensively studied [FL92, GW95, Hås01, KKMO07, MOO05, OW08]. This paper studies the approximability of this problem for constant $k$.

We fix some convenient notation. Let the weighted graphs $G_1, \ldots, G_k$ be given by weight functions $\mathcal{E}_1, \ldots, \mathcal{E}_k$, which assign to each pair in $\binom{V}{2}$ a weight in $[0, 1]$. We assume that for each $i \in [k]$, the total weight of all edges under $\mathcal{E}_i$ equals $1$. Let $f : V \to \{0, 1\}$ be a function, which we view as a partition of the vertex set. We define $\mathsf{val}(f, \mathcal{E}_i)$ to be the total weight (under $\mathcal{E}_i$) of the edges cut by the partition $f$. Given this setup, we can formally state the notions of approximation that we consider.

- **$\alpha$-minimum approximation:** Let $c$ be the maximum, over all partitions $f^* : V \to \{0,1\}$, of the quantity $\min_{i \in [k]} \mathsf{val}(f^*, \mathcal{E}_i)$. The goal is to output an $f : V \to \{0,1\}$ such that $\min_{i \in [k]} \mathsf{val}(f, \mathcal{E}_i) \geqslant \alpha \cdot c$.
- **$\alpha$-Pareto approximation:** Let $c_1, c_2, \ldots, c_k$ be given such that there exists $f^* : V \to \{0,1\}$ with $\mathsf{val}(f^*, \mathcal{E}_i) \geqslant c_i$ for each $i \in [k]$. The goal is to output an $f : V \to \{0,1\}$ such that $\mathsf{val}(f, \mathcal{E}_i) \geqslant \alpha \cdot c_i$ for all $i \in [k]$.

For $k = 1$, there is a celebrated polynomial time $\alpha_{GW} = 0.8786\ldots$ factor (Pareto) approximation algorithm by Goemans and Williamson [GW95]. This approximation is in both the minimum and Pareto senses. Furthermore, it is Unique-Games hard to achieve a better approximation factor [KKMO07], and the entire polynomial time "approximation curve" is also known.

For larger (but constant) $k$, far less is understood. Clearly, the hardness results from the $k = 1$ case carry over, and thus it is UniqueGames hard to approximate this to a factor better than $\alpha_{GW}$. [ABG06] gave a polynomial time $0.439$-Pareto approximation algorithm for this problem for the case $k = 2$. Subsequently, [BKS15] gave a polynomial time $(1/2 - \varepsilon)$-Pareto approximation algorithm for this problem. For the case of unweighted graphs[1], [BKS15] showed that there is a polynomial time $(1/2 + \Omega(1/k^2))$-minimum approximation algorithm. Furthermore, [BKS15] gave a matching integrality gap of $(1/2 + O(1/k^2))$ for a natural SDP relaxation of the minimum approximation problem.

[1] We call an instance of simultaneous MAX-CUT *unweighted* if for any $i$, all the nonzero weight edges under $\mathcal{E}_i$ have the same weight.

Our main result is a polynomial time $0.8780$-factor Pareto approximation algorithm for simultaneous MAX-CUT for arbitrary constant $k$.

**Theorem 3.1.1** *For all constant $k$ and $c > 0$, given weighted graphs $(G_i(V, \mathcal{E}_i))_{i=1}^k$ with $|V| = n$ and where all non-zero edge weights are lower bounded by $exp(-n^c)$, there is a poly$(n)$ time algorithm which computes a $0.8780$-factor Pareto approximation (and hence min approximation) to the simultaneous* MAX-CUT *problem with $k$ instances.*

**Remark 3.1.2** *We assume that the non-zero edge weights are lower bounded by $exp(-|V|^c)$ for some constant $c > 0$. We are interested in an algorithm which runs in time polynomial in $|V|$ and hence it is natural to assume the edge weights are lower bounded by $exp(-|V|^c)$ as otherwise the bit complexity of the input will be super polynomial in $|V|$.*

**Remark 3.1.3** *Our approximation ratio matches the Goemans-Williamson constant $\alpha_{GW} = 0.8786\ldots$ up to three decimal places. It might be possible to improve the approximation ratio through small modifyications our rounding procedure. However, we believe that getting the exact $\alpha_{GW}$-approximation (if it exists) might require new techniques. See Remark 3.2.12 for more details.*

We give a brief overview of ideas involved in our algorithm next. The main ingredients of the algorithm are: a sum-of-squares hierarchy SDP relaxation, a generalization of the [RT12], [ABG12] approach to rounding such relaxations, and some ideas from [BKS15].

### 3.1.1 Overview of the algorithm

We begin by considering the unweighted case; later we will discuss how to remove this restriction. One crucial observation about the unweighted case is that if there are enough edges in every graph (as a function of $k$), then a random cut simultaneously cuts a constant fraction of edges from each graph with high probability. Thus, we can always assume that each target value is $c_i = \Omega_k(1)$, which is a constant for a constant $k$.

There is a natural SDP relaxation for the simultaneous MAX-CUT problem, generalizing the Goemans-Williamson SDP for the $k = 1$ case. If we solve this SDP and round the resulting vector solution using the Goemans-Williamson hyperplane rounding procedure, this gives us a distribution of partitions of the vertex set $V$, such that for *each* $i \in [k]$, the total weight of edges cut in instance $i$ is at least $\alpha_{GW}$ times the corresponding SDP cut value. However, unlike in the $k = 1$ case, this does not guarantee the existence of a single partition of $V$ which is achieves a large cut value for all the $k$ instances simultaneously! This distinction between distributions of solutions which are good in expectation for each instance and single solutions that are simultaneously good for all instances is at the heart of the difficulty in designing simultaneous approximation algorithms.

One of the basic ingredients underlying mathematical programming relaxation hierarchies for combinatorial optimization problems is the idea of expanding the search space, from the discrete space of pure assignments to the continuous space of distributions over assignments. For simultaneous approximation of MAX-CUT beyond a factor $1/2$, this idea alone is not enough. An example from [BKS15] shows that there are cases of simultaneous MAX-CUT on $k$-instances, for which there is a distribution of partitions of $V$ cutting $(1 - \frac{1}{k})$-

fraction of edges *in expectation* for each instance, but for which any single partition of $V$, there is an instance $i \in [k]$, such that at most $1/2$ of the edges in instance $i$ are cut by the partition. This is where the sum-of-squares SDP hierarchy comes in – even though it is also modeled on the idea of expanding the search space to distributions of assignments – it allows us to *condition* on partial assignments and impose a constraint that the SDP cut value is large in expectation for each instance and for every possible conditioning on a small number of variables. This is what allows us to overcome the aforementioned obstacle.

Having formulated the SDP relaxation, we now discuss the rounding procedure. The motivating observation is this: if the rounding procedure is such that for each instance the expected cut value is large, and further the cut value is concentrated around its expectation with high probability, then by a union bound, the rounding procedure will produce a cut that is simultaneously good for all instances. The rounding procedure we will use will be closely related to the Goemans-Williamson rounding (but different – it was found by computer search given various technical conditions required by the rest of the algorithm). Our algorithm now tries to improve the concentration of the cut-value produced by the rounding procedure, via a beautiful information-theoretic approach of Raghavendra-Tan [RT12]. If the cut-value for a certain instance turns out to be not concentrated under the rounding procedure, then it must be because of high correlation between many pairs of edges of that instance (more precisely, correlation between the events that the edge is cut). This in turn means that conditioning on the variables in a random edge should significantly decrease the amount of entropy of the rounded cut. Iterating this several times, and using the fact that the initial entropy is not too large, we conclude that conditioning on a small number of variables leads to good concentration for the rounding procedure. The key point is that the sum-of-squares SDP relaxation we use gives us access to a vector solution for the conditioned SDP, with the promise that the SDP cut-value (and hence the expected integral cut-value) is still large. By the concentration property and a union bound, we get a simultaneously good cut.This completes the description of the algorithm in the unweighted case.

To handle the general weighted case, we essentially need to overcome few technical obstacles. Following [BKS15], we add a preprocessing and postprocessing phase. The preprocessing phase identifies "wild" instances, i.e. those instances with an abnormally large number of high (weighted-)degree vertices (which would increase the variance of the cut value of that instance under random rounding). Then the SDP based algorithm described above is run only on the "tame" instances.

With conditioning on constantly many variables, we can only manage to bring the variance down to arbitrarily small constant. Hence, in order to use second moment method to get concentration, we would need a good lower bound on the expected value of a cut given by our rounding procedure. If the graphs are weighted then it is not necessarily true that the simultaneous cut value is large for all instances. One important property of the tame instances we used is that they have a good simultaneous MAX-CUT value. We crucially use this property while formulating the SDP for tame instances.

Finally in the postprocessing phase, we find suitable assignment to the high degree vertices of the wild instances to ensure that those instance have a large cut value (without spoiling the large cut value of the tame instances that the SDP guaranteed) – this uses a new and much simpler perturbation argument compared to [BKS15].

This concludes the high-level description of the algorithm.

### 3.1.2 Note about the rounding procedure

We mentioned earlier that our SDP solution after conditioning on a small number of variables is rounded by a rounding algorithm similar to the Goemans-Williamson rounding algorithm, but is different. We discuss this rounding procedure here, and compare it to the previous results that used similar rounding procedures.

For convenience, we switch the notation from $0/1$ to $+1, -1$, such that any function $f : V \to \{-1, +1\}$ defines a cut in a natural way. Define the bias of a $\{+1, -1\}$ random variable $x$ as $\mathbf{E}[x]$. The SDP solution induces a consistent local distribution on every set of variables of size at most some constant $r$, and we define the *SDP-bias* of a variable as the bias with respect to this local distribution. For a given rounding procedure, we define the *rounding-bias* of a variable as the bias with respect to the rounding procedure. Note that in the original hyperplane rounding of Goemans-Williamson, the rounding-bias of each vertex is 0.

In the rounding procedure for the MAX-BISECTION from [RT12], the rounding bias for each variable induced by the rounding procedure is the same as the SDP-bias. Their algorithm gave a 0.85 approximation for MAX-BISECTION, and using the same bias function for the rounding along with the analysis of our algorithm, we can get 0.85 approximation for simultaneous MAX-CUT as well (See Section 3.2.3.6 for more details). The approximation factor given by [RT12] was subsequently improved in [ABG12] to 0.8776, where they used new techniques to relax the restriction on the choice of the bias function. Nevertheless, the rounding procedure was still quite constrained by the need to maintain the balance of the cut, as required by the MAX-BISECTION problem.

In our setting, we do not need equal sized partition of the vertex set, we have more freedom in our rounding procedure with respect to the rounding-bias. It turns out that we only have to ensure that when the bias of a variable is high, the side of the cut it falls on is almost fixed (that this condition suffices heavily depends on features of our algorithm and its analysis). This helps us achieve an improved approximation factor of 0.8780. The rounding function we come up with was arrived at by computer search (along with some trial-and-error).

The approximation ratio for our rounding procedure is proved by a computer assisted prover, using techniques similar to those of [Sjo09] and [ABG12].

### 3.1.3 Other related work

The simultaneous MAX-CUT problem is a special case of the simultaneous approximation problem for general constraint satisfaction problems. This general problem was studied in [BKS15], where it was shown that there is a polynomial time constant factor Pareto approximation algorithm for every simultaneous CSP (with approximation factor independent of $k$). The algorithm there was based on understanding the structure of CSP instances whose value is highly concentrated under a random assignment to the variables, in addition to linear-programming. It was also observed that there are CSPs for which the best polynomial time approximation factor for the simultaneous version (with $k > 1$) is *different* from the best polynomial time approximation factor achievable in the standard $k = 1$ case (assuming $P \neq NP$). This makes the study of simultaneous approximation factors very interesting.

The simultaneous MAXSAT problem was studied in [GRW11], where a $1/2$-Pareto approximation algorithm was given. For bounded width MAXSAT, the approximation factor was improved to $(3/4 - \varepsilon)$ in [BKS15].

It remains an open and very interesting problem to determine for which CSPs the simultaneous approximation problem for $k > 1$ is harder than the classical $k = 1$ case.

## 3.2 Algorithm for simultaneous weighted MAX-CUT

In this section, we give our approximation algorithm for simultaneous weighted MAX-CUT and the analysis.

### 3.2.1 Notation

We use the same notation as in [BKS15], which we reproduce here. Let $\mathcal{E} = \binom{V}{2}$ be the set of all possible edges. Given an edge $e$ and a vertex $v$, we say $v \in e$ if $v$ appears in the edge $e$. For an edge $e$, let $e_1, e_2$ denote the endpoints of $e$ (arbitrary order). Let $f : V \to \{0, 1\}$ be an assignment. For an edge $e \in \mathcal{E}$, define $e(f)$ to be 1 if the edge $e$ is cut by the assignment $f$, and define $e(f) = 0$ otherwise. Note that an assignment cuts an edge if it assigns different values to the end points. Then, we have the following expression for the cut value of the assignment:

$$\mathsf{val}(f, \mathcal{E}) \overset{\text{def}}{=} \sum_{e \in \mathcal{E}} \mathcal{E}(e) \cdot e(f).$$

A partial assignment $h : S \to \{0, 1\}$ is an assignment to $S$ where $S \subseteq V$. We say an edge is *active* with respect to $S$ if at least one of the end vertices is not in

$S$. We denote by $\mathsf{Active}(S)$ the set of all edges which are active with respect to $S$. For two edges $e, e' \in \mathcal{E}$, we say $e \sim_S e'$ if they share a vertex that is contained in $V \backslash S$. Note that if $e \sim_S e'$, then $e, e'$ are both in $\mathsf{Active}(S)$, and also $e \sim_S e, \forall e \in \mathcal{E}$. Let $\mathsf{actdist}_S(\ell)$ denote the distribution over $\mathsf{Active}(S)$, obtained by renormalizing $\mathcal{E}_\ell$ to have total weight $1$ over $\mathsf{Active}(S)$.

Define the active degree given $S$ of a variable $v \in V \backslash S$ for instance $\ell$ by:

$$\mathsf{actdeg}_S(v, \ell) \overset{\text{def}}{=} \sum_{e \in \mathsf{Active}(S), e \ni v} \mathcal{E}_\ell(e).$$

We then define the active degree of the whole instance $\ell$ given $S$:

$$\mathsf{actdeg}_S(\ell) \overset{\text{def}}{=} \sum_{v \in V \backslash S} \mathsf{actdeg}_S(v, \ell).$$

Note that we count weight of an active edge in $\mathsf{actdeg}_S(\ell)$ at most twice. For a partial assignment $h : S \to \{0, 1\}$, we define

$$\mathsf{val}(h, \mathcal{E}_\ell) \overset{\text{def}}{=} \sum_{\substack{e \in \mathcal{E} \\ e \notin \mathsf{Active}(S)}} \mathcal{E}_\ell(e) \cdot e(h)$$

which is the total weight of non-active edges cut by the partial assignment $h$. Thus, for an assignment $g : V \backslash S \to \{0, 1\}$, to the remaining set of variables, we have the equality:

$$\mathsf{val}(h \cup g, \mathcal{E}_\ell) - \mathsf{val}(h, \mathcal{E}_\ell) = \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

### 3.2.2 Algorithm

In Figure 3.1 and 3.2, we give the algorithm for Simultaneous Max-Cut. The input to the algorithm consists of an integer $k \geqslant 1$, $\varepsilon \in (0, 1/5]$, $k$ instances of Max-Cut, specified by weight functions $\mathcal{E}_1, \ldots, \mathcal{E}_k$, and $k$ target objective values $c_1, \ldots, c_k$.

```
**Input**: $k$ instances of MAX-CUT, with weights defined by $\mathcal{E}_1, \ldots, \mathcal{E}_k$ on the set of variables $V$, target objective values $c_1, \ldots, c_k$, and $\varepsilon \in (0, 1/5]$.
**Output**: An assignment to $V$.
**Parameters:** $\delta_0 = \frac{1}{10k}$, $\varepsilon_0 = \frac{\varepsilon}{2}$, $t = \frac{2k}{\gamma} \cdot \log\left(\frac{21}{\gamma}\right)$, $\tau = \varepsilon$, $\gamma = \frac{\tau^2 \varepsilon_0^2 \delta_0}{4}$.
**Pre-processing:**
  1. Initialize $S \leftarrow \varnothing$.
  2. For each instance $\ell \in [k]$, initialize $\mathsf{count}_\ell \leftarrow 0$ and $\mathsf{flag}_\ell \leftarrow$ TRUE.
  3. Repeat the following until for every $\ell \in [k]$, either $\mathsf{flag}_\ell =$ FALSE or $\mathsf{count}_\ell = t$:
      (a) For each $\ell \in [k]$, compute $\mathsf{Uvar}_\ell = \sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e')$.
      (b) For each $\ell \in [k]$ compute $\mathsf{Lmean}_\ell \overset{\text{def}}{=} \tau \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e)$.
      (c) For each $\ell \in [k]$, if $\mathsf{Uvar}_\ell \geqslant \delta_0 \varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then set $\mathsf{flag}_\ell =$ TRUE, else set $\mathsf{flag}_\ell =$ FALSE.
      (d) Choose any $\ell \in [k]$, such that $\mathsf{count}_\ell < t$ AND $\mathsf{flag}_\ell =$ TRUE (if any):
          i. Find $v \in V$ such that $\mathsf{actdeg}_S(v, \ell) \geqslant \gamma \cdot \mathsf{actdeg}_S(\ell)$.
          ii. Set $S \leftarrow S \cup \{v\}$. We say that $v$ was brought into $S$ because of instance $\ell$.
          iii. Set $\mathsf{count}_\ell \leftarrow \mathsf{count}_\ell + 1$.
  4. After exiting the loop:
      • Let $\mathcal{L}$ denote the set of all $\ell \in [k]$ for which $\mathsf{flag}_\ell$ is set to FALSE (these will be called "low-variance" instances).
      • Let $\mathcal{H}$ denote the set of all $\ell \in [k]$ for which $\mathsf{count}_\ell = t$ (these will be called "high-variance" instances).
```

Figure 3.1: Part 1 of Algorithm ALG-SIM-MAXCUT for approximating weighted simultaneous MAX-CUT

**Main algorithm:**

1. For each possible partial fixing $h : S \to \{0, 1\}$ do the following

   (a) Solve the SDP given in Figure 3.4 (Refer Section 3.2.3.3).

   (b) Follow the procedure in Figure 3.5 to make the solution locally independent. (Refer Section 3.2.3.4)

   (c) Round the solution based on the rounding procedure described in Figure 3.6 to get a partial assignment $g : V \backslash S \to \{0, 1\}$. (Refer Section 3.2.3.5)

   (d) **Post-processing step:** For every assignment $h' : S \to \{0, 1\}$, compute $\min\limits_{\ell} \frac{\mathsf{val}(h' \cup g, \mathcal{E}_\ell)}{c_\ell}$ and return the assignment $h' \cup g$ that maximizes this.

Figure 3.2: Part 2 of Algorithm ALG-SIM-MAXCUT for approximating weighted simultaneous MAX-CUT

### 3.2.3 Analysis of the Algorithm

The algorithm broadly proceeds in 3 sections, the pre-processing step, the SDP step and the post processing step. The pre-processing step consists of identifying a small subset $S \subseteq V$ carefully. We then attempt all assignments to vertices in $S$ by brute force iteratively and use SDP with the partial assignment followed by a rounding to assign vertices in $V \backslash S$. The post-processing step involves perturbing the assignments to the vertices in $S$, the need for which is explained in detail in Section 3.2.3.7.

In what follows, we stick to the following notation. Let $S^\star$ denote the final set $S$ that we get at the end of Step 3. of ALG-SIM-MAXCUT. Let $f^\star : V \to \{0,1\}$ be the assignment that achieves $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$ for all $l \in [k]$ and $h^\star$ be the restriction of $f^\star$ to the set $S^\star$.

#### 3.2.3.1 Pre-processing: Low and High variance instances

**Definition 3.2.1 ($\tau$-smooth distribution)** *A distribution $D$ on $\{0,1\}$ is called $\tau$-smooth if*

$$\Pr_{x \sim D}[x = 1] \geqslant \tau, \quad \Pr_{x \sim D}[x = 0] \geqslant \tau.$$

Let $h : S \to \{0,1\}$ be an arbitrary partial assignment to the vertices in $S$. Let $g : V \backslash S \to \{0,1\}$ be the random assignment such that each of the marginals $g(v)$ is $\tau$-smooth. For an instance $\ell$, define the random variable

$$Y_\ell \overset{\text{def}}{=} \mathsf{val}(h \cup g, \mathcal{E}_\ell) - \mathsf{val}(h, \mathcal{E}_\ell) = \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

$Y_\ell$ measures the total active edge weight cut by the assignment in the instance $\ell$.

Consider the two quantities defined in Step 3. of the algorithm. They depend only on $S$ (and importantly, not on $h$), which will be useful in controlling the expectation and variance of $Y_\ell$. The first quantity is an upper bound on $\mathsf{Var}[Y_\ell]$:

$$\mathsf{Uvar}_\ell \overset{\text{def}}{=} \sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e').$$

The second quantity is a lower bound on $\mathbf{E}[Y_\ell]$:

$$\mathsf{Lmean}_\ell \overset{\text{def}}{=} \tau \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

**Lemma 3.2.2** *Let $S \subseteq V$ be a subset of vertices and $h : S \to \{0,1\}$ be an arbitrary partial assignment to $S$. Let $Y_\ell, \mathsf{Uvar}_\ell, \mathsf{Lmean}_\ell$ be as above.*

1. *If $\mathsf{Uvar}_\ell \leqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then $\Pr[Y_\ell < (1-\varepsilon_0)\,\mathbf{E}[Y_\ell]] < \delta_0$.*

2. *If $\mathsf{Uvar}_\ell \geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then there exists $v \in V \backslash S$ such that*

$$\mathsf{actdeg}_S(v,\ell) \geqslant \frac{1}{4}\tau^2\varepsilon_0^2\delta_0 \cdot \mathsf{actdeg}_S(\ell).$$

We defer the formal proof to Section 3.4.2. The first part is a simple application of the Chebyshev inequality. For the second part, we use the assumption that $\mathsf{Uvar}_\ell$ is large, to deduce that there exists an edge $e$ such that the total weight of edges adjacent to the vertex/vertices in $e$ that belong to $V \backslash S$, *i.e.*, $\sum_{e_2 \sim_S e} \mathcal{E}(e_2)$, is large. It then follows that at least one variable $v \in e$ must have large active degree given $S$.

The above lemma (Lemma 3.2.2) ensures that Step 3.(d)i in the algorithm always succeeds in finding a variable $v$. Next, we note that Step 3. always terminates. Indeed, whenever we find an instance $\ell \in [k]$ in Step 3.d such that $\mathsf{count}_\ell < t$ and $\mathsf{flag}_\ell = \textsc{True}$, we increment $\mathsf{count}_\ell$. This can happen only $tk$ times before the condition $\mathsf{count}_\ell < t$ fails for all $\ell \in [k]$. Thus the loop must terminate within $tk$ iterations.

To analyze the approximation guarantee of the algorithm, we classify instances according to how many vertices were brought into $S^\star$ because of them.

**Definition 3.2.3 (Low and High variance instances)** *At the completion of Step 3.d in Algorithm* ALG-SIM-MAXCUT*, if $\ell \in [k]$ satisfies $\mathsf{count}_\ell = t$, we call instance $\ell$ a* high variance *instance. Otherwise we call instance $\ell$ a* low variance *instance.*

The next two sections describes the SDPs that we formulate and solve for just the low variance instances. The claim that step 1d of the algorithm shown in Figure 3.2 handles the high variance instances is discussed and proved in Section 3.2.3.7.

#### 3.2.3.2 Warmup: Basic SDP formulation for simultaneous MAX-CUT.

Our algorithm involves formulating a *Lasserre Hierarchy* SDP relaxation of the residual MAX-CUT problem after giving a partial assignment $h : S^\star \to \{0, 1\}$. In this section, as a warmup to its analysis, we present and study the *basic* version of that SDP.

We write the SDP$^\star$ for simultaneous MAX-CUT problem, after the partial fixing given by pre-processing step, as in Figure 3.3. Let $\mathcal{L}$ denote the set of indices of the low variance instances. We have vectors $\boldsymbol{v_{T,\alpha}}$ for all $T$ and $\alpha$ where $T$ is a subset of $V$ of size at most 2, and $\alpha$ is an assignment to the vertices in $T$.

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e)(\|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{0},\boldsymbol{1})\}}\|_2^2 + \|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{1},\boldsymbol{0})\}}\|_2^2) \geqslant (1-3\varepsilon)c_\ell \qquad \forall \ell \in [k], \tag{3.2.1}$$

$$\langle \boldsymbol{v}_{\{\boldsymbol{i},\boldsymbol{0}\}}, \boldsymbol{v}_{\{\boldsymbol{i},\boldsymbol{1}\}} \rangle = 0 \qquad \forall i \in [n],$$

$$\|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{b_1},\boldsymbol{b_2})\}}\|^2 = \langle \boldsymbol{v}_{\{\boldsymbol{i},\boldsymbol{b_1}\}}, \boldsymbol{v}_{\{\boldsymbol{j},\boldsymbol{b_2}\}} \rangle \qquad \forall i, j \in [n] \text{ and } b_1, b_2 \in \{0,1\}$$

$$\|\boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\alpha}\}}\|^2 = \langle \boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\alpha}\}}, \boldsymbol{v}_\varnothing \rangle \qquad \forall T \subset V, |T| \leqslant 2, \alpha \in \{0,1\}^{|T|}$$

$$\boldsymbol{v}_{\{\boldsymbol{i},\boldsymbol{b}\}} = \boldsymbol{v}_\varnothing \qquad \forall i \in S^\star, b = h(i)$$

$$\|\boldsymbol{v}_\varnothing\|^2 = 1$$

$$\sum_{e=\{i,j\}\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(e)(\|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{0},\boldsymbol{1})\}}\|_2^2 + \|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{1},\boldsymbol{0})\}}\|_2^2) \geqslant {}^{\varepsilon}\!/_{3}.\mathsf{actdeg}_{S^\star}(\ell) \qquad \forall \ell \in \mathcal{L} \tag{3.2.2}$$

Figure 3.3: SDP$^\star(h : S^\star \to \{0,1\})$ for simultaneous MAX-CUT with partial fixing

If we consider the SDP$^\star$ without the constraint (3.2.2), it is easy to see that this is a relaxation. Given a partition $(U, \bar{U})$ of $V$ that achieves a simultaneous optimum, we can set vectors $\boldsymbol{v}_{\boldsymbol{T},\boldsymbol{\alpha}} = \boldsymbol{v}_\varnothing$ if the pair $(T, \alpha)$ is consistent with $1_U$ (i.e. $1_U$ assigns $\alpha$ to $T$) and $\boldsymbol{v}_{\boldsymbol{T},\boldsymbol{\alpha}} = 0$ otherwise. $\boldsymbol{v}_\varnothing$ can be viewed as a vector that denotes 1.

A part of our analysis require that for every low variance instance, the expected weighted fraction of active edges that we cut is at least a constant fraction of its active degree. An optimal SDP solution without constraint (3.2.2) may not guarantee this condition (for the rounding procedure we choose). Hence, we force the SDP solution to satisfy this property by adding constraint (3.2.2). We need to relax constraint (3.2.1) to make sure that there is a solution that satisfies all the constraints.

We now prove that $\mathsf{SDP}^\star$, in its present form, has feasible solutions.

**Lemma 3.2.4** $\mathsf{SDP}^\star(h^\star)$ *shown in Figure 3.3 has a feasible solution.*

**Proof:** To show that $\mathsf{SDP}^\star$ has a feasible solution, it suffices to show that there exists an integral solution that satisfies the constraints.

Fix an optimal assignment $f^\star : V \to \{0,1\}$ to the simultaneous instance. $f^\star$ satisfies $\forall \ell \in [k]$, $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$. Consider the following random assignment: For all $v \in V \backslash S^\star$

$$r(v) = \begin{cases} f^\star(v) & \text{with probability } (1-\varepsilon) \\ \overline{f^\star(v)} & \text{otherwise} \end{cases}$$

where $\overline{f^\star(v)}$ is $f^\star(v)$ flipped. For $v \in S^\star$, set $r(v) = f^\star(v)$. Now, for any $\ell \in \mathcal{L}$, let $Y_\ell$ denote the random variable

$$Y_\ell = \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot e(r).$$

We have $\mathbf{E}[e(r)] \geqslant \varepsilon$, hence $\mathbf{E}[Y_\ell] \geqslant \varepsilon/2 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Also,

$$\begin{aligned}
\mathop{\mathbf{E}}_{r}[\mathsf{val}(r, \mathcal{E}_\ell)] &\geqslant \sum_{e \notin \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] + \sum_{\substack{e \in \mathsf{Active}(S^\star), \\ e(f^\star)=1}} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] \\
&= \sum_{e \notin \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot e(f^\star) + \sum_{\substack{e \in \mathsf{Active}(S^\star), \\ e(f^\star)=1}} \mathcal{E}_\ell(e) \cdot \min((1-\varepsilon)^2 + \varepsilon^2, 1-\varepsilon) \\
&\geqslant (1-2\varepsilon) \sum_{e : e(f^\star)=1} \mathcal{E}_\ell(e) \\
&= (1-2\varepsilon)\mathsf{val}(f^\star, \mathcal{E}_\ell) \\
&\geqslant (1-2\varepsilon)c_\ell.
\end{aligned}$$

Thus, we have,

1. $\mathbf{E}[Y_\ell] \geqslant {}^{\varepsilon}\!/\!_2 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.

2. $\underset{r}{\mathbf{E}}[\mathsf{val}(r, \mathcal{E}_\ell)] \geqslant (1 - 2\varepsilon)c_\ell$.

Recall that the SDP$^\star$ involves only the low variance instances. Also, the assignment $r$ is $\varepsilon$-smooth on the set $V \backslash S^\star$. Therefore, we have concentration guarantees as given by point 1 of Lemma 3.2.2.

$$\Pr[Y_\ell \leqslant (1 - \varepsilon_0)\, \mathbf{E}[Y_\ell]] \leqslant \delta_0$$

$$\Pr[\mathsf{val}(r, \mathcal{E}_\ell) \leqslant (1 - \varepsilon_0)\, \mathbf{E}[\mathsf{val}(r, \mathcal{E}_\ell)]] \leqslant \delta_0.$$

Hence, with probability at least $1 - 2\delta_0$, we have $Y_\ell \geqslant (1 - {}^{\varepsilon}\!/\!_2) \cdot {}^{\varepsilon}\!/\!_2 \cdot \mathsf{actdeg}_{S^\star}(\ell) \geqslant {}^{\varepsilon}\!/\!_3 \cdot \mathsf{actdeg}_{S^\star}(\ell)$ and $\mathsf{val}(r, \mathcal{E}_\ell) \geqslant (1 - {}^{\varepsilon}\!/\!_2)(1 - 2\varepsilon)c_\ell \geqslant (1 - 3\varepsilon)c_\ell$.

Now we do union bound over all low variance instances, we get with a probability at least $1 - 2 \cdot \delta_0 \cdot k = {}^4\!/\!_5$, all the SDP constraints are satisfied by integral solution $r$. Thus, there exists an *integral* solution which satisfies all SDP$^\star(h^\star)$ constraints and hence is feasible. ■

#### 3.2.3.3 Lasserre Hierarchy SDP formulation.

We now describe the $r^{th}$-level Lasserre SDP for the SDP in Figure 3.3.

The SDP formulation has vectors $\boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\alpha}\}}$ for all $T \subseteq V$ such that $|T| \leqslant r$ and $\alpha \in \{0,1\}^{|T|}$. In terms of local distribution, the SDP solution consists of *consistent local distribution* on every set $T$ of size at most $r$ (denoted by $\mu_T$). The random variable corresponding to set $T$ is denoted by $X_T$ distributed over $\{0,1\}^{|T|}$. The vector solution and the local distribution are related as follows: Suppose $T$ and

$$\begin{aligned}\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \big(\ \mathcal{E}_\ell(e)(\|\boldsymbol{v}_{\{\boldsymbol{S}\cup\{\boldsymbol{i},\boldsymbol{j}\},\boldsymbol{\alpha}\circ(\mathbf{0},\mathbf{1})\}}\|_2^2 \qquad & \forall S\subseteq V, |S|\leqslant r-2,\\ +\|\boldsymbol{v}_{\{\boldsymbol{S}\cup\{\boldsymbol{i},\boldsymbol{j}\},\boldsymbol{\alpha}\circ(\mathbf{1},\mathbf{0})\}}\|_2^2)\ \big) \qquad & \forall\alpha\in\{0,1\}^{|S|},\\ \geqslant(1-3\varepsilon)c_\ell\|\boldsymbol{v}_{\{\boldsymbol{S},\boldsymbol{\alpha}\}}\|^2 \qquad & \forall\ell\in[k]\end{aligned}\tag{3.2.3}$$

$$\begin{aligned}\sum_{e=\{i,j\}\in\mathsf{Active}(S^\star)} \big(\ \mathcal{E}_\ell(e)(\|\boldsymbol{v}_{\{\boldsymbol{S}\cup\{\boldsymbol{i},\boldsymbol{j}\},\boldsymbol{\alpha}\circ(\mathbf{0},\mathbf{1})\}}\|_2^2 \qquad & \forall S\subseteq V, |S|\leqslant r-2,\\ +\|\boldsymbol{v}_{\{\boldsymbol{S}\cup\{\boldsymbol{i},\boldsymbol{j}\},\boldsymbol{\alpha}\circ(\mathbf{1},\mathbf{0})\}}\|_2^2)\ \big) \qquad & \forall\alpha\in\{0,1\}^{|S|},\\ \geqslant {}^{\varepsilon}\!/_{3}.\mathsf{actdeg}_{S^\star}(\ell)\|\boldsymbol{v}_{\{\boldsymbol{S},\boldsymbol{\alpha}\}}\|^2 \qquad & \forall\ell\in\mathcal{L}\end{aligned}\tag{3.2.4}$$

$$\begin{aligned}\langle\boldsymbol{v}_{\{\boldsymbol{S},\boldsymbol{\alpha}\}},\boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\beta}\}}\rangle=\|\boldsymbol{v}_{\{\boldsymbol{S}\cup\boldsymbol{T},\boldsymbol{\alpha}\circ\boldsymbol{\beta}\}}\|_2^2 \qquad & \forall S,T\subseteq V, |S\cup T|\leqslant r,\\ & \alpha\in\{0,1\}^{|S|},\beta\in\{0,1\}^{|T|},\end{aligned}\tag{3.2.5}$$

$$\begin{aligned}\langle\boldsymbol{v}_{\boldsymbol{S},\boldsymbol{\alpha}},\boldsymbol{v}_{\boldsymbol{T},\boldsymbol{\beta}}\rangle=0 \qquad & \forall S,T\subseteq V, |S\cup T|\leqslant r,\\ & \alpha\in\{0,1\}^{|S|},\beta\in\{0,1\}^{|T|},\\ & \text{s.t. } \alpha_{|S\cap T}\neq\beta_{|S\cap T}\end{aligned}\tag{3.2.6}$$

$$\begin{aligned}\|\boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\alpha}\}}\|^2=\langle\boldsymbol{v}_{\{\boldsymbol{T},\boldsymbol{\alpha}\}},\boldsymbol{v}_\varnothing\rangle \qquad & \forall T\subseteq V, |T|\leqslant r,\alpha\in\{0,1\}^{|T|}\\ \langle\boldsymbol{v}_{\{\boldsymbol{S},\boldsymbol{\alpha}\}},\boldsymbol{v}_{\{\boldsymbol{i},\boldsymbol{b}\}}\rangle=\langle\boldsymbol{v}_{\{\boldsymbol{S},\boldsymbol{\alpha}\}},\boldsymbol{v}_\varnothing\rangle \qquad & \forall S\subseteq V, |S|\leqslant r-1,\alpha\in\{0,1\}^{|S|}\\ & \forall i\in S^\star, b=h(i)\end{aligned}\tag{3.2.7}$$

$$\|\boldsymbol{v}_\varnothing\|^2=1$$

Figure 3.4: $r$-round Lasserre lift of SDP$^\star(h:S^\star\to\{0,1\})$ for simultaneous Max-Cut with partial fixing

$U$ are subsets of $V$ such that $|T\cup U|\leqslant r$ and the assignments $\alpha\in\{0,1\}^{|T|}$ and $\beta\in\{0,1\}^{|U|}$ are consistent on $T\cap U$ then

$$\langle\boldsymbol{v}_{\boldsymbol{T},\boldsymbol{\alpha}},\boldsymbol{v}_{\boldsymbol{U},\boldsymbol{\beta}}\rangle=\Pr_{\mu_{T\cup U}}(X_T=\alpha,X_U=\beta).$$

To ensure the consistency among local distributions, we have to add the constraints 3.2.5 and 3.2.6 to the SDP in Figure 3.4. Here if $\alpha \in \{0,1\}^{|S|}$ is an assignment to the vertices in $S$, and if $S' \subset S$, $\alpha_{|S'} \in \{0,1\}^{|S'|}$ denotes the assignment $\alpha$ restricted to the vertices in $S'$. Also, if $\alpha$ and $\beta$ are assignments to sets $S$ and $T$ agreeing on $S \cap T$, then we denote $\alpha \circ \beta$ an assignment to $S \cup T$. We also add the set of constraints (Equation 3.2.7 in Figure 3.4) to capture the partial assignment $h : S^\star \to \{0,1\}$ given by pre-processing.

With these definitions and constraints, the objective is to ensure that for all $\ell \in [k]$,

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e) \Pr\big[X_{\{i,j\}} = (0,1) \vee X_{\{i,j\}} = (1,0)\big]$$
$$\geqslant (1-3\varepsilon)c_\ell$$

A simple way to capture this would be to write the objective of the SDP solution similar to the basic SDP formulation, as follows.

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e)\big(\|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\mathbf{1},\mathbf{0})\}}\|_2^2\big)$$
$$\geqslant (1-3\varepsilon)c_\ell$$

**Lemma 3.2.5** *$r$-round Lasserre SDP shown in Figure 3.4 has a feasible solution.*

**Proof:** Note that the feasible solution given for the basic SDP in Lemma 3.2.4 is integral. Therefore, we can directly conclude that the Lasserre lift of the SDP is feasible, as the same solution can be extended to the Lasserre SDP.

Assign $\boldsymbol{v}_{S,\alpha}$ to $\boldsymbol{v}_\varnothing$ if in the integral solution, the vertices in the set $S$ were assigned to $\alpha$ in that order, otherwise assign $\boldsymbol{v}_{S,\alpha}$ to $0$. ■

In order to make the solution locally independent, we will need to condition based on the local distribution (Refer Section 3.2.3.4). Therefore, we need to rewrite the objective so that it is satisfied (w.r.t the conditioned local distribution) even after conditioning on at most $r$ variables, as shown in Equation 3.2.3 in the SDP formulation.

Also, similar to the previous case, we need to ensure that the solution post-conditioning still cuts at least a constant fraction of the active edges, which is ensured by adding the set of constraints specified in Equation 3.2.4 in the SDP.

We observe that solving the SDP using ellipsoid method can result in a small additive error, and if $\mathsf{actdeg}_{S^\star}(\ell)$ is small compared to this additive error, the error would be significant. This will not cause any issues and we elaborate on this more. We can solve the SDP using ellipsoid method with an error of $\varepsilon$ in time polynomial in $n$ and $\log(1/\varepsilon)$. Therefore, we can take $\varepsilon$ to be $\exp(-\text{poly}(n))$ and still solve the SDP in time polynomial in $n$. We assumed that the non-zero edge weights are at least $\exp(-n^c)$ for some constant $c > 0$. Therefore, if the active degree is non-zero, it is at least $\exp(-n^c)$. If we take $\varepsilon = \exp(-n^{c'})$ for $c' >> c$, we can solve the SDP in time polynomial in $n$ and get a vector solution which satisfies all the constraints upto additive error $\varepsilon$ which is upto multiplicative factor of $(1 + o(1))$. This will not have a major effect on our analysis and hence we assume from here onward that the vector solution that we get satisfies the all the constraints exactly.

#### 3.2.3.4 Obtaining independent local solution

The notion of an independent solution (which is formalized below in Definition 3.2.6) that we need is different from [RT12]. Following procedure in Figure 3.5 is used to achieve the kind of independence we need.

**Definition 3.2.6** *A Lasserre solution is $\delta$-independent if it satisfies the following condition.*

$$\forall \ell \in \mathcal{L}, \underset{a,b\sim \mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}} \left[ \sum_{i,j\in\{1,2\}} I(X_{a_i}; X_{b_j}) \right] \leqslant \delta.$$

**Input**: $r+2$ round Lasserre solution of a given simultaneous MAX-CUT instance, $\delta \geqslant \frac{32k}{r}$
**Output**: $\frac{\delta}{2}$-*independent* 2-round Lasserre solution.

1. For all $\ell_1, \ldots, \ell_{r/2} \in \mathcal{L}$, and for all edges $e^i \in \mathsf{actdist}_{S^\star}(\ell_i)$ for all $i \in [r/2]$.
    - Let $S = \cup_{i\in[r/2]}\{e^i_1, e^i_2\}$ be the endpoints of all the edges from $(1)$.
    - For every $\alpha \in \{0,1\}^{|S|}$ such that $\Pr[X_S = \alpha] > 0$ in the local disctibution:
        - Condition the SDP solution on the event $X_S = \alpha$.
        - Output if conditioned solution if it is $\frac{\delta}{2}$-independent.

Figure 3.5: Making locally independent solution

**Lemma 3.2.7** *For all $\delta > 0$, there exists $t \leqslant 2k/\delta$ and edges $e^1, e^2, \ldots, e^t \in \mathcal{E}$ such that*

$$\forall \ell \in \mathcal{L}, \tag{3.2.8}$$

$$\underset{a,b\sim \mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}} [I(X_{a_1}, X_{a_2}; X_{b_1}, X_{b_2} | X_{e^1_1}, X_{e^1_2}, \ldots, X_{e^t_1}, X_{e^t_2})] \leqslant \delta$$

**Proof:**

Consider the following potential function,

$$\phi = \sum_{\ell\in\mathcal{L}} \underset{a\in \mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}} H(X_{a_1}, X_{a_2}).$$

As entropy of a bit is at most $1$, clearly $\phi \leqslant 2k$. We have the following identity for each $\ell \in \mathcal{L}$ which follows from conditional entropy and linearity of expectation

$$\begin{aligned}&\mathop{\mathbf{E}}_{a,b\in \mathsf{actdist}_{S^\star}(\ell)}[H(X_{a_1},X_{a_2}|X_{b_1},X_{b_2})]\\ &=\mathop{\mathbf{E}}_{a\in \mathsf{actdist}_{S^\star}(\ell)}[H(X_{a_1},X_{a_2})]-\\ &\qquad\mathop{\mathbf{E}}_{a,b\in \mathsf{actdist}_{S^\star}(\ell)} I(X_{a_1},X_{a_2};X_{b_1},X_{b_2})\end{aligned}$$

This identity suggests that if for some $\ell \in \mathcal{L}$, $\mathbf{E}_{a,b\in \mathsf{actdist}_{S^\star}(\ell)} I(X_{a_1},X_{a_2};X_{b_1},X_{b_2}) > \delta$ then there exists a conditioning which reduces the potential function by at least $\delta$. Thus, either the current conditioned solution satisfies (3.2.8) in which case we are done or there exists an edge $b$ such that if we condition the SDP solution based on the value of its endpoints $(b_1, b_2)$ according to the local distribution then the potential function decreases by at least $\delta$. So, if we fail to achieve (3.2.8) then $\phi$ decreases by at least $\delta$. As entropy is always non-negative and conditioning never increases entropy (Fact 2.1.5), this process cannot go beyond $2k/\delta$ conditioning. Thus, before at most $2k/\delta$ conditioning, we are guaranteed to achieve (3.2.8).

■

The following fact follows from the data processing inequality given earlier (Theorem 2.1.7).

**Fact 3.2.8** *If $X_1, X_2, Y_1$ and $Y_2$ are random variables then for $i, j \in \{1, 2\}$, we have*

$$I(X_i;Y_j) \leqslant I(X_1,X_2;Y_1,Y_2).$$

The following corollary follows from Lemma 3.2.7 and Fact 3.2.8.

**Corollary 3.2.9** *For all $\delta > 0$, there exists $t \leqslant \frac{2k}{\delta}$, and edges $e^1, e^2, \ldots, e^t \in \mathcal{E}$, such that*

$$\forall \ell \in \mathcal{L},$$

$$\mathop{\mathbf{E}}_{a,b \sim \mathsf{actdist}_{S^\star}(\ell)} \left[ \sum_{i,j \in \{1,2\}} I(X_{a_i}; X_{b_j} | X_{e_1^1}, X_{e_2^1}, \ldots, X_{e_1^{t-1}}, X_{e_2^{t-1}}) \right] \leqslant 4\delta$$

**Lemma 3.2.10** *There exists a fixing of at most $\frac{32k}{\delta}$* variables *such that the conditioned solution is $\delta/2$ independent as well as satisfies all constraints from* $\mathsf{SDP}^\star(h^\star)$*. In particular, the algorithm in Figure 3.5 returns such a $\delta/2$ independent solution. Also, the running time is bounded by $n^{O(r)}$.*

**Proof:** $\delta/2$ independence follows from Corollary 3.2.9 for $t = \frac{16t}{\delta}$ and Fact 3.2.8. Also, we can verify if a given SDP solution is $\delta/2$-independent or not in time polynomial in $n$. We now prove the later part.

As the conditioning maintains the marginal distribution of variables and because of the the Inequality (3.2.3) and (3.2.4), the constraints about the SDP cut value as well as the fraction of active edges that are cut remain valid in the conditioned solution. Hence, from Lemma 3.2.4 $\mathsf{SDP}^\star(h^\star)$ remains feasible. ■

#### 3.2.3.5 Rounding Procedure

In this section, we describe the rounding procedure for variables in $V \backslash S^\star$. The input to this procedure is 2 round Lasserre solution which is $\delta$-independent. We use a slight variation of GW rounding procedure to round the SDP vector solution. In particular, we want to maintain the bias of heavily biased random variable in our rounding procedure.

SDP gives the vector solution $\boldsymbol{v_{i,0}}, \boldsymbol{v_{i,1}}$ for all $i \in [n]$. Let $\mu_i = 2\,\mathbf{E}[X_i] - 1$, the expectation is according to the local distribution. Define $\boldsymbol{v_i} = \boldsymbol{v_{i,1}} - \boldsymbol{v_{i,0}}$. These $\boldsymbol{v_i}$ are the unit vectors (as $\|\boldsymbol{v_i}\|^2 = \|\boldsymbol{v_{i,1}} - \boldsymbol{v_{i,0}}\|^2 = \|\boldsymbol{v_{i,1}}\|^2 + \|\boldsymbol{v_{i,0}}\|^2 - 2\langle v_{i0}, v_{i1}\rangle = \Pr[X_i = 0] + \Pr[X_i = 1] - 0 = 1$). Let $\boldsymbol{w_i}$ be component of $\boldsymbol{v_i}$ orthogonal to $\boldsymbol{v}_{\varnothing}$ ($\boldsymbol{v_i} = \mu_i \boldsymbol{v}_{\varnothing} + \boldsymbol{w_i}$), $\|\boldsymbol{w_i}\|_2 = \sqrt{1 - \mu_i^2}$. Let $\overline{\boldsymbol{w}}_i$ be the normalized unit vector of $\boldsymbol{w_i}$. The rounding procedure is applied on vectors $\overline{\boldsymbol{w}}_i$ along with the "bias" of each variable $\langle \boldsymbol{v_i}, \boldsymbol{v}_{\varnothing}\rangle$. The rounding procedure is shown in Figure 3.6.

**Input**: $\delta$-independent $2$ round Lasserre solution, biases $\mu_i \in [-1, +1]$ and a function $f_R : [-1, 1] \to [-1, 1]$ which is bounded by above and below with some constant degree polynomials
**Output**: A partition of $V$.

1. Pick a random Gaussian vector $\boldsymbol{g}$ orthogonal to $\boldsymbol{v}_{\varnothing}$ with each co-ordinate distributed as $\mathcal{N}(0, 1)$.

2. For each $i \in [n]$

    - Calculate $\xi_i = \langle \boldsymbol{g}, \overline{\boldsymbol{w}}_i \rangle$.
    - Let $r_i \leftarrow f_R(\mu_i)$
    - Set $y_i = 1$ if $\xi_i \leqslant \Phi^{-1}(r_i/2 + 1/2)$, otherwise set $y_i = -1$. (Here, $\Phi$ is the Gaussian CDF)

Figure 3.6: Rounding procedure

#### 3.2.3.6 Analysis of the rounding procedure

We use the notation $\mathsf{poly}_{<1}(x)$ to denote a "polynomial" in $x$ with exponents as real numbers in $(0, 1)$, such that $\mathsf{poly}_{<1}(x) \to 0$ as $x \to 0$.

Note that if we simply use the rounding function $f_R(x) = x$ as used in [RT12] the we get for each instance, in expectation the cut produced by the rounding procedure is at least $0.85$ times the SDP value (and hence eventually $0.85$ approximation for simultaneous MAX-CUT). Here, we leverage the fact that the constraints on what rounding functions are good for us are mild compared to [RT12] as explained in Section 3.1.2.

**Lemma 3.2.11** *For a fixed low variance instance, the rounding procedure described in Figure 3.6 gives an approximation ratio $0.878001(1-3\varepsilon)$ in expectation for the following $f_R$,*

$$f_R(x) = 0.79 \cdot x + 0.07 \cdot x^3 + 0.14 \cdot x^7$$

**Proof:** The proof of this lemma is numerical. We arrive at a informal approximate value for the bound using Matlab code (0.878001) and verify it using computer assisted techniques. The multiplicative loss of $(1-3\varepsilon)$ is because of using SDP$^\star$. We elaborate on the exact constant 0.878001 that we get next. The probability $p_{ij}$ that a given edge $(i,j)$ is cut by the rounding procedure is a function of $\mu_i$ and $\mu_j$, whereas its SDP contribution is a quantity $q_{ij} := 1 - \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle / 2$. Thus to show a lower bound on approximation ratio it is sufficient to prove the same lower bound on $p_{i,j}/q_{ij}$ for all possible valid configurations of vectors. The program works in a recursive fashion, by continuously splitting the cube (all possible valid configuration) into sub-cubes. In each sub-cube, the program checks if either across all points in the region, the lower bound on $\alpha$ exceeds the approximation ratio we try to prove or if the upper bound on $\alpha$ is lower than the approximation ratio we try to prove. It proceeds with further division into smaller sub-cubes until one of the above is satisfied. If the latter is true at any point, the code returns a failure, and it returns a success if the entire region can be proved to come under the former case. The prover was adapted from [ABG12] and modified to suit our rounding procedure. For more details on the workings of the prover, refer [ABG12]. ■

**Remark 3.2.12** *It seems possible to improve the constant 0.878001 by using a different $f_R$ which is continuous and satisfies $f_R(1) = 1$ and $f_R(-1) = -1$ However we suspect that a serious new idea would be needed to get a $\alpha_{GW}$-approximation algorithm.*

We need the following lemma from [RT12].

**Lemma 3.2.13 ([RT12])** *Let $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ be the unit vectors, $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ be the components of $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ that are orthogonal to $\boldsymbol{v}_\varnothing$. Then $|\langle \boldsymbol{w_i}, \boldsymbol{w_j} \rangle| \leqslant 2I(X_i; X_j)$.*

Above lemma along with Lemma 3.2.10 implies that if we sample an edge $(i_1, i_2), (j_1, j_2) \sim \mathsf{actdist}_{S^\star}(\ell)$ then we have on average,

$$|\langle \boldsymbol{w_{i_1}}, \boldsymbol{w_{j_1}} \rangle| + |\langle \boldsymbol{w_{i_1}}, \boldsymbol{w_{j_2}} \rangle| + |\langle \boldsymbol{w_{i_2}}, \boldsymbol{w_{j_1}} \rangle| + |\langle \boldsymbol{w_{i_2}}, \boldsymbol{w_{j_2}} \rangle| \leqslant \delta.$$

The rounding procedure is assigning values $\pm 1$ to variables $y_i$ where $y_i$ is the variable for vertex $i \in V$ and its value decides on which side of cut the vertex $i$ is present in the final solution. Thus $y_i$ is a random variable taking values in $\{+1, -1\}$. We now wish to prove similar guarantee as the following lemma from [RT12], which relates the mutual information between the pair of rounded variables with the inner product of the corresponding vectors $w$.

**Lemma 3.2.14 ([RT12])** *For $f_R$ such that $f_R(x) = x$, if $|\langle \boldsymbol{w_i}, \boldsymbol{w_j} \rangle| \leqslant \delta$ then $I(y_i; y_j) \leqslant \delta^{1/3}$.*

In our case, we need that the mutual information between the events that a pair of edges are cut is small on average. Thus, our notion of local independence will be useful in proving this guarantee about mutual information.

**Lemma 3.2.15** *Fix $f_R$ to be the rounding function given by Lemma 3.2.11. For a pair of edges $(i_1, i_2)$ and $(j_1, j_2)$, suppose the vectors $w$ corresponding to their endpoints satisfy the following condition,*

$$\begin{aligned}|\langle \boldsymbol{w_{i_1}}, \boldsymbol{w_{j_1}} \rangle| + |\langle \boldsymbol{w_{i_1}}, \boldsymbol{w_{j_2}} \rangle| + \\ |\langle \boldsymbol{w_{i_2}}, \boldsymbol{w_{j_1}} \rangle| + |\langle \boldsymbol{w_{i_2}}, \boldsymbol{w_{j_2}} \rangle| \leqslant \delta\end{aligned}$$

*then $I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leqslant \mathsf{poly}_{<1}(\delta)$.*

**Proof:** Since $\overline{\boldsymbol{w}}_i$ is a normalized vector of $\boldsymbol{w_i}$ and $\|\boldsymbol{w_i}\| = \sqrt{1-\mu_i^2}$, we have

$$\left.\begin{array}{r}\sqrt{1-\mu_{i_1}^2} \cdot \sqrt{1-\mu_{j_1}^2} \cdot |\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_1} \rangle| \\ +\sqrt{1-\mu_{i_1}^2} \cdot \sqrt{1-\mu_{j_2}^2} \cdot |\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_2} \rangle| \\ +\sqrt{1-\mu_{i_2}^2} \cdot \sqrt{1-\mu_{j_1}^2} \cdot |\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_1} \rangle| \\ +\sqrt{1-\mu_{i_2}^2} \cdot \sqrt{1-\mu_{j_2}^2} \cdot |\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_2} \rangle| \end{array}\right\} \leqslant \delta. \tag{3.2.9}$$

Since the total sum is bounded and each quantity is non-negative, at least one of the three quantities in each summand is at most $\delta^{1/3}$. We use two crucial properties of the rounding procedure:

- For the heavily biased variable according to the local distribution, the rounding procedure also keeps the rounded value heavily biased and
- If two vectors $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ are nearly orthogonal, corresponding rounded values $y_i$ and $y_j$ are nearly independent.

We need following claim which we prove in Section 3.4.

**Claim 3.2.16** *If all these quantities $|\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_1}\rangle|$, $|\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_2}\rangle|$, $|\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_1}\rangle|$, and $|\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_2}\rangle|$ are upper bounded by $\delta^{1/3}$, then we can upper bound $I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$.*

We now formally prove the upper bound on $I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2})$ by case analysis. We use the following upper bound which follows from data processing inequality.

$$I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2}) \leqslant I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})).$$

We now bound the right hand side based on following case analysis.

- Case 1: If all these quantities $|\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_1}\rangle|$, $|\langle \overline{\boldsymbol{w}}_{i_1}, \overline{\boldsymbol{w}}_{j_2}\rangle|$, $|\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_1}\rangle|$, $|\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_2}\rangle|$ are upper bounded by $\delta^{1/3}$ then using Claim 3.2.16, we can upper bound $I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$

- Case 2: Consider the case when both the endpoints of an edge (w.l.o.g. of $(i_1, i_2)$) have large bias i.e. $\sqrt{1-\mu_{i_1}^2} \leqslant \delta^{1/3}$, $\sqrt{1-\mu_{i_2}^2} \leqslant \delta^{1/3}$. It implies,

  $$\min(|1-\mu_{i_1}|, |1+\mu_{i_1}|) \leqslant \delta^{1/3}$$

  $$\min(|1-\mu_{i_2}|, |1+\mu_{i_2}|) \leqslant \delta^{1/3}$$

  Assume both $\mu_{i_1}, \mu_{i_2} > 0$ (there cases can be handled in a similar way). Then we have, $1-\mu_{i_1} \leqslant \delta^{1/3}$ and $1-\mu_{i_2} \leqslant \delta^{1/3}$. Since the rounding procedure maintains the bias of a variable for a heavily biased variables, up to some constant polynomial factor, we have,

$$
\begin{aligned}
&I((y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \\
&\quad \leqslant H(y_{i_1}, y_{i_2}) \\
&\quad \leqslant H(y_{i_1}) + H(y_{i_2}) \\
&\quad = O(-(1 - \mathsf{poly}_{<1}(\mu_{i_1})) \log(1 - \mathsf{poly}_{<1}(\mu_{i_1})))+ \\
&\qquad O(-(1 - \mathsf{poly}_{<1}(\mu_{i_2})) \log(1 - \mathsf{poly}_{<1}(\mu_{i_2}))) \\
&\quad \leqslant \mathsf{poly}_{<1}(\delta).
\end{aligned}
$$

- Case 3: Consider the case when exactly two non-endpoints of an edge (w.l.o.g. of $(i_1, j_i)$) have large bias. This implies that $\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_2} \rangle \leqslant \delta^{1/3}$. Using the analysis of the previous case we have $H(y_{i_1}), H(y_{j_1}) \leqslant \mathsf{poly}_{<1}(\delta)$. Mutual information can be bounded as follows:

$$
\begin{aligned}
&I((y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \\
&\quad \leqslant H((y_{i_1}, y_{i_2})) - H((y_{i_1}, y_{i_2})|(y_{j_1}, y_{j_2})) \\
&\quad \leqslant H(y_{i_1}) + H(y_{i_2}) - H(y_{i_2}|(y_{j_1}, y_{j_2})) \\
&\quad = H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2}) && (3.2.10) \\
&\quad = \mathsf{poly}_{<1}(\delta) + I((y_{j_1}, y_{j_2}); y_{i_2}). && (3.2.11)
\end{aligned}
$$

Now,

$$
\begin{aligned}
&I((y_{j_1}, y_{j_2}), y_{i_2}) \\
&\qquad = H((y_{j_1}, y_{j_2})) - H((y_{j_1}, y_{j_2})|y_{i_2}) \\
&\qquad \leqslant H(y_{j_1}) + H(y_{j_2}) - H(y_{j_2}|y_{i_2}) \\
&\qquad = H(y_{j_1}) + I(y_{j_2}; y_{i_2}) \\
&\qquad = \mathsf{poly}_{<1}(\delta) + I(y_{j_2}; y_{i_2}).
\end{aligned}
$$

Therefore, we have

$$I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leqslant \mathsf{poly}_{<1}(\delta) + I(y_{j_2}; y_{i_2}).$$

From Claim 3.2.16, $I(y_{j_2}; y_{i_2})$ is bounded above by $\mathsf{poly}_{<1}(\delta)$ as $\langle \overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_2} \rangle \leqslant \delta^{1/3}$.

- Case 4: Consider the only remaining case in which exactly one variable, say $X_{i_1}$, has a large bias i.e. $\sqrt{1 - \mu_{i_1}^2} \leqslant \delta^{1/3}$. From (3.2.9), it implies that pairwise inner products of $\overline{\boldsymbol{w}}_{i_2}, \overline{\boldsymbol{w}}_{j_1}$ and $\overline{\boldsymbol{w}}_{j_2}$ are at most $\delta^{1/3}$. Hence by Claim 3.2.16, we have $I(y_{i_2}; (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$. As before from (3.2.10),

$$
\begin{aligned}
I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) &\leqslant H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2}) \\
&\leqslant \mathsf{poly}_{<1}(\delta).
\end{aligned}
$$

■

We can now upper bound the variance of a cut produced by the randomized rounding in graph $\ell \in \mathcal{L}$. Define $Y_\ell$ to be a random variable which is equal to the total weight of active edges cut by the rounding procedure.

$$Y_\ell = \sum_{C \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(C) e(g).$$

**Lemma 3.2.17** *Fix a rounding function $f_R$ given in Lemma 3.2.11 and let the* SDP *solution is $\delta$ independent then*

$$\mathsf{Var}(Y_\ell) \leqslant \frac{\mathsf{poly}_{<1}(\delta)}{\varepsilon^2} \mathbf{E}[Y_\ell]^2.$$

**Proof:** Let $\alpha := 0.8780$. Note that by Lemma 3.2.11, we have for an active edge $e(i,j)$,

$$\Pr[e(i,j) \text{ is cut }] \geqslant \alpha \cdot \frac{1 - \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle}{2}. \tag{3.2.12}$$

We now lower bound the expected value of $Y_\ell$.

$$\mathbf{E}[Y_\ell] = \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \Pr[e(i,j) \text{ is cut}]$$

( from (3.2.12))

$$\geqslant \alpha \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \frac{1 - \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle}{2}$$

$$= \alpha \cdot \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) (\| \boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{0},\boldsymbol{1})\}} \|_2^2 + \| \boldsymbol{v}_{\{(\boldsymbol{i},\boldsymbol{j}),(\boldsymbol{1},\boldsymbol{0})\}} \|_2^2)$$

( from (3.2.2))

$$\geqslant \alpha \cdot {}^{\varepsilon}\!/_{3} \cdot \mathsf{actdeg}_{S^\star}(\ell)$$

We can now bound the variance as follows:

$$
\begin{aligned}
\mathsf{Var}(Y_\ell) &= \sum_{i,j \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{Cov}\left[\frac{1-y_{i_1}y_{i_2}}{2}, \frac{1-y_{j_1}y_{j_2}}{2}\right] \\
&= \sum_{i,j \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\left(\frac{1}{4}\cdot \mathsf{Cov}[y_{i_1}y_{i_2}, y_{j_1}y_{j_2}]\right) \\
&\leqslant \sum_{i,j \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)[O(\sqrt{I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2})})]) \qquad \text{(from Lemma 3.2.15)} \\
&\leqslant \sum_{i,j \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{poly}_{<1}\left(\sum_{\substack{a\in\{i_1,i_2\},\\ b\in\{j_1,j_2\}}} |\langle \boldsymbol{w_a}, \boldsymbol{w_b}\rangle|\right) \\
&\qquad \text{(from Lemma 3.2.13)} \\
&\leqslant \sum_{i,j \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{poly}_{<1}\left(\mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)]\right) \\
&\leqslant \mathsf{actdeg}_{S^\star}(\ell)^2 \times \mathop{\mathbf{E}}_{i,j\sim \mathsf{actdist}_{S^\star}(\ell)} \mathsf{poly}_{<1} \mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)] \\
&\qquad \text{(from concavity of } \mathsf{poly}_{<1}) \\
&\leqslant \mathsf{actdeg}_{S^\star}(\ell)^2 \times \mathsf{poly}_{<1}\left(\mathop{\mathbf{E}}_{\substack{(i_1,i_2),\\ (j_1,j_2)}\sim \mathsf{actdist}_{S^\star}(\ell)} \mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)]\right) \\
&\leqslant \mathsf{poly}_{<1}(\delta)\cdot \mathsf{actdeg}_{S^\star}(\ell)^2,
\end{aligned}
$$

Thus, we have

$$
\mathsf{Var}(Y_\ell) \leqslant \frac{\mathsf{poly}_{<1}(\delta)}{\varepsilon^2}\mathbf{E}[Y_\ell]^2.
$$

■

**Corollary 3.2.18** *If we set $r := \text{poly}(k, 1/\varepsilon)$ then for every low variance instance $\ell \in [k]$,with probability at least $1 - 1/10k$ we have $\mathsf{val}(h^\star \cup g) \geqslant (0.878001 - 4\varepsilon)c_\ell$.*

**Proof:** Choosing $r$ a large constant (and thus $\delta$ very small), by Lemma 3.2.17 and application of Chebyshev's Inequality, we can deduce that with probability at least $1 - 1/10k$, we have $Y_\ell \geqslant (1-\varepsilon)\,\mathbf{E}[Y_\ell]$. Thus, with probability at least $1 - 1/10k$, we have,

$$\begin{aligned}
\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) &= \mathsf{val}(h^\star, \mathcal{E}_\ell) + Y_\ell \\
&\geqslant \mathsf{val}(h^\star, \mathcal{E}_\ell) + (1-\varepsilon)\,\mathbf{E}[Y_\ell] \\
&\geqslant (1-\varepsilon)\cdot \mathbf{E}[\mathsf{val}(h^\star, \mathcal{E}_\ell) + Y_\ell] \\
&= (1-\varepsilon)\cdot \mathbf{E}[\mathsf{val}(h^\star \cup g, W_\ell)] \\
&\geqslant (1-\varepsilon)\cdot 0.878001 \cdot (1-3\varepsilon)\cdot c_\ell \\
&\geqslant (0.878001 - 4\varepsilon)\cdot c_\ell,
\end{aligned}$$

where we have used Lemma 3.2.11 for the lower bound $\mathbf{E}[\mathsf{val}(h^\star \cup g, W_\ell)] \geqslant 0.878001 \cdot (1-3\varepsilon)c_\ell$, ■

#### 3.2.3.7 Post-Processing

**Lemma 3.2.19** *For all high variance instances $\ell \in [k]$, we have*

1. $\mathsf{actdeg}_{S^\star}(\ell) \leqslant 2(1-\gamma)^t$.
2. *For each of the first $t/2$ variables that were brought inside $S^\star$ because of instance $\ell$, the total weight of edges from $\mathcal{E}_\ell$ incident on each of that variable and totally contained inside $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.*

**Proof:** Consider any *high variance* instance $\ell \in [k]$. Initially, when $S = \varnothing$, we have $\mathsf{actdeg}_{\varnothing}(\mathcal{E}_\ell) \leqslant 2$ since the weight of every edge is counted at most twice, once for each of the 2 active vertices of the edge, and $\sum_{e \in \mathcal{E}} \mathcal{E}_\ell(e) = 1$. For every $v$, note that $\mathsf{actdeg}_{S_2}(v, \mathcal{E}_\ell) \leqslant \mathsf{actdeg}_{S_1}(v, \mathcal{E}_\ell)$ whenever $S_1 \subseteq S_2$.

Let $u$ be one of the vertices that ends up in $S^\star$ because of instance $\ell$. Let $S_u$ denote the set $S \subseteq S^\star$ just before $u$ was brought into $S^\star$. When $u$ is added to $S_u$, we know that $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma \cdot \mathsf{actdeg}_{S_u}(\ell)$. Hence, $\mathsf{actdeg}_{S_u \cup \{u\}}(\ell) \leqslant \mathsf{actdeg}_{S_u}(\ell) - \mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \leqslant (1-\gamma) \cdot \mathsf{actdeg}_{S_u}(\ell)$. Since $t$ vertices were brought into $S^\star$ because of instance $\ell$, and initially $\mathsf{actdeg}_{\varnothing}(\ell) \leqslant 2$, we get $\mathsf{actdeg}_{S^\star}(\ell) \leqslant 2(1-\gamma)^t$.

Now, let $u$ be one of the first $t/2$ vertices that ends up in $S^\star$ because of instance $\ell$. Since at least $t/2$ vertices are brought into $S^\star$ because of instance $\ell$, after $u$, as above, we get $\mathsf{actdeg}_{S^\star}(\ell) \leqslant (1-\gamma)^{t/2} \cdot \mathsf{actdeg}_{S_u}(\ell)$. Combining with $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma \cdot \mathsf{actdeg}_{S_u}(\ell)$, we get $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma(1-\gamma)^{-t/2}\mathsf{actdeg}_{S^\star}(\ell)$, which is at least $21 \cdot \mathsf{actdeg}_{S^\star}(\ell)$, by the choice of parameters. Since any edge incident on a vertex in $V \backslash S^\star$ contributes its weight to $\mathsf{actdeg}_{S^\star}(\ell)$, the total weight of edges incident on $u$ and totally contained inside $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$ as required. ■

We now describe a procedure PERTURB (see Figure 3.7) which takes $h^\star : S^\star \to \{0,1\}$ and $g : V \backslash S^\star \to \{0,1\}$, and produces a new $h : S^\star \to \{0,1\}$ such that for all (low variance as well as high variance) instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell)$ is not much smaller than $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$, and furthermore, for all high variance instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell)$ is large. The procedure works by picking a special vertex in $S^\star$ for every high variance instance and perturbing the assignment of $h^\star$ to these special vertices. The partial assignment $h$ is what we will be using to argue that Step 1d of the algorithm produces a good Pareto approximation. More formally, we have the following Lemma.

**Input**: $h^\star : S^\star \to \{0,1\}$ and $g : V \backslash S^\star \to \{0,1\}$
**Output**: A perturbed assignment $h : S^\star \to \{0,1\}$.

1. Initialize $h \leftarrow h^\star$.
2. For $\ell = 1, \ldots, k$, if instance $\ell$ is a high variance instance case (i.e., $\mathsf{count}_\ell = t$), we pick a special variable $v_\ell \in S^\star$ associated to this instance as follows:
    (a) Let $B = \{v \in V \mid \exists \ell \in [k] \text{ with } \sum_{e \in \mathcal{E}, e \ni v} \mathcal{E}_\ell(e) \cdot e(h \cup g) \geqslant \frac{\varepsilon}{2k} \cdot \mathsf{val}(h \cup g, \mathcal{E}_\ell)\}$. Since the weight of each edge is counted at most twice, we know that $|B| \leqslant \frac{4k^2}{\varepsilon}$.
    (b) Let $U$ be the set consisting of the first $t/2$ vertices brought into $S^\star$ because of instance $\ell$.
    (c) Since ${}^{t}\!/_{2} > |B| + k$, there exists some $u \in U$ such that $u \notin B \cup \{v_1, \ldots, v_{\ell-1}\}$. We define $v_\ell$ to be $u$.
    (d) By Lemma 3.2.19, the total $\mathcal{E}_\ell$ weight of edges that are incident on $v_\ell$ and only containing vertices from $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. We update $h$ by setting $h(v_\ell)$ to be that value from $\{0,1\}$ such that at least half of the $\mathcal{E}_\ell$ weight of these edges is satisfied.
3. Return the assignment $h$.

Figure 3.7: Procedure PERTURB for perturbing the optimal assignment

**Lemma 3.2.20** *For the assignment $h$ obtained from Procedure* PERTURB *(see Figure 3.7), for each $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$. Furthermore, for each* high variance *instance $\mathcal{E}_\ell$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.*

**Proof:** Consider the special vertex $v_\ell$ that we choose for *high variance* instance $\ell \in [k]$. Since $v_\ell \notin B$, the edges incident on $v_\ell$ only contribute at most a $\varepsilon/2k$ fraction of the objective value in each instance. Thus, changing the assignment $v_\ell$ can reduce the value of any instance by at most a $\frac{\varepsilon}{2k}$ fraction of their current objective value. Also, we pick different special variables for each *high variance* instance. Hence, the total effect of these perturbations on any instance is that it reduces the objective value (given by $h^\star \cup g$) by at most $1 - (1 - \frac{\varepsilon}{2k})^k \leqslant \frac{\varepsilon}{2}$ fraction. Hence for all instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$.

For a *high variance instance* $\ell \in [k]$, since $v_\ell \in U$, the vertex $v_\ell$ must be one of the first $t/2$ variables brought into $S^\star$ because of $\ell$. Hence, by Lemma 3.2.19 the total weight of edges that are incident on $v_\ell$ and entirely contained inside $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Hence, there is an assignment to $v_\ell$ that satisfies at least at least half the weight of these MAX-CUT constraints in $\ell$. At the end of the iteration when we pick an assignment to $v_\ell$, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 10 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Since the later perturbations do not affect value of this instance by more than $\varepsilon/2$ fraction, we get that for the final assignment $h$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot 10 \cdot \mathsf{actdeg}_{S^\star}(\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. ■

**Theorem 3.2.21** *Suppose we're given $\varepsilon \in (0, 1/5]$, $k$ simultaneous* MAX-CUT *instances $\mathcal{E}_1, \ldots, \mathcal{E}_k$ on $n$ variables, and target objective value $c_1, \ldots, c_k$ with the guarantee that there exists an assignment $f^\star$ such that for each $\ell \in [k]$, we have $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$. Then, the algorithm* ALG-SIM-MAXCUT *runs in time $\exp(k^3/\varepsilon^2 \log(k/\varepsilon^2)) \cdot n^{poly(k)}$, and with* *probability at least* $0.9$, *outputs an assignment* $f$ *such that for each* $\ell \in [k]$, *we have,* $\mathsf{val}(f, \mathcal{E}_\ell) \geqslant (0.878001 - 5\varepsilon) \cdot c_\ell$.

**Proof:** Let $\alpha := 0.878001$. By Corollary 3.2.18 and a union bound, with probability at least $0.9$, over the choice of $g$, we have that for *every* low variance instance $\ell \in [k]$, $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 4\varepsilon) \cdot c_\ell$. Henceforth we assume that the assignment $g$ sampled in Step 1c of the algorithm is such that this event occurs. Let $h$ be the output of the procedure PERTURB given in Figure 3.7 for the input $h^\star$ and $g$. By Lemma 3.2.20, $h$ satisfies

1. For every instance $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$.

2. For every high variance instance $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.

We now show that the desired Pareto approximation behavior is achieved when $h$ is considered as the partial assignment in Step 1d of the algorithm. We analyze the guarantee for low and high variance instances separately.

For any *low variance* instance $\ell \in [k]$, from property 1 above, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$. Since we know that $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 4\varepsilon) \cdot c_\ell$, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 5\varepsilon) \cdot c_\ell$.

For every high variance instance $\ell \in [k]$, since $h^\star = f^\star|_{S^\star}$, for any $g$ we must have,

$$\begin{aligned}\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) &\geqslant \mathsf{val}(f^\star, \mathcal{E}_\ell) - \mathsf{actdeg}_{S^\star}(\ell)\\ &\geqslant c_\ell - \mathsf{actdeg}_{S^\star}(\ell)\end{aligned}$$

Combining this with properties 1 and 2 above, we get,

$$\begin{aligned}
&\mathsf{val}(h \cup g, \mathcal{E}_\ell) \\
&\quad \geqslant (1 - \varepsilon/2) \cdot \max\{c_\ell - \mathsf{actdeg}_{S^\star}(\ell), 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)\} \\
&\quad \geqslant (\alpha - \varepsilon) \cdot c_\ell.
\end{aligned}$$

Thus, for all instances $\ell \in [k]$, we get $\mathsf{val}(h \cup g) \geqslant (\alpha - 5\varepsilon) \cdot c_\ell$. Since we are taking the best assignment $h \cup g$ at the end of the algorithm ALG-SIM-MAXCUT, the theorem follows.

■

Plugging the appropriate value of $\varepsilon$ in Theorem 3.2.21 completes the proof of $0.8780$-factor Pareto approximation (and hence min approximation) for simultaneous MAX-CUT for arbitrary constant $k$.

## 3.3 Open Questions

The main open question we would like to highlight is the question of determining optimal approximability and inapproximability results for simultaneous approximation of constraint satisfaction problems (CSPs). In particular, it would be very interesting to develop techniques for showing nontrivial hardness of approximation in this context.

# Acknowledgement

We would like to thank the authors of [ABG12] for making the prover code available for us. Our implementation of prover involves minor modifications of their code to suit our rounding algorithm. We also want to thank anonymous referees for helpful comments.

## 3.4 Deferred Proofs

### 3.4.1 Proof of Claim 3.2.16

We need following bounds on the gaussian random variables.

**Claim 3.4.1** *For all* $x > 0$, $\Pr_{g \sim \mathcal{N}(0,1)}[|g| > x] \leqslant e^{-x^2/2}$.

**Claim 3.4.2** *For all* $1 > x > 0$, $\Pr_{g \sim \mathcal{N}(0,1)}[|g| < x] \leqslant x$.

**Random process $\mathcal{P}$:** Let $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4 \in \mathbf{R}^4$ be unit vectors and $\mu_1, \mu_2, \mu_3, \mu_4$ be any real numbers. Consider the following random variables $(y_1, y_2, y_3, y_4)$ where $y_i \in \{-1, +1\}$ which are sampled as follows: Pick a random vector $\boldsymbol{g} := (g_1, g_2, g_3, g_4) \in \mathbf{R}^4$ with each entry distributed as $\mathcal{N}(0,1)$. Set

$$\begin{aligned} y_i &= -1 \qquad \text{if } \langle \boldsymbol{g}, \boldsymbol{w}_i \rangle \leqslant \mu_i, \\ &= +1 \qquad \text{otherwise.} \end{aligned}$$

The following lemmas gives sufficient conditions when $I(y_1, y_2; y_3, y_4)$ is *small*.

**Lemma 3.4.3** *Suppose* $|\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle| \leqslant \delta$ *for all* $i, j \in [4]$*,* $i \neq j$ *and* $y_i s$ *are sampled according to the random process* $\mathcal{P}$*, then for all* $\boldsymbol{b} \in \{-1, +1\}^4$*, we have*

$$\left| \Pr[(y_1, y_2, y_3, y_4) = \boldsymbol{b}] - \prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] \right| = O(\delta^{1/4}),$$

*In fact, the joint distribution on any subset of variables is close to its product distribution* pointwise *with an additive error of at most* $O(\delta^{1/4})$*.*

**Proof:** Assume that $0 < \delta < 1/100$ (otherwise, the lemma is trivial). Let $\boldsymbol{e}_i$ is a unit vector with $1$ in the $i^{\text{th}}$ coordinate. By rotational symmetry, we can assume that $\langle \boldsymbol{w}_i, \boldsymbol{e}_i \rangle \geqslant 1 - 20\delta$ for all $i$. We can write vector $\boldsymbol{w}_i = \sqrt{1-\delta_i}\boldsymbol{e}_i + \sqrt{\delta_i}\boldsymbol{\eta}_i$ where $\boldsymbol{\eta}_i$ is a unit vector orthogonal to $\boldsymbol{e}_i$. The conditions on inner products therefore imply each $\delta_i < 40\delta$. We will prove the lemma for $\boldsymbol{b} = (-1, -1, -1, -1)$ (all other cases are similar). We have,

$$\begin{aligned} \Pr[y_i = -1, \forall i \in [4]] &= \Pr[\forall i, \langle \boldsymbol{g}, \boldsymbol{w}_i \rangle \leqslant \mu_i] \\ &= \Pr[\forall i, \sqrt{1-\delta_i} g_i + \sqrt{\delta_i} \langle \boldsymbol{g}, \boldsymbol{\eta}_i \rangle \leqslant \mu_i] \end{aligned}$$

Let $B$ be the following event,
$B$ : There exists $1 \leqslant i \leqslant 4$, such that $|\langle \boldsymbol{g}, \boldsymbol{\eta}_i \rangle| \geqslant 1/\delta^{1/4}$.

By union bound,

$$\begin{aligned}\Pr[B] &= \sum_i \Pr[|\langle \boldsymbol{g}, \boldsymbol{\eta}_i\rangle| \geqslant 1/\delta^{1/4}] \\ &\leqslant 4 \cdot \Pr[|\langle \boldsymbol{g}, \boldsymbol{\eta}_1\rangle| \geqslant 1/\delta^{1/4}] \\ &= 4 \cdot \Pr_{g\sim\mathcal{N}(0,1)}[|g| \geqslant 1/\delta^{1/4}] \\ &\leqslant 4e^{-\frac{1}{2\sqrt{\delta}}},\end{aligned}$$

where last inequality uses Claim 3.4.1. Now,

$$\begin{aligned}\Pr[y_i = -1, \forall 1 \leqslant i \in [4]] &= \Pr[B] \cdot \Pr[y_i = -1, \forall i \in [4] | B] + \\ &\qquad \Pr[\overline{B}] \cdot \Pr[y_i = -1, \forall i \in [4] | \overline{B}] \\ &\leqslant 4e^{-\frac{1}{2\sqrt{\delta}}} \cdot 1 + \Pr[y_i = -1, \forall i \in [4] | \overline{B}], \qquad (3.4.1)\end{aligned}$$

We now estimate the probability conditioned on event $\overline{B}$.

$$\begin{aligned}\Pr[y_i = -1, \forall i \in [4] | \overline{B}] &= \Pr[\forall i, \sqrt{1-\delta_i} g_i + \sqrt{\delta_i}\langle \boldsymbol{g}, \boldsymbol{\eta}_i\rangle \leqslant \mu_i | \overline{B}] \\ &\leqslant \Pr[\forall i, \sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \quad (g_i \text{ independent}) \\ &= \prod_i \Pr[\sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \quad (\text{using } \delta_i \leqslant 40\delta) \\ &\leqslant \prod_i \Pr[\sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{40}\delta^{1/4} \quad (\text{using } \delta_i \leqslant 1/2) \\ &\leqslant \prod_i \Pr[g_i \leqslant (1+\delta_i)(\mu_i + \sqrt{40}\delta^{1/4})] \quad (\text{using } \delta_i \leqslant 1/2) \\ &\leqslant \prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 3/2 \cdot \sqrt{40}\delta^{1/4})] \\ &\leqslant \prod_i \Pr[g_i \leqslant (\mu_i + \delta_i\mu_i + 15\delta^{1/4})].\end{aligned}$$

We now analyse the above probability in cases, and show the following:

$$\Pr[g_i \leqslant \mu_i + \delta_i \mu_i + 15\delta^{1/4})] \leqslant \prod_i \Pr[g_i \leqslant \mu_i] + O(\delta^{1/4}) \tag{3.4.2}$$

Notice that

$$\begin{aligned}
\prod_i \Pr[g_i \leqslant \mu_i + c\delta^{1/4}] &\leqslant \prod_i \Pr[g_i \leqslant \mu_i] + \Pr[|g_i| \leqslant c\delta^{1/4}] \\
\text{(from Claim 3.4.2)} &\leqslant \left( \prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] + c\delta^{1/4} \right) \\
&\leqslant \prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] + O(\delta^{1/4}) \qquad (3.4.3)
\end{aligned}$$

- Case 1: $\mu_i < 0$.

  In this case, we can directly say the following.

  $$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i \mu_i + 15\delta^{1/4})] \leqslant \prod_i \Pr[g_i \leqslant \mu_i + 15\delta^{1/4}].$$

- Case 2: $0 \leqslant \mu_i \leqslant \frac{10}{\delta^{3/4}}$ We can say the following because $\delta_i < 40\delta$.

  $$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i \mu_i + 15\delta^{1/4}] \leqslant \prod_i \Pr[g_i \leqslant \mu_i + O(\delta^{1/4})]$$

- Case 3: $\mu_i > \frac{10}{\delta^{3/4}}$ In this case, since $\mu_i$ is large, we have the following from Claim 3.4.1.

  $$\prod_i \Pr[g_i \leqslant \mu_i] \geqslant 1 - o(\delta^{1/4})$$

Therefore,

$$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 15\delta^{1/4}] \leqslant 1 \leqslant \prod_i \Pr[g_i \leqslant \mu_i] + o(\delta^{1/4})$$

Form (3.4.1), (3.4.2) and (3.4.3) we get

$$\Pr[(y_1, y_2, y_3, y_4) = \boldsymbol{b}] - \prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] \leqslant O(\delta^{1/4}).$$

The other direction can be shown in an analogous way. ■

We can now bound the Mutual information between $(y_1, y_2)$ and $(y_3, y_4)$ if the vectors $\boldsymbol{w}_i$ satisfy the condition from Lemma 3.4.3

**Lemma 3.4.4** *Suppose $|\langle \boldsymbol{w}_i, \boldsymbol{w}_j\rangle| \leqslant \delta$ for all $i, j \in [4]$ and $i \neq j$, then $I((y_1, y_2)\ ;\ (y_3, y_4)) \leqslant \mathsf{poly}_{<1}(\delta)$, where $y_i$ are sampled according to the random process $\mathcal{P}$.*

**Proof:** The lemma follows from Lemma 3.4.3 as the distribution is *close* to the product distribution.

To formally prove the lemma, first we assume that each of the random variables $y_i$ is not heavily biased i.e. $\Pr[y_i = -1] \in [\delta^{1/100}, 1 - \delta^{1/100}]$. Using the definition of mutual information,

$$\begin{aligned} &I((y_1, y_2); (y_3, y_4)) \\ &\qquad = \sum_{\substack{b_1,b_2,b_3,b_4 \\ \{-1+1\}}} \left[ [\Pr[\boldsymbol{y} = \boldsymbol{b}] \cdot \log \frac{\Pr[\boldsymbol{y} = \boldsymbol{b}]}{\Pr[(y_1, y_2) = (b_1, b_2)] \cdot \Pr[(y_3, y_4) = (b_3, b_4)]} \right] \end{aligned} \tag{3.4.4}$$

Form Lemma 3.4.3, we have

$$\Pr[(y_1, y_2) = (b_1, b_2)] \geqslant \Pr[y_1 = b_1] \Pr[y_2 = b_2] - O(\delta^{1/4})$$
$$\Pr[(y_3, y_4) = (b_3, b_4)] \geqslant \Pr[y_3 = b_3] \Pr[y_4 = b_4] - O(\delta^{1/4})$$

Plugging in and simplifying (3.4.4), we get

$$I((y_1, y_2); (y_3, y_4)) \leqslant \sum_{b_1, b_2, b_3, b_4 \{-1+1\}} \Pr[\boldsymbol{y} = \boldsymbol{b}] \cdot \log \frac{\prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] + O(\delta^{1/4})}{\prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] - O(\delta^{1/4})}$$

As each variable is not heavily biased, we have $\prod_{1 \leqslant i \leqslant 4} \Pr[y_i = b_i] \geqslant \delta^{1/25}$ and hence the log in the above expression can be upper bounded by $\log \frac{\delta^{1/25} + O(\delta^{1/4})}{\delta^{1/25} - O(\delta^{1/4})}$ which is at most $\log(1 + O(\delta^{1/10})) \leqslant O(\delta^{1/10})$. Hence we have

$$I((y_1, y_2); (y_3, y_4)) \leqslant O(\delta^{1/10}).$$

If a variable is heavily biased, suppose say $y_1$ has large bias, then we can claim $I((y_1, y_2); (y_3, y_4)) \leqslant \mathsf{poly}_{<1}(\delta) + I(y_2; (y_3, y_4))$ using derivation similar to ( 3.2.11) and then proceed by upper bounding $I(y_2; (y_3, y_4))$ in a similar fashion as above.

■

**Proof of Claim 3.2.16:** The proof follows from Lemma 3.4.4 noting the fact that the upper bound is independent of $\mu_i$.

### 3.4.2 Proof of Lemma 3.2.2

**Proof:** Item 1 of the lemma follows from Chebyshev's inequality. We now focus on the proof of Item 2. We have

$$\begin{aligned} \mathsf{Uvar}_\ell &\geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2 \\ \Rightarrow \sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e') &\geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2 \end{aligned}$$

Let $e_0$ be an edge in $\mathsf{Active}(S)$ that maximizes $\sum_{e \sim_S e_0} \mathcal{E}_\ell(e)$. We can now upper bound the expression on the left as follows

$$\sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e') \leqslant \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

Therefore, we have

$$\begin{aligned} \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) &\geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2 \\ &\geqslant \delta_0\varepsilon_0^2 \cdot \tau^2 \cdot \left( \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \right)^2 \\ \Rightarrow \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) &\geqslant \delta_0\varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \end{aligned}$$

Let $v$ be the end vertex of $e_0$ that has greater weight of active edges adjacent to it, $v \in V \backslash S$. We can say the following

$$\mathsf{actdeg}_S(v, \ell) \geqslant \frac{1}{2} \cdot \delta_0\varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

From the definition of $\mathsf{actdeg}_S(\ell)$, we can say the following

$$\mathsf{actdeg}_S(\ell) \leqslant 2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e),$$

as each edge could contribute at most twice to the sum, once for each end vertex. This gives us the following required result.

$$\mathsf{actdeg}_S(v,\ell) \geqslant \frac{1}{4} \cdot \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \mathsf{actdeg}_S(\ell).$$

■

# Chapter 4

# Improved Hardness for 3LIN

## 4.1 Introduction

In this paper, we study the 3-LIN problem. An instance of 3-LIN consists of a set of $n$ variables over $\mathbf{F}_2$ and a set of $m$ equations that contain at most three variables each, and the goal is to find an assignment to the $n$ variables that satisfies the most number of equations.[1] If the given set of linear equations admits an assignment that satisfies every equation, then one such assignment can be found in polynomial time by Gaussian elimination. However, the general problem of finding the most of number of equations is **NP**-hard when the instance does not admit a satisfying assignment, and a large amount of research has been done on the limit of polynomial time approximation algorithms.

Assigning random values satisfies exactly half the equations in expectation, and gives a $1/2$-approximation algorithm. Håstad and Venkatesh [HV02] get an approximation factor of $1/2 + 1/O(\sqrt{m})$, which was improved by Khot and

[1] This maximization version is also known as MAX 3-LIN in the literature.

Naor [KN07] to $1/2 + O(\sqrt{\log n/n})$.

From the hardness side, there are strong hardness results even when the instance is *almost-satisfiable.* For $1 \geqslant c > s > 0$, let GAP 3-LIN$(c, s)$ denote the problem of distinguishing whether the given instance of 3-LIN is at least $c$-satisfiable or at most $s$-satisfiable. Håstad's classic hardness results [Hås01] show the following.

**Theorem 4.1.1 ([Hås01])** *The following hardness results for* GAP 3-LIN *hold.*

1. *For any constant $\varepsilon > 0$,* GAP 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *is* **NP***-hard.*
2. *There exists a constant $c > 0$ such that for $\varepsilon = 1/(\log n)^c$, there is no polynomial time algorithm that solves* GAP 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *unless* $\mathbf{NP} \subseteq \mathbf{DTIME}[n^{O(\log\log n)}]$.

Håstad's results are proved by giving the reduction from LABEL COVER to 3-LIN. LABEL COVER is a common starting point for hardness results, and we define the optimization problem below.

**Definition 4.1.2 (LABEL COVER)** *An instance of* LABEL COVER *contains a regular bipartite multi- graph $G = (A, B, E)$ and two finite sets $\Sigma_A$ and $\Sigma_B$, where $|\Sigma_A| \geqslant |\Sigma_B|$. Every vertex in $A$ is supposed to get a label in $\Sigma_A$, and every vertex in $B$ is supposed to get a label in $\Sigma_B$. For each edge $e \in E$ there is a projection $\pi_e : \Sigma_A \to \Sigma_B$. Given a labeling to the vertices of the graph, i.e., functions $\phi_A : A \to \Sigma_A$ and $\phi_B : B \to \Sigma_B$, an edge $e = (a, b) \in E$ is said to be "satisfied" if $\pi_e(\phi_A(a)) = \phi_B(b)$. For $1 \geqslant c > s > 1$,* GAP LABEL COVER$(c, s)$ *is the problem if distinguishing whether the given instance of* LABEL COVER *is at least $c$-satisfiable or at most $s$-satisfiable.*

The aforementioned Håstad's theorem can also be re-stated in terms of a reduction from GAP LABEL COVER$(1, \delta)$ as follows.

**Theorem 4.1.3 ([Hås01])** *For every $\varepsilon \in (0,1)$ and positive integer $\ell$, there exists a $\delta = poly(\varepsilon)$ and a $poly(n, 2^{\ell}, 2^{1/\varepsilon})$-time reduction to* GAP 3-LIN$(1-\varepsilon, 1/2+\varepsilon)$ *from $n$-sized instances of* GAP LABEL COVER$(1, \delta)$ *with label size $\ell$.*

When [Hås01] was published, the hardness of LABEL COVER was achieved by the PCP theorem [AS98, ALM$^+$98] and parallel repetition [Raz98]. More precisely, GAP LABEL COVER$(1, \varepsilon)$ with label size $\mathrm{poly}(1/\delta)$ was **NP**-hard under $\mathrm{poly}(n^{\log 1/\delta})$-time reductions. The two results of Håstad stated in Theorem 4.1.1 follow from this hardness of GAP LABEL COVER and Theorem 4.1.3 by setting $\delta$ to be an arbitrarily small constant and $1/\log n$ respectively. Since achieving a subconstant soundness for LABEL COVER by parallel repetition requires a superpolynomial blowup in the instance size, $\varepsilon > 0$ could not be taken to subconstant under *polynomial time* reductions. Later in a celebrated paper, Moshkovitz and Raz [MR08] gave an improved hardness of LABEL COVER that achieves subconstant error under polynomial time reductions. Their main result can be stated as follows.

**Theorem 4.1.4 ([MR08, Theorem 11])** *For every $n$, and every $\delta > 0$ (that can be any function of $n$),* 3-SAT *on inputs of size $n$ can be reduced to* GAP LABEL COVER$(1, \delta)$ *when* LABEL COVER *instance has $n^{1+o(1)} \cdot poly(1/\delta)$ vertices and $|\Sigma_A| \leqslant \exp(poly(1/\delta))$, $|\Sigma_B| \leqslant poly(\log 1/\delta)$.*

A corollary of the above result, obtained by combining it with Håstad's reduction from Theorem 4.1.3, is that given a system of linear equations, it is NP-hard to distinguish between cases where $1 - o(1)$ fraction of equations are satisfied vs at most $1/2 + o(1)$ fraction are satisfied, where the $o(1)$ term is $1/(\log\log n)^{-\Omega(1)}$.

**Theorem 4.1.5 ([MR08])** *There exists a constant $c > 0$ such that for $\varepsilon = {}^{1}\!/\!_{(\log\log n)^c}$,* Gap 3-Lin$(1-\varepsilon, {}^{1}\!/\!_{2}+\varepsilon)$ *is* **NP***-hard.*

Later, an improved parallel repetition by Dinur and Steurer [DS14] allowed $c$ to be an arbitrary constant.

The above route prove hardness of 3-Lin is restricted by the large size of the alphabet in the resulting Label Cover instance in Theorem 4.1.4. Quantitatively, the alphabet size is exponential in poly$({}^{1}\!/\!_{\varepsilon})$. The fact that the long code in Håstad's reduction has size exponential in the alphabet size restricts $\varepsilon = {}^{1}\!/\!_{(\log\log n)^{O(1)}}$.

Our main contribution for 3-Lin is to bring $\varepsilon$ in the above result down to ${}^{1}\!/\!_{(\log n)^c}$ for any constant $c$, while keeping the size of the reduced instance polynomial (albeit the reduction becomes randomized).

**Theorem 4.1.6 (Main)** *For any constant $c > 0$ and $\varepsilon = {}^{1}\!/\!_{(\log n)^c}$, there is no polynomial time algorithm for* Gap 3-Lin$(1-\varepsilon, {}^{1}\!/\!_{2}+\varepsilon)$ *unless* **NP** $\subseteq$ **BPP**.

We get around the above alphabet barrier by starting with a reduction that would make the resulting Label Cover *linear*, and use Hadamard codes instead of long codes. Since the Hadamard code keeps the reduction size polynomial in the alphabet size, we can take $\varepsilon = {}^{1}\!/\!_{(\log n)^{\Omega(1)}}$. A similar idea was previously used by Khot [Kho01]. We define Linear Label Cover as follows.

**Definition 4.1.7 (Linear Label Cover)** *A* Linear Label Cover *is a special case of* Label Cover *where the alphabets are of the form $\Sigma_A = \mathbb{F}_2^a, \Sigma_B = \mathbb{F}_2^b$ where $a, b$ are natural numbers. Each projection $\pi : \mathbb{F}_2^a \to \mathbb{F}_2^b$ is* affine *in the sense that $\pi(x) = \alpha x + \beta$ for some $\alpha \in \mathbf{F}_2^{b\times a}, \beta \in \mathbf{F}_2^b$.*

*For* $1 \geqslant c > s > 0$*, the* GAP LINEAR LABEL COVER$(c, s)$ *is defined similarly to* GAP LABEL COVER$(c, s)$.

We prove the following hardness result for LINEAR LABEL COVER, which may be of independent interest.

**Theorem 4.1.8 (Hardness of Linear Label Cover)** *For any constant* $c > 0$*, for* $\delta = 1/(\log n)^c$*, there is no polynomial time algorithm for* GAP LINEAR LABEL COVER$(1 - \delta, \delta)$ *unless* **NP** $\subseteq$ **BPP***, when* LABEL COVER *instance has poly*$(n)$ *vertices and* $|\Sigma_A| =$ *poly*$(n)$, $|\Sigma_B| = \mathsf{polylog}(n)$.

### 4.1.1 Proof Ideas

Our main technical contribution is Theorem 4.1.8 for LINEAR LABEL COVER, essentially proving a linear analogue of the Moshkovitz-Raz PCP [MR08] followed by the Dinur-Steurer parallel repetition [DS14]. The proof is given through a long sequence of reductions. We split them in 3 major steps.

1. Interestingly, the starting point of our reduction is again the hardness of (not necessarily linear) LABEL COVER proved by Moshkovitz and Raz [MR08] augmented by Dinur and Steurer [DS14], proving **NP**-hardness of GAP LABEL COVER$(1, 1/\log^c n)$ for any $c > 0$, while keeping the reduction size and the alphabet size polynomial. In Section 4.2, we give a *randomized* reduction from this LABEL COVER to GAP LIN$(1 - 1/\log^c n, 0.9)$. This style of reduction appeared previous from LABEL COVER to CLOSEST VECTOR PROBLEM [Kho10]. Note that the standard proof of the PCP theorem encodes 3-SAT (or CIRCUIT SAT) by solving quadratic equations over $\mathbf{F}_2$, and this is essentially the only place that needs where nonlinearity occurs.

Our hardness result for solving linear equations with completeness very close to (but not exactly) $1$ allows us to follow previous PCP constructions that will ensure linearity of the LABEL COVER instance in the subsequent steps.

2. To prove the hardness of LINEAR LABEL COVER given the above hardness of LIN, we closely follow the steps of Dinur and Harsha [DH13], who gave a simpler and modular proof of [MR08]. The two basic building blocks in their proof are robust PCPs and decodable PCPs. Robust PCPs are PCPs where in the soundness case, for any proof and most random choices of the verifier, not only are the local views non-accepting, but they are also very far from any accepting string. It is indeed equivalent to LABEL COVER. Using our previous hardness for LINas the starting point and following the standard robust PCP construction (e.g., low-degree extension and sum-check protocol), we can prove a polynomial time reduction to LINEAR LABEL COVER$(1 - 1/\log^c n, 1/\log^c n)$ for any $c > 1$, but the alphabet size will be always $\exp(\log^{c_0} n)$ for some $c_0 > 1$, which is superpolynomial.

3. The second building block, decodable PCP, is similar to robust PCP with the additional requirement that the prover is given a position $i$ in the original string and supposed to output the value of the $i$th position if the given proof is a honest encoding of a valid original string. The main idea of Dinur and Harsha [DH13] is to iteratively compose a robust PCP with a suitable decodable PCP, where the composed PCP is another robust PCP that consists of a decodable PCP for each constraint of the original robust

PCP. This iteratively reduces the query complexity and the alphabet size of the robust PCP, which is related to the alphabet size of the equivalent LABEL COVER instance. This iterative composition is interleaved and preprocessed by technical operations that reduce the alphabet size of the robust PCP and make it regular.

Once these two building blocks are linear, the operations of [DH13] can be used verbatim in our construction. Our main observation is that every step of this construction preserves (1) the robust completeness $1-\delta$ for some $\delta = {}^{1}\!/_{\mathsf{polylog}(n)}$, and (2) the linearity, which were not issues in [DH13]. In Section 4.3, we introduce the basic building blocks and these operations, and show how they preserve robust completeness and linearity. These iterative operations will eventually reduce the alphabet size of the LINEAR LABEL COVER polynomial, proving Theorem 4.1.8.

After the hardness of LINEAR LABEL COVER is proved, we give a reduction from LINEAR LABEL COVER with the above parameters to 3-LIN with the required parameters. We do this by composing with the Hadamard Code to get a $(1-\varepsilon)$ vs $({}^{1}\!/_{2}+\varepsilon)$ **NP**-hardness result for 3LIN. Similar PCP constructions based on Hadamard codes were presented in [Kho01]. Details of this step can be found in Section 4.4.

## 4.2 Reduction to System of Linear Equations

In this section, we first prove the hardness of approximate solving linear equations over large fields, where each equation can involve as many variables as possible. It will serve as the starting point towards proving hardness of LINEAR

LABEL COVER.

**Theorem 4.2.1** *For any constant $c > 0$, $\varepsilon = {}^{1}\!/\!_{(\log n)^c}$,* GAP LIN$(1 - {}^{1}\!/\!_{(\log n)^c}, 0.9)$ *is* **NP**-*hard under polynomial time randomized reductions.*

**Proof:** The proof starts from the following hardness of LABEL COVER, which is obtained by combining the main result of Moshkovitz and Raz [MR08] with the parallel repetition of Dinur and Steurer [DS14].

**Theorem 4.2.2 ([MR08, DS14])** *For any constant $c > 0$, and for $\delta = {}^{1}\!/\!_{(\log n)^c}$, when the* LABEL COVER *instance satisfies $|\Sigma_A|, |\Sigma_B| \leqslant |A| + |B|$,* GAP LABEL COVER$(1, \delta)$ *is* **NP**-*hard.*

Let $G = (A, B, E)$, $\Sigma_A$, $\Sigma_B$, and $\{\pi_e\}_{e \in E}$ be an instance of LABEL COVER. We show a reduction to LINover $\mathbf{F}_2$ where

- If all LABEL COVER edges are satisfiable, at least $(1 - \frac{1}{|\Sigma_A|})$ fraction of equations are satisfiable.
- If at most $\delta$ fraction of LABEL COVER edges are satisfiable, at most $(1 - \frac{1}{(\delta|\Sigma_A|)})$ fraction of equations are satisfiable.

For each vertex $v \in \Sigma_A \cup \Sigma_B$ and possible label $\ell$ on the Label Cover instance, we have a variable $x_{v,\ell}$ in the LIN instance. Let $n = |A||\Sigma_A| + |B||\Sigma_B| = \text{poly}(|A| + |B|)$ be the number of variables. Consider the following four kinds of equations. Recall that every arithmetic is performed over $\mathbf{F}_2$.

$$
\begin{array}{lll}
(1) & \sum_{\ell \in \Sigma_A} x_{v,\ell} = 1 & \forall v \in A \\
(2) & \sum_{\ell \in \Sigma_B} x_{v,\ell} = 1 & \forall v \in B \\
(3) & \sum_{r : \pi_{uv}(r) = \ell} x_{v,r} = x_{u,\ell} & \forall (u,v) \in E, \forall \ell \in \Sigma_B \\
(4) & x_{v,\ell} = 0 & \forall (v,\ell) \in A \times \Sigma_A
\end{array}
$$

In our final LINinstance, we treat (1), (2), and (3) as *hard constraints* that need to be always satisfied, and find $x$ that always satisfies all hard constraints and as many constraints in (4) as possible. Also note that in (4), we only consider vertices in $A$.

This is equivalent to the usual LINproblem with hard constraints by *folding*. Formally, let $V$ be the set of assignments that satisfy (1), (2), and (3). If $V$ is empty, we can conclude that the LABEL COVER instance is unsatisfiable. Otherwise, there exist $c \in \mathbb{N}$ and $c$ linearly independent vectors $y_0, \ldots, y_c \in \mathbf{F}_2^{(A \times \Sigma_A) \cup (B \times \Sigma_B)}$ such that $V = \{y_0 + \sum_{i=1}^{c} y_i z_i : z_1, \ldots, z_c \in \mathbf{F}_2\}$. This gives an one-to-one correspondence between $\mathbf{F}_2^c$ and $V$, so we can treat $z_1, \ldots, z_c$ as the variables of LINand write the fourth constraints $x_{v,\ell} = 0$ in terms of $z$, which gives an instance of LINwithout hard constraints.

**Completeness.** If the LABEL COVER instance is satisfiable, $x_{v,\ell} = 1$ if and only if $v$ is assigned with $\ell$ gives an assignment that satisfies (1), (2), and (3), and violates one equation in (4) for each $v \in A$.

**Soundness.** Let $x$ be an assignment that satisfies (1), (2), and (3). For $v \in A \cup B$, let $L_v := \{\ell : x_{v,\ell} = 1\}$. Since (1) and (2) require $\sum_\ell x_{v,\ell} = 1$ for every $v \in A \cup B$, $L_v$ is not empty for every $v$.

Consider the randomized strategy for LABEL COVER where each $v \in A \cup B$ is assigned with a uniform random label from $L_v$ independently. For $(u, v) \in E$ with $u \in A, v \in B$, by (3), $x_{v,\ell} = 1$ for some $\ell \in \Sigma_B$ implies that there exists $r \in \Sigma_A$ with $\pi_{uv}(r) = \ell$ such that $x_{u,r} = 1$. This implies $(u, v)$ is satisfied with probability at least $\frac{1}{|L_u|}$ by the randomized strategy. Then the expected fraction of the LABEL COVER constraints satisfied by the strategy is at least

$$\mathop{\mathbf{E}}_{u \in A}\left[\frac{1}{|L_u|}\right] \geqslant \frac{1}{\mathbf{E}_{u \in A}[|L_u|]}.$$

Therefore, if at most $\delta$ fraction of LABEL COVER constraints are simultaneously satisfiable, we can conclude that

$$\delta \geqslant \frac{1}{\mathbf{E}_{u \in A}[|L_u|]} \quad \Leftrightarrow \quad \mathop{\mathbf{E}}_{u \in A}[|L_u|] \geqslant \frac{1}{\delta}.$$

So in total, at least $\frac{1}{(\delta|\Sigma_A|)}$ fraction of equations are violated.

**Gap Amplification.** We have a hardness of LINover $\mathbf{F}_2$ where the completeness value is at least $1 - \frac{1}{|\Sigma_A|}$ and the soundness value is at most $1 - \frac{1}{(\delta|\Sigma_A|)}$. Consider a new system of linear equations where we sample $m$ linear equations independently, where each new equation randomly chooses $\delta \cdot |\Sigma_A|$ old equations and takes a random linear combination of them. In the completeness case, at least an $(1 - O(\delta))$ fraction of new equations can be satisfied by a good assignment to old equations.

In the soundness case, fix an assignment to $n$ possible variables. (There are $2^n$ of them.) It satisfies at most an $1 - \frac{1}{(\delta|\Sigma_A|)}$ fraction of old equations. Note that if a new equation chooses an old equation not satisfied by the assignment, it is satisfied with probability exactly $1/2$. Therefore, the expected number of new equations satisfied by this fixed assignment is at most

$$m \cdot \left( (1 - \frac{1}{(\delta|\Sigma_A|)})^{\delta \cdot |\Sigma_A|} + \frac{1}{2} \right) \leqslant m \cdot \left( \frac{1}{e} + \frac{1}{2} \right) \leqslant 0.87m.$$

For a given $c \in \mathbb{N}$, let $\delta = {}^{1}\!/\!_{\log^c n}$. By taking sufficiently large $m = O(n)$, we can apply the Chernoff and union bound to conclude that no assignment satisfies more than a $0.9$ fraction of new equations. So we reduce from LABEL COVER to GAP LIN$(1 - O(\delta), 0.9)$, which finishes the proof. ■

We remark that the sampling performed above is the only step in our reduction involving randomization.

## 4.3 Reduction to Linear Label Cover

In this section, we show for any $c > 0$, there is no polynomial time algorithm for GAP LINEAR LABEL COVER$(1 - \varepsilon, \varepsilon)$ with $\varepsilon = {}^{1}\!/\!_{(\log n)^c}$ unless **NP** $\subseteq$ **BPP**, proving Theorem 4.1.8.

The construction we employ is almost identical to that of Dinur and Harsha [DH13], except that the basic building blocks (robust PCP and decodable PCP) try to prove (almost) satisfiability of linear equations instead of standard quadratic equations. They are introduced in Sections 4.3.1 and 4.3.2.

After constructing the building blocks, the result of [DH13] is proved by iterative composition of them followed by technical steps including alphabet and degree reduction. Our main observation in this part is that each of the steps in the construction preserves *linearity* so that the final LABEL COVER instance produced also has a liear structure. We present them in Section 4.3.3 and Section 4.3.4. Finally, Section 4.3.5 shows how to combine all these steps to prove Theorem 4.1.8.

### 4.3.1 Robust PCPs

In this subsection, we define robust PCPs. For two strings $x, y$ of the same length $n$, let $\mathsf{agr}(x, y)$ denote the relative agreement of the strings $x, y$, defined as

$$\mathsf{agr}(x, y) := \Pr_{i \in [n]}[x_i = y_i]$$

If $S$ is a set of strings, $\mathsf{agr}(x, S)$ is defined as $\max_{y \in S}\{\mathsf{agr}(x, y)\}$.

**Definition 4.3.1 (Robust PCPs)** *For functions* $\mathsf{r}, \mathsf{q}, \mathsf{m}, \mathsf{a}, \mathsf{s} : \mathbb{N} \to \mathbb{N}$ *and* $c, \delta : \mathbb{N} \to [0, 1]$, *a verifier* $V$ *is a* robust probabilistically checkable proof (robust PCP) system *for a promise problem* $L = (L_{\text{YES}}, L_{\text{NO}})$ *with randomness complexity* $\mathsf{r}$, *query complexity* $\mathsf{q}$, *proof length* $\mathsf{m}$, *alphabet size* $\mathsf{a}$, *robust completeness* $c$, *and robust soundness error* $\delta$ *if* $V$ *is a probabilistic polynomial-time algorithm that behaves as follows: On input* $x$ *of length* $n$ *and oracle access to a proof string* $\pi \in \Sigma^{\mathsf{m}(n)}$ *over the (proof) alphabet* $\Sigma$ *where* $|\Sigma| = \mathsf{a}(n)$, $V$ *reads the input* $x$, *tosses at most* $\mathsf{r} = \mathsf{r}(n)$ *random coins, and generates a sequence of locations* $I = (i_1, \ldots, i_q) \in [\mathsf{m}]^{\mathsf{q}(n)}$ *and a predicate* $f : \Sigma^{\mathsf{q}} \to \{0, 1\}$, *which satisfy the following properties.*

**Robust Completeness.** *If $x \in L_{\text{YES}}$ then there exists $\pi$ such that*

$$\mathop{\mathbf{E}}_{(I,f)}\left[\mathsf{agr}(\pi_I, f^{-1}(1))\right] \geqslant c. \tag{4.3.1}$$

**Robust Soundness.** *If $x \in L_{\text{NO}}$ then for every $\pi$,*

$$\mathop{\mathbf{E}}_{(I,f)}\left[\mathsf{agr}(\pi_I, f^{-1}(1))\right] \leqslant \delta, \tag{4.3.2}$$

*where the distribution over $(I, f)$ is determined by $x$ and the random coins of $V$.*

*We say that $V$ is* linear *if $\Sigma = \mathbf{F}_2^b$ for some $b$ and for every $f$, the accepting sets of the predicate $f$, i.e., $f^{-1}(1)$, forms an affine subspace of $\Sigma^{\mathsf{q}} = \mathbf{F}_2^{b\mathsf{q}}$ over the field $\mathbf{F}_2$.*

Robust completeness and soundness must be contrasted with (regular) completeness and soundness of standard PCP verifiers in which the expression for completeness and soundness given in (4.3.1) and (4.3.2) respectively are replaced as follows:

$$\text{Completeness: } \Pr_{I,f}[f(\pi_I) = 1] \geqslant c,$$
$$\text{Soundness: } \Pr_{I,f}[f(\pi_I) = 1] \leqslant \delta.$$

In fact, this is the only difference between the above definition and the standard definition of a PCP system. The robust soundness states that not only does the local view violate the local predicate $f$, but in fact has very little agreement with any of the satisfying assignments of $f$ (and thus is a strengthening of standard robustness). Robust completeness on the other hand is a weakening of standard completeness.

Another crucial aspect of robust PCP is its equivalence to LABEL COVER.

Namely, existence of robust PCP for $L$ with parameters $\mathsf{r}, \mathsf{q}, \mathsf{m}, \mathsf{a}, \mathsf{s}, c, \delta$ is equivalent to existence of a reduction from $L$ to GAP LABEL COVER$(c, \delta)$ where $|A| = 2^{\mathsf{r}}, |B| = \mathsf{m}, |\Sigma_A| \leqslant \mathsf{a}^{\mathsf{q}}, |\Sigma_B| = \mathsf{a}$ and each $v \in A$ has degree $\mathsf{q}$. See Lemma 2.5 of [DH13]. Also note that the definition of linearity is equivalent in robust PCP and LABEL COVER.

**Theorem 4.3.2 (Robust PCP, Analogous to [DH13, Theorem 6.4])** *Constants $b_1$, $b_2 > 0$, $c_0 > 1$ exist such that for any $c > c_0$ and $\varepsilon = {}^{1}/_{\log^c n}$,* GAP LIN$(1 - \varepsilon, 0.9)$ *with $n$ variables has a linear robust verifier with robust completeness $1 - \varepsilon$, robust soundness error $\varepsilon$, query complexity $1/\varepsilon^{b_1}$, proof length poly$(n)$, randomness complexity $O(\log n)$, and proof alphabet size at most $1/\varepsilon^{b_2}$.*

*Equivalently, there is a (deterministic) polynomial time reduction from* GAP LIN $(1 - \varepsilon, 0.9)$ *to* GAP LINEAR LABEL COVER$(1 - \varepsilon, 0.9)$*, where the* LABEL COVER *instance has poly$(n)$ veritces, $|\Sigma_A| \leqslant \exp(1/\varepsilon^{b_1} \log(1/\varepsilon^{b_2}))$, $|\Sigma_B| \leqslant 1/\varepsilon^{b_2}$, and each $v \in A$ has degree $1/\varepsilon^{b_1}$.*

The proof of this theorem is identical to that of [DH13, Theorem 6.4] and omitted here. The only difference is that our starting point is GAP LIN$_{\mathbf{F}_q}(1-\varepsilon, \varepsilon)$ with $q, 1/\varepsilon = \log^{O(c)} n$ instead of standard quadratic equations when performing the low degree-extension and the sum-check protocol. The theorem follows by observing that all the operations are linear and hence the final predicate is also linear. The completeness of the robust PCP is dictated by the completeness value in Theorem 4.2.1.

Combining this reduction and randomized reduction from Theorem 4.2.1, we obtain the following theorem (which is a more formal version of Theorem 4.1.8).

**Theorem 4.3.3 (Hardness of Linear Label Cover)** *There exist constants $b_1, b_2 > 0$,*

$c_0 > 1$ *such that for any* $c > c_0$ *and* $\varepsilon = {}^{1}\!/\!_{\log^c n}$, *unless* $\mathbf{NP} \subseteq \mathbf{BPP}$, *there is no polynomial time algorithm for* GAP LINEAR LABEL COVER$(1-\varepsilon, \varepsilon)$ *where the* LABEL COVER *instance has poly*$(n)$ *veritces,* $|\Sigma_A| \leqslant \exp(1/\varepsilon^{b_1} \log(1/\varepsilon^{b_2}))$, $|\Sigma_B| \leqslant 1/\varepsilon^{b_2}$, *and each* $v \in A$ *has degree* $1/\varepsilon^{b_1}$.

### 4.3.2 Decodable PCPs

We now discuss the decodable PCP (dPCP), which differs from a PCP in that it has a decoder as opposed to a verifier. A *decoder* is similar to a verifier in that it checks whether a string is in the given language or not by probabilistically checking a small number of positions in the proof, but it is additionally supposed to return the $i$th position of the original string for given $i$.

For $\Sigma = \mathbf{F}_2^a$ for some $a \in \mathbb{N}$, let $LIN_\Sigma$ denote the problem of solving linear equations where an instance consists of $k$ variables that can have a value from $\Sigma$, and a system of linear equations $C$ on $k \cdot a$ variables over $\mathbf{F}_2$ canonically represented by the $k$ variables over $\Sigma$. It is equivalent to LINover $\mathbf{F}_2$ on $k \cdot a$ variables, except that we consider each block of $a$ variables as one variable that can take a value from $\Sigma$. We define a decoder for $LIN_\Sigma$ below.

**Definition 4.3.4 (Decoder for $LIN_\Sigma$)** *Let* $\Sigma = \mathbf{F}_2^a$ *and* $\sigma = \mathbf{F}_2^b$ *for some* $a$ *and* $b$. *A decoder for* $LIN_\Sigma$ *over a proof alphabet* $\sigma$ *with parameters* $\mathsf{m}, \mathsf{q}, \mathsf{r} : \mathbb{N} \to \mathbb{N}$ *is a probabilistic polynomial-time algorithm* $\mathcal{D}$. *It is given a system of linear equations* $C$ *on* $n$ *variables over* $\Sigma$, *and an index* $j \in [n]$ *as input, and oracle access to a proof* $\pi$ *of length* $\mathsf{m}(n)$ *over proof alphabet* $\sigma$. *It tosses* $\mathsf{r} = \mathsf{r}(n)$ *random coins and generates (1) a sequence of* $\mathsf{q} = \mathsf{q}(n)$ *locations* $I = (i_1, \ldots, i_{\mathsf{q}})$ *and (2) a (local decoding) function* $f : \sigma^{\mathsf{q}} \to \Sigma \cup \{\perp\}$. $\mathcal{D}$ *is called linear if for every* $f$, $P := f^{-1}(\Sigma)$ *is an affine space of*

$\sigma^{\mathsf{q}} = (\mathbf{F}_2^{\mathsf{q}b})$ *and* $f : P \to \Sigma$ *is an affine function over the base field* $\mathbf{F}_2$.

Now we define a dPCP for $LIN_\Sigma$. The dPCP in [DH13] is defined for CIRCUIT SAT, whereas ours is for $LIN_\Sigma$. Note that unlike in [DH13], the dPCP we will construct does not imply any computational hardness, because it only proves whether the given system of linear equations is perfectly satisfiable or not, which is a computationally easy problem. The key point is it proves the system is satisfiable using a proof which is in some sense "locally decodable". The dPCP will then be composed with the previous linear robust PCP, which is a system of linear equations with *imperfect completeness*, to reduce the query complexity.

**Definition 4.3.5 (Decodable PCPs for $LIN_\Sigma$)** *For functions $\delta : \mathbb{N} \to [0, 1]$ and $\mathsf{L} : \mathbb{N} \to \mathbb{N}$, we say that a PCP decoder $\mathcal{D}$ is a* decodable probabilistically checkable proof (dPCP) system *for $LIN_\Sigma$ with perfect completeness, soundness $\delta$ and list size $\mathsf{L}$ if the following completeness and soundness properties hold for every system of linear equations $C$ on $n$ variables over $\Sigma$.*

**Completeness.** *For any $y \in \Sigma^n$ that satisfies every equation in $C$, there exists a proof $\pi \in \sigma^{\mathsf{m}}$, also called a decodable PCP, such that*

$$\Pr_{j,I,f}[f(\pi_I) = y_j] = 1,$$

*where $j \in [n]$ is chosen uniformly at random and $I, f$ are distributed according to $C, j$, and the verifier's random coins.*

**Soundness.** *For any $\pi \in \sigma^{\mathsf{m}}$, there is a list of $0 \leqslant \ell \leqslant \mathsf{L}$ strings $y^1, \ldots, y^\ell$, where each $y^i$ satisfies all equations in $C$, such that*

$$\Pr_{j,I,f}[f(\pi_I) \notin \{\perp, y_j^1, \ldots, y_j^\ell\}] \leqslant \delta.$$

**Robust soundness.** *We say that $\mathcal{D}$ is a* robust dPCP system *for $LIN_\Sigma$ with robust soundness error $\delta$, if the soundness criterion above can be strengthened to the following robust soundness criterion,*

$$\mathop{\mathbf{E}}_{j,I,f}[\mathsf{agr}(\pi_I, \mathrm{BAD}(f))] \leqslant \delta,$$

*where*

$$\mathrm{BAD}(f) := \{w \in \sigma^{\mathsf{q}} : f(w) \notin \{\perp, y_j^1, \ldots, y_j^\ell\}\}.$$

The dPCP result we use is the following.

**Theorem 4.3.6 (dPCP, Analogous to [DH13, Theorem 6.5])** *There exist constants $\alpha, \gamma > 0$ such that for every $\delta \geqslant n^{-\alpha}$ and input alphabet size $\Sigma$ of size at most $n^\gamma$, $LIN_\Sigma$ has a linear robust decodable PCP system with perfect completeness, robust soundness error $\delta > 0$ and list size $\mathsf{ConjSAT\text{-}LP} \leqslant 2/\delta$, query complexity $n^{1/8}$, proof alphabet $\sigma$ of size $n^\gamma$, proof length poly$(n)$, and randomness complexity $O(\log n)$.*

The proof of this theorem is identical to that of [DH13, Theorem 6.5], except that the initial starting point is $LIN_\Sigma$ instead of CIRCUIT SAT$_\Sigma$. Since the starting point is linear and all transformations are linear, the final object is also linear. The perfect completeness is also maintained. As mentioned before, the dPCP constructed here does not imply any computational hardness unlike in [DH13].

### 4.3.3 Composition

After having building blocks, Dinur and Harsha [DH13] show how to compose those blocks iteratively to reduce the query complexity and the alphabet size. Each composition involves several other operations including alphabet and degree reductions. While the soundness analyses for them are already proved in [DH13], we show that all of their operations preserve linearity and robust completeness.

**Efficient Composition ([DH13, Theorem 4.2]).** In the composition, given a regular robust linear PCP verifier $V$ and a robust linear PCP decoder $\mathcal{D}$, the composed verifier $V'$ expects a decodable PCP for each constraint of $V$. Recall that the linearity of $V$ is equivalent to the fact that each constraint of $V$ is a system of linear equations over $\mathbf{F}_2$, which is exactly what $\mathcal{D}$ expects. An informal description of the composed verifier is as follows:

1. Randomly choose a location $i$ of the proof for $V$. Let $C_1, \ldots, C_D$ be the constraints of $V$ containing the location.

2. Using a $(\varepsilon, \varepsilon^2)$-sampler $([D], [D], E)$ and a random $s \in [D]$, choose a subset $S \subseteq \{1, \ldots, D\}$ and run the inner PCP decoder $\mathcal{D}$ for each $C_j$ with $j \in S$ to decode the $i$th symbol in the original proof.

3. Accept if all the values returned by the PCP decoders are the same.

For the second step above, we use $(\varepsilon, \varepsilon^2)$-samplers given in [Gol11]. Theorem 4.2 of [DH13] shows the soundness of the composed verifier $V'$, yielding Table 4.1 below (Table 4.2 in [DH13]).

We check this composition preserves robust completeness and linearity.

|  | $V$ | $\mathcal{D}$ | $V'$ |
|---|---|---|---|
| proof alphabet | $\Sigma$ | $\sigma$ | $\sigma$ |
| randomness complexity | $\mathsf{R}$ | $\mathsf{r}$ | $\log M + \mathsf{r} + \log D$ |
| query complexity | $Q$ | $q$ | $4/\varepsilon^4 \cdot q$ |
| proof degree | $D$ | $d$ | $d$ |
| proof length | $M$ | $\mathsf{m}$ | $2^{\mathsf{R}} \cdot \mathsf{m}$ |
| robust soundness error | $\Delta$ | $\delta$ | $\Delta\mathsf{L} + 4\mathsf{L}\varepsilon + \delta$ |
| list size | - | $\mathsf{L}$ | - |

Table 4.1: Parameters for Composition.

- Linearity: Linearity (over $\mathbf{F}_2$) is preserved if both $V$ and $\mathcal{D}$ are linear, since the only additional check we perform is to check whether the returned values are equal.

- Robust completeness: Suppose there exists a proof $\Pi$ for $V$ that achieves the robust completeness of at least $1 - \xi$. Recall that the composed verifier expects, for each constraint of the outer PCP, a satisfying assignment encoded by the inner dPCP. The proof for the composed verifier is the concatenation of all these encodings. Consider the proof to the composed verifier constructed by the honest encoding of the assignment that achieves the robust completeness for the outer PCP verifier. We will show that this proof achieves robust completeness $1 - \xi$.

  Let $i$ be a proof location in the outer PCP and $C_1, \ldots, C_D$ be the constraints involving $i$. Furthermore, let $\xi_i$ be the fraction of these constraints violated by the proof. Since $\Pi$ is at least $(1-\xi)$-robustly complete, we have $\mathbf{E}_i[\xi] \leqslant \xi$. For each sample $s$ chosen in the sampler, let $\xi_{i,s}$ be the fraction of constraints in $S$ (chosen by sampler) that are violated. By regularity of sampler, we have $\mathbf{E}_s[\xi_{i,s}] \leqslant \xi_i$.

  A local view of the composed verifier (corresponding to $i, s$ and the inner dPCP randomness) comprises of the concatenation of the local views of the dPCP encodings corresponding to the constraints in $S$. Since the the inner dPCP has perfect completeness we have the following. Whenever the constraint is satisfied, the corresponding inner dPCP's encodings satisfies all constraints while we have no guarantee when the constraint is not satisfied. Since for each $(i, s)$, the fraction of violated constraints is $\xi_{i,s}$, we have that at least $(1 - \xi_{i,s})$-fraction of the local inner views corresponding to $(i, s)$ are satisfying and furthermore they all decode to the same $\Pi(i)$. Hence, the local view of the composed verifier corresponding to $(i, s)$ is at least $(1 - \xi_{i,s})$-close to a satisfying view. Hence, the robust completeness of this honest proof is at least $\mathbf{E}_{i,s}[1 - \xi_{i,s}] \geqslant 1 - \xi$.

### 4.3.4 Label Cover Operations

After the composition, the alphabet reduction step is applied to ensure that the alphabet size is polynomial in the query complexity and the inverse of the soundness. Also, since the basic robust PCP given in Theorem 4.3.2 is not necessarily regular, we also need to show how to make the initial robust PCP regular. This subsection introduces various such operations and explains why they preserve robust completeness and linearity.

**Degree Reduction ([DH13, Theorem 5.1])** If we are given an instance of Label Cover $G = (A, B, E)$, the degree reduction makes the instance right regular by appropriately duplicating right vertices and each edge exactly the same number of times. Theorem 5.1 of [DH13] ensures that by increasing robust soundness by $4\mu$ additively, we can ensure that the right degree is $4/\mu^4$ for all right vertices. We check that this operation preserves linearity and robust completeness.

- Linearity: Linearity is obviously preserved, because there is no change in the constraint.
- Robust completeness: Since each edge is duplicated the same number of times, robust completeness does not decrease.

**Alphabet Reduction ([DH13, Theorem 5.5])** If we are given an instance of LABEL COVER $G = (A, B, E)$ where $\Sigma_A$ and $\Sigma_B$ are the alphabet set of the left (bigger) side and the right (smaller) side respectively, the alphabet reduction replaces $\Sigma_B$ by a smaller set $\sigma$ by finding a suitable linear code $C : \Sigma_B \to \sigma^k$ and replacing each vertex $b \in B$ by $k$ vertices $b_1, \dots, b_k$. Then assigning $x \in \Sigma_B$ to $b$ corresponds to assigning $(C(x))_i$ to $b_1, \dots, b_i$. Theorem 5.5 of [DH13] ensures that if $C$ has a relative distance $1 - \eta^3$, this operation increases robust soundness by at most $3\eta$ additively. We check that this operation preserves linearity and robust completeness.

- Linearity: Linearity over $\mathbf{F}_2$ is preserved if the code $C : \Sigma_B \to \sigma^k$ is linear with $\sigma = \mathbf{F}_{2^a}$ as the base field for some $a \in \mathbb{N}$. The code used in Remark 5.4 of [DH13] is already linear.
- Robust completeness: If an edge $(a, b)$ of the original LABEL COVER instance is preserved and the new instance follows the honest encoding, all $k$ edges of the new instance corresponding to $(a, b)$ will be satisfied. Therefore, robust completeness cannot decrease.

**Flip Sides ([DH13, Section 5.3]).** Given an instance of LABEL COVER $G = (A, B, E)$ where each right vertex $b \in B$ has degree $d$, the flip side is achieved by flipping $A$ and $B$, and assigning each $v \in B$ a label from $\Sigma_A^d$, which is supposed to denote the assignments to its neighbors in the original instance. If $v \in B$ has $u_1, \ldots, u_d \in A$ as neighbors, $(v, u_i)$ in the new instance is satisfied (i) if the label $(a_1, \ldots, a_d) \in \Sigma_A^d$ for $v$ has $b \in \Sigma_B$ such that the label pair $(a_i, b)$ satisfies the edge $(u_i, v)$ in the old instance, and (ii) if $a_i$ is equal to the label assigned to $u_i$. This obviously does not change the robust soundness. We check that it also preserves linearity and robust completeness.

- Linearity: Linearity is preserved, because for each $v \in B$, the set of $(a_1, ..., a_d)$ satisfying (i) above is an affine subspace of $(\Sigma_A)^d$, and the new constraint is merely a projection.

- Robust completeness: Cannot decrease since if $v \in B$ was assigned $b \in \Sigma_B$ in the original instance, it can be assigned $(a_1, \ldots, a_d) \in \Sigma_A$ such that (i) $\pi_{(u_i,v)}(a_i) = b$, and (ii) $a_i$ was assigned to $u_i$ if $(u_i, v)$ was satisfied in the original instance.

We use a combination of the above 3 operations to get a regular LABEL COVER instance, as shown in Table 4.2.

Given an $\varepsilon > 0$, by using $(O(\varepsilon), O(\varepsilon^2))$-samplers in the composition and doing the above operations with $\eta = O(\varepsilon)$, $d = O(1/\varepsilon^4)$, distance $1 - O(\varepsilon^3)$, $|\sigma| = O(1/\varepsilon^6), k = O(1/\varepsilon^6) \cdot |\Sigma'| \leqslant O(1/\varepsilon^6) \cdot q|\Sigma|$, we can deduce the following lemma.

**Lemma 4.3.7 ([DH13, Lemma 5.7])** *For all $\varepsilon : \mathbb{N} \to [0, 1]$, suppose $L$ has a robust linear PCP verifier $V$ with randomness complexity $r$, query complexity $q$, proof length*

| LABEL COVER<br>(Robust PCPs) | $I$ | Degree<br>Red. ($\to d$) | Flip | Degree<br>Red. ($\to d$) | Alphabet<br>Red. ($\to \sigma$) |
|---|---|---|---|---|---|
| # left vertices<br>(randomness) | $n$ | $n$ | $mD_B$ | $mD_B$ | $mD_B$ |
| # right vertices<br>(proof length) | $m$ | $mD_B$ | $n$ | $nD_Ad$ | $nD_Adk$ |
| left degree<br>(query complexity) | $D_A^*$ | $dD_A^*$ | $d$ | $d^2$ | $d^2k$ |
| right degree<br>(proof degree) | $D_B^*$ | $d$ | $D_Ad^*$ | $d$ | $d$ |
| left alphabet<br>(# accepting conf.) | $\Sigma_A$ | $\Sigma_A$ | $\Sigma_A^d$ | $\Sigma_A^d$ | $\Sigma_A^d$ |
| right alphabet<br>(proof alphabet) | $\Sigma_B$ | $\Sigma_B$ | $\Sigma_A$ | $\Sigma_A$ | $\sigma$ |
| soundness error<br>(rob. soundness error) | $\delta$ | $\delta + 4\mu$ | $\delta + 4\mu$ | $\delta + 8\mu$ | $\delta + 8\mu + 3\eta$ |
| robust completeness<br>(rob. completeness) | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ |

Table 4.2: Sequence of steps to regularize the LABEL COVER instance. * denotes irregular instances where the number denotes the average degree.

*$m$, average proof degree $D_B$, robust completeness $c$, robust soundness error $\delta$ over a proof alphabet $\Sigma$. Then $L$ has a regular reduced linear robust PCP verifier, which we shall denote by $\mathsf{regular}_\varepsilon(V)$ with*

- *randomness complexity* $\log m + \log D_B$,
- *query complexity $O(q \log |\Sigma| / \varepsilon^{14})$,*
- *proof length $O(q^2 2^r \log |\Sigma| / \varepsilon^{10})$,*
- *proof degree $O(1/\varepsilon^4)$,*
- *proof alphabet $\sigma$ of size at most $O(1/\varepsilon^6)$,*
- *robust completeness $c$,*
- *and robust soundness $\delta + \varepsilon$.*

### 4.3.5 Putting things together

Finally we prove Theorem 4.1.8 on the hardness of LINEAR LABEL COVER. Let $c > 0$ be an arbitrary constant. Let $\mathcal{D}$ be the PCP decoder from Theorem 4.3.6 and $\mathcal{V}$ be the robust PCP from Theorem 4.3.2 with robust completeness $1 - \delta$ with $\delta = \log^c n$, robust soundness error $\varepsilon = {}^1\!/\!_{\log^{c_0} n}$ for some $c_0 > 1$, query complexity $1/\varepsilon^{O(1)}$, randomness complexity $O(\log n)$ and proof length poly$(n)$.

**Lemma 4.3.8 ([DH13, Lemma 6.6])** *Let $\mathcal{D}$, $\mathcal{V}$, $\varepsilon, \delta$ be as defined above and set $\varepsilon_i = (\varepsilon)^{1/3^i}$. There exist constants $c_0$, $c_1$, $c_3 > 0$ such that for every $i \geqslant 0$ as long as $\varepsilon_i < c_0$, the following holds.* GAP LIN$(1-\delta, 0.9)$ *has a regular linear robust PCP verifier $V_i$ with query complexity $1/\varepsilon_i^{c_1}$, robust completeness $1-\delta$, robust soundness error $2\varepsilon_i$, proof alphabet $\Sigma_i$ of size $c_3/{\varepsilon_i}^6$, randomness complexity $O(\log n)$ and proof length poly$(n)$.*

**Proof:** The proof is similar to [DH13], and is a sequence of compositions. We start with the regularized robust verifier given by applying the sequence of steps given in Table 4.2 to the robust PCP verifier given in Theorem 4.3.2. In each subsequent step, we compose the robust verifier obtained in the previous step with a dPCP, and apply the alphabet reduction (Theorem 5.5 of [DH13]) to reduce the size of the alphabet to $c_3/\varepsilon_{i+1}^6$. All the parameters remain the same as in [DH13], and we only need to focus on the two additional properties we need, linearity and robust completeness.

Recall that a PCP with robust completeness $1-\delta$, when composed with a dPCP with perfect completeness, yields a composed PCP with robust completeness $1 - \delta$. In each step the inner PCP decoder has perfect completeness, therefore the robust completeness of the composed PCP is preserved. Recall that the alphabet reduction step also doesn't affect the perfect completeness.

Linearity is also preserved because all basic components are linear and all steps (e.g., composition, alphabet reduction, and regularization) preserve linearity as previously discussed. ■

The above lemma shows that we can iteratively reduce the query complexity until some absolute constant while maintaining the soundness and the alphabet size polynomial in the query complexity.(And the total size of the instance always remains polynomial in $n$.) Only a constant number of iterations is needed until (proof alphabet size)$^{\text{(query complexity)}}$, an upper bound on the size of alphabet in the equivalent LABEL COVER instance, becomes polynomial in $n$. This proves our main Theorem 4.1.8 for LINEAR LABEL COVER.

**Proof:** [Proof of Theorem 4.1.8] Set $i$ from Theorem 4.3.8 so that

$$\begin{aligned}\text{(proof alphabet size)}^{\text{(query complexity)}} &= (c_3/\varepsilon_i^6)^{1/\varepsilon_i^{c_1}}\\ &= \exp\left(\frac{1}{\varepsilon_i^{c_1}} \cdot \log\left(\frac{c_3}{\varepsilon_i^6}\right)\right)\\ &\leqslant \text{poly}(n).\end{aligned}$$

This ensures that $\varepsilon_i = {}^1\!/\!_{\log^{c_4} n}$ for some $c_4 > 0$. Using the equivalence between LABEL COVER and robust PCP, we have a hardness of LABEL COVER where the number of vertices and the size of label are bounded by $\text{poly}(n)$, and the completeness is at least $1 - {}^1\!/\!_{\log^c n}$, the soundness is ${}^1\!/\!_{\log^{c_4} n}$. Applying the parallel repetition of [DS14] $O(c/c_4)$ times to reduce the soundness to ${}^1\!/\!_{\log^c n}$ finishes the proof. ■

## 4.4 Reduction from Linear Label Cover to 3LIN

In this section, we prove our main Theorem 4.1.6 for 3-LIN. Recall that for any constant $c > 0$, Theorem 4.3.3 shows a randomized polynomial reduction from 3-SAT to GAP LINEAR LABEL COVER$(1 - \log^c n, \log^c n)$ where the number of vertices as well as the number of labels are bounded by a polynomial. Therefore, the following theorem finishes the proof of Theorem 4.1.6. The main idea is to use Hadamard codes instead of long codes using the fact that the LABEL COVER instance is linear. A similar argument was used in [Kho01].

**Lemma 4.4.1** *There exists a reduction from* GAP LINEAR LABEL COVER$(1 - \delta, s)$ *to* GAP 3-LIN$(1 - \delta, 1/2 + \sqrt{s}/2)$ *that runs in polynomial time, where the size of the* 3-LIN *instance is polynomial in the number of vertices and the size of label in the* LABEL COVER *instance.*

**Proof:** Let $G = (A, B, E)$, $\Sigma_A$, $\Sigma_B$, and $\{\pi_e\}_{e \in E}$ be an instance of GAP-LINEAR LABEL COVER $(1 - \delta, s)$. Moreover, since the label cover is linear, let the labels to left hand side vertices come from $\mathbb{F}_2^\ell$ and the right hand side vertices from $\mathbb{F}_2^r$, and the mapping on each edge is an affine mapping. Our reduction is described by the following test.

**Test**

- Consider an edge $(u, v)$. The labels $x \in \mathbb{F}_2^\ell, y \in \mathbb{F}_2^r$ corresponding to the vertices have to satisfy $x = Ay + b$.
- From the proof, we randomly sample the Hadamard code of $x$ at location $\alpha$, and that of $y$ at locations $\beta$ and $\beta + \gamma$, where $\gamma = A^T \cdot \alpha$.

- Check if $\langle \alpha, x\rangle + \langle \beta, y\rangle + \langle \beta + \gamma, y\rangle = \langle \alpha, b\rangle$

**Completeness.** In the completeness case, if the labels $x, y$ satisfy the edge in the LINEAR LABEL COVER, then we can see that the test will pass.

$$\begin{aligned}
&\langle \alpha, x\rangle + \langle \beta, y\rangle + \langle \beta + \gamma, y\rangle \\
=&\langle \alpha, Ay\rangle + \langle \alpha, b\rangle + \langle \beta, y\rangle + \langle \beta + \gamma, y\rangle \\
=&\langle \alpha, Ay\rangle + \langle \alpha, b\rangle + \langle A^T\alpha, y\rangle \\
=&\langle \alpha, b\rangle
\end{aligned}$$

Therefore, if $1-\delta$ edges are satisfiable in the linear LABEL COVER, at least $1-\delta$ fraction of 3LINconstraints are satisfied.

**Soundness.** Consider the case where at most $s$ fraction of edges can be satisfied for any labeling in the LINEAR LABEL COVER. Let the Hadamard code encoding function for the left vertices be $L$ and right vertices be $R$. Consider their Fourier transforms,

$$L(\alpha) = \sum_x \hat{L}(x)\chi_x(\alpha)$$

$$R(\beta) = \sum_y \hat{R}(y)\chi_y(\beta)$$

Let's fix an edge, and analyze the probability that the test will accept. We switch to a -1,+1 notation for convenience.

$$\begin{aligned}\Pr[\text{Test accepts}] &= \Pr_{\alpha,\beta}[\langle \alpha, x\rangle + \langle \beta, y\rangle + \langle \beta + A^T\alpha, y\rangle + \langle \alpha, b\rangle = 0] \\ &= \Pr_{\alpha,\beta}[(-1)^{\langle \alpha, x\rangle + \langle \beta, y\rangle + \langle \beta + A^T\alpha, y\rangle + \langle \alpha, b\rangle} = 1] \\ &= \frac{1 + \mathbf{E}_{\alpha,\beta}\left[L(\alpha)R(\beta)R(\beta + A^T\alpha)(-1)^{\langle \alpha, b\rangle}\right]}{2}\end{aligned}$$

Consider the expectation on the right hand side of the above equation.

$$\begin{aligned}&\mathop{\mathbf{E}}_{\alpha,\beta}\left[L(\alpha)R(\beta)R(\beta + A^T\alpha)(-1)^{\langle \alpha, b\rangle}\right] \qquad (4.4.1)\\ &\leqslant \sum_{x,y} \hat{L}(x)\hat{R}(y)^2 \mathop{\mathbf{E}}_{\alpha,\beta}\left[\chi_x(\alpha)\chi_y(\beta)\chi_z(\beta + A^T\alpha)(-1)^{\langle \alpha, b\rangle}\right] \\ &\leqslant \sum_{x,y,x=Ay+b} \hat{L}(x)\hat{R}(y)^2 \\ &\leqslant \sqrt{\sum_{x,y,,x=Ay+b} \hat{R}(y)^2}\sqrt{\sum_{x,y,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2}\end{aligned}$$

In the above equation, the first term is bounded by 1, and therefore,

$$(4.4.1) \leqslant \sqrt{\sum_{x,y,,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2}$$

Consider a random assignment where a left vertex gets a label $x$ with probability $\hat{L}(x)^2$ and a right vertex gets a label $y$ with probability $\hat{R}(y)^2$. The probability that such a random assignment would satisfy the edge, and therefore the expected fraction of edges satisfied, is exactly

$$\sum_{x,y,x=Ay+b} \hat{L}(x)^2 \hat{R}(y)^2$$

If at most $s$ fraction of edges can be satisfied by any assignment, then

$$s \geqslant \sum_{x,y,x=Ay+b} \hat{L}(x)^2 \hat{R}(y)^2 \geqslant (2 \cdot \Pr[\text{Test accepts}] - 1)^2$$

or

$$\Pr[\text{Test accepts}] \leqslant \frac{1}{2} + \frac{\sqrt{s}}{2}$$

Therefore, the expected fraction of 3LIN constraints satisfied is at most $\frac{1}{2} + \frac{\sqrt{s}}{2}$.

■

# Chapter 5

# Dictatorship Test with perfect completeness

## 5.1 Introduction

Boolean functions are the most basic objects in the field of theoretical computer science. Studying different properties of Boolean functions has found applications in many areas including hardness of approximation, communication complexity, circuit complexity etc. In this paper, we are interested in studying Boolean functions from a property testing point of view.

In *property testing*, one has given access to a function $f : \{0,1\}^n \to \{0,1\}$ and the task is to decide if a given function has a particular property or whether it is *far* from it. One natural notion of farness is what fraction of $f$'s output we need to change so that the modified function has the required property. A verifier can have an access to random bits. This task of property testing seems trivial if we do not have restrictions on how many queries one can make and also on the computation. One of the main questions in this area is can we still decide if $f$ is very far from having the property by looking at a very few locations with high probability.

There are few different parameters which are of interest while designing such tests including the number of random bits used, the number of locations queried, the amount of computation the verifier is allowed to do etc. The test can either be *adaptive* or *non-adaptive*. In an adaptive test, the verifier is allowed to query a function at a few locations and based on the answers that it gets, the verifier can decide the next locations to query whereas a non-adaptive verifier queries the function in one shot and once the answers are received makes a decision whether the function has the given property. In terms of how good the prediction is we want the test to satisfy the following two properties:

- **Completeness:** If a given function has the property then the test should accept with high probability

- **Soundness:** If the function is far from the property then the test should accept with very tiny probability.

A test is said to have *perfect completeness* if in the completeness case the test always accepts. A test with *imperfect completeness* (or almost perfect completeness) accepts a dictator function with probability arbitrarily close to $1$. Let us define the soundness parameter of the test as how small we can make the acceptance probability in the soundness case.

A function is called a *dictator* if it depends on exactly one variable i.e $f(x_1, x_2, \ldots, x_n) = x_i$ for some $i \in [n]$. In this work, we are interested in a non-adaptive test with perfect completeness which decides whether a given function is a dictator or far from it. This was first studied in [BGS98, PRS02] under the name of Dictatorship test and Long Code test. Apart from a natural property, dictatorship test has been used extensively in the construction of probabilistically checkable proofs (PCPs) and hardness of approximation.

An instance of a *Label Cover* is a bipartite graph $G((A, B), E)$ where each edge $e \in E$ is labeled by a projection constraint $\pi_e : [L] \to [R]$. The goal is to assign labels from $[L]$ and $[R]$ to vertices in $A$ and $B$ respectivels so that the number of edge constraints satisfied is maximized. Let $\texttt{GapLC}(1, \varepsilon)$ is a promise gap problem where the task is to distinguish between the case when all the edges can be satisfied and at most $\varepsilon$ fraction of edges are satisfied by any assignment. As a consequence of the PCP Theorem [ALM+98, AS98] and the Parallel Repetition Theorem[Raz98], $\texttt{GapLC}(1, \varepsilon)$ is NP-hard for any constant $\varepsilon > 0$. In [Hås01], Håstad used various dictatorship tests along with the hardness of Label Cover to prove optimal inapproximability results for many constraint satisfaction problems. Since then dictatorship test has been central in proving hardness of approximation.

A dictatorship test with $k$ queries and $P$ as an accepting predicate is usually useful in showing hardness of approximating Max-$P$ problem. Although this is true for many CSPs, there is no black-box reduction from such dictatorship test to getting inapproximability result. One of the main obstacles in converting dictatorship test to NP-hardness result is that the constraints in Label Cover are $d$-to-$1$ where the the parameter $d$ depends on $\varepsilon$ in $\texttt{GapLC}(1, \varepsilon)$. To remedy this, Khot in [Kho02] conjectured that a Label Cover where the constraints are $1$-to-$1$, called *Unique Games*, is also hard to approximate within any constant. More specifically, Khot conjectured that $\texttt{GapUG}(1 - \varepsilon, \varepsilon)$, an analogous promise problem for Unique Games, is NP-hard for any constant $\varepsilon > 0$. One of the significance of this conjecture is that many dictatorship tests can be composed easily with $\texttt{GapUG}(1-\varepsilon, \varepsilon)$ to get inapproximability results. However, since the Unique Games problem lacks perfect completeness it cannot be used to show hardness of approximating *satisfying* instances.

From the PCP point of view, in order to get $k$-bit PCP with perfect completeness, the first step is to analyze $k$-query dictatorship test with perfect completeness. For its application to construction PCPs there are two important things we need to study about the dictatorship test. First one is how to compose the dictatorship test with the known PCPs and second is how sound we can make the dictatorship test. In this work, we make a progress in understanding the answer to the later question. To make a remark on the first question, there is a dictatorship test with perfect completeness and soundness $\frac{2^{\tilde{O}(k^{1/3})}}{2^k}$ and also a way to compose it with $\texttt{GapLC}(1, \varepsilon)$ to get a $k$-bit PCP with perfect completeness and the same soundness that of the dictatorship test. This was done in [Hua13] and is currently the best know $k$-bit non-adaptive PCP with perfect completeness.

**Distance from a dictator function:** There are multiple notions of closeness to a dictator function. One natural definition is the minimum fraction of values we need to change such that the function becomes a dictator. There are other relaxed notions such as how close the function is to *juntas* - functions that depend on constantly many variables. Since our main motivation is the use of dictatorship test in the construction of PCP, we can work with even more relaxed notion which we describe next: For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ an influence of $i^{th}$ variable is the probability that for a random input $x \in \{0,1\}^n$ flipping the

$i^{th}$ coordinate flips the value of the function. Note that a dictator function has a variable whose influence is $1$. The influence of $i^{th}$ variable can be expressed in terms of the fourier coefficients of $f$ as $\texttt{inf}_i[f] = \sum_{S \subseteq [n] | i \in S} \hat{f}(S)^2$. Using this, a degree $d$ influence of $f$ is $\texttt{inf}_i^{\leqslant d}[f] = \sum_{S \subseteq [n] | i \in S, |S| \leqslant d} \hat{f}(S)^2$. We say that $f$ is far from any dictator if for a constant $d$ all its degree $d$ influences are upper bounded by some small constant.

In this paper, we investigate the trade-off between the number of queries and the soundness parameter of a dictatorship test with perfect completeness w.r.t to the above defined distance to a dictator function. A random function is far from any dictator but still it passes any (non-trivial) $k$-query test with probability at least $1/2^k$. Thus, we cannot expect the test to have soundness parameter less than $1/2^k$. The main theorem in this paper is to show there exists a dictatorship test with perfect completeness and soundness at most $\frac{2k+1}{2^k}$.

**Theorem 5.1.1** *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, for every $k$ of the form $2^m - 1$ for any $m > 2$, there is a $k$ query dictatorship test with perfect completeness and soundness $\frac{2k+1}{2^k}$.*

Our theorem improves on the result of Tamaki-Yoshida[TY15], which had a soundness of $\frac{2k+3}{2^k}$.

**Remark 5.1.2** *Tamaki-Yoshida [TY15] studied a $k$ functions test where if a given set of $k$ functions are all the same dictator then the test accepts with probability $1$. They use low degree* cross influence *(Definition 2.4 in [TY15]) as a criteria to decide closeness to a dictator function. Our whole analysis also goes through under the same setting as that of [TY15], but we stick to single function version for a cleaner presentation.*

### 5.1.1 Previous Work

The notion of Dictatorship Test was introduced by Bellare et al. [BGS98] in the context of Probabilistically Checkable Proofs and also studied by Parnas et al. [PRS02]. As our focus is on non-adpative test, for an adaptive $k$-bit dictatorship test, we refer interested readers to [ST09, HW03, HK05, EH08]. Throughout this section, we use $k$ to denote the number of queries and $\varepsilon > 0$ an arbitrary small constant.

Getting the soundness parameter for a specific values of $k$ had been studied earlier. For instance, for $k = 3$ Håstad [Hås01] gave a 3-bit PCP with completeness $1 - \varepsilon$ and soundness $1/2 + \varepsilon$. It was earlier shown by Zwick [Zwi97] that any 3-bit dictator test with perfect completeness must have soundness at at least $5/8$. For a 3-bit dictatorship test with perfect completeness, Khot-Saket [KS06] acheived a soundness parameter $20/27$ and they were also able to compose their test with Label Cover towards getting 3-bit PCP with similar completeness and soundness parameters. The dictatorship test of Khot-Saket [KS06] was later improved by O'Donnell-Wu [OW09a] to the optimal value of $5/8$. The dictatorship test of O'Donnell-Wu [OW09a] was used in O'Donnell-Wu [OW09b] to get a conditional (based on Khot's $d$-to-1 conjecture) 3-bit PCP with perfect completeness and soundness $5/8$ which was later made unconditional by Håstad [Hås14].

For a general $k$, Samorodensky and Trevisan [ST00] constructed a $k$-bit PCP with imperfect completeness and soundness $2^{2\sqrt{k}}/2^k$. This was improved later by Engebretsen and Holmerin [EH08] to $2^{\sqrt{2k}}/2^k$ and by Håstad-Khot [HK05] to $2^{4\sqrt{k}}/2^k$ with perfect completeness. To break the $2^{O(\sqrt{k})}/2^k$ barrier, Samorodensky and Trevisan [ST09] introduced the relaxed notion of soundness (based on the low degree influences) and gave a dictatorship test (called Hypergraph dictatorship test) with almost perfect completeness and soundness $2k/2^k$ for every $k$ and also $(k+1)/2^k$ for infinitely many $k$. They combined this test with Khot's Unique Games Conjecture [Kho02] to get a conditional $k$-bit PCP with similar completeness and soundness guarantees. This result was improved by Austrin-Mossel [AM09] and they achieved $k + o(k)/2^k$ soundness.

For any $k$-bit CSP for which there is an instance with an integrality gap of $c/s$ for a certain SDP, using a result of Raghavendra [Rag08] one can get a dictatorship test with completeness $c - \varepsilon$ and soundness $s + \varepsilon$. Getting the explicit values of $c$ and $s$ for a given value of $k$ is not clear from this result and also it cannot be used to get a dictatorship test with perfect completeness. Similarly, using the characterization of strong approximation restance of Khot et. al [KTW14] one can get a dictatorship test but it also lacks peferct completeness. Recently, Chan [Cha13] significantly improved the parameters for a $k$-bit PCP which achieves soundness $2k/2^k$ albeit losing perfect completeness. Later Huang [Hua13] gave a $k$-bit PCP with perfect completeness and soundness $2^{\tilde{O}(k^{1/3})}/2^k$.

As noted earlier, the previously best known result for a $k$-bit dictatorship test with perfect completeness is by Tamaki-Yoshida [TY15]. They gave a test with soundness $\frac{2k+3}{2^k}$ for infinitely many $k$.

## 5.2 Proof Overview

Let $f : \{-1,+1\}^n \to \{-1,+1\}$ be a given balanced Boolean function [1]. Any non-adaptive $k$-query dictatorship test queries the function $f$ at $k$ locations and receives $k$ bits which are the function output on these queries inputs. The verifier

[1] Here we switch from $0/1$ to $+1/-1$ for convenience. With this notation switch, balanced function means $\mathbf{E}[f(\boldsymbol{x})] = 0$

then applies some predicate, let's call it $\mathcal{P} : \{0,1\}^k \to \{0,1\}$, to the received bits and based on the outcome decides whether the function is a dictator or far from it. Since we are interested in a test with perfect completeness this puts some restriction on the set of $k$ queried locations. If we denote $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k$ as the set of queried locations then the $i^{th}$ bit from $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k)$ should satisfy the predicate $\mathcal{P}$. This is because, the test should always accept no matter which dictator $f$ is.

Let $\mu$ denotes a distribution on $\mathcal{P}^{-1}(1)$. One natural way to sample $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k)$ such that the test has a perfect completeness guarantee is for each coordinate $i \in [n]$ independently sample $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k)_i$ from distribution $\mu$. This is what we do in our dictatorship test for a specific distribution $\mu$ supported on $\mathcal{P}^{-1}(1)$. It is now easy to see that the test accepts with probability $1$ of $f$ is an $i^{th}$ dictator for any $i \in [n]$.

Analyzing the soundness of a test is the main technical task. First note that the soundness parameter of the test depends on $\mathcal{P}^{-1}(1)$ as it can be easily verified that if $f$ is a random function, which is far from any dictator function, then the test accepts with probability at least $\frac{|\mathcal{P}^{-1}(1)|}{2^k}$. Thus, for a better soundness guarantee we want $P$ to have as small support as possible. The acceptance probability of the test is given by the following expression:

$$\begin{aligned}\Pr[\text{Test accepts } f] &= \mathbf{E}[\mathcal{P}(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_k))] \\ &= \frac{|\mathcal{P}^{-1}(1)|}{2^k} + \mathbf{E}\left[\sum_{S\subseteq[k], S\neq\emptyset} \hat{\mathcal{P}}(S) \prod_{i\in S} f(\boldsymbol{x}_i)\right]\end{aligned}$$

Thus, in order to show that the test accepts with probability at most $\frac{|\mathcal{P}^{-1}(1)|}{2^k} + \varepsilon$ it is enough to show that all the expectations $E_S := |\,\mathbf{E}[\prod_{i\in S} f(\boldsymbol{x}_i)]|$ are small if $f$ is far from any dictator function. Recall that at this point, we can have any predicate $\mathcal{P}$ on $k$ bits which the verifier uses. As we will see later, for the soundness analysis we need the predicate $\mathcal{P}$ to satisfy certain properties.

For the rest of the section, assume that the given function $f$ is such that the low degree influence of every variable $i \in [n]$ is very small constant $\tau$. If $f$ is a constant degree function (independent of $n$) then the usual analysis goes by invoking invariance principle to claim that the quantity $E_S$ does not change by much if we replace the distribution $\mu$ to a distribution $\xi$ over Gaussian random variable with the same first and second moments. An advantage of moving to a Gaussian distribution is that if $\mu$ was a uniform and pairwise independent distribution then so is $\xi$ and using the fact that a pairwise independence implies a total independence in the Gaussian setting, we have $E_S \approx |\prod_{i\in S} \mathbf{E}[f(\boldsymbol{g}_i)]|$. Since we assumed that $f$ was a balanced function we have $\mathbf{E}[f(\boldsymbol{g}_i)]| = 0$ and hence we can say that the quantity $E_S$ is very small.

There are two main things we need to take care in the above argument. $1)$ We assumed that $f$ is a low degree function and in general it may not be true. $2)$ The argument crucially needed $\mu$ to satisfy pairwise independence condition and hence it puts some restriction on the size of $\mathcal{P}^{-1}(1)$ (Ideally, we would like $|\mathcal{P}^{-1}(1)|$ to be as small as possible for a better soundness guarantee). We take care of $(1)$, as in the previous works [TY15, OW09a, AM09] etc., by requiring the distribution $\mu$ to have *correlation* bounded away from $1$. This can be achieved by making sure the support of $\mu$ is *connected* - for every coordinate $i \in [k]$ there exists $a, b \in \mathcal{P}^{-1}(1)$ which differ at the $i^{th}$ location. For such distribution, we can add independent *noise* to each co-ordinate without changing the quantity $E_S$ by much. Adding independent noise has the effect that it damps the higher order fourier coefficients of $f$ and the function behaves as a low degree function. We can now apply invariance principle to claim that $E_S \approx 0$. This was the approach in [TY15] and they could find a distribution $\mu$ whose support size is $2k+3$ which is connected and pairwise independent.

In order to get an improvement in the soundness guarantee, our main technical contribution is that we can still get the overall soundness analysis to go through even if $\mu$ does not support pairwise independence condition. To this end, we start with a distribution $\mu$ whose support size is $2k+1$ and has the property that it is *almost* pairwise independent. Since we lack pairwise independence, it introduces few obstacles in the above mentioned analysis. First, the *amount* of noise we can add to each co-ordinate has some limitations. Second, because of the limited amount of independent noise, we can no longer say that the function $f$ behaves as a low degree function after adding the noise. With the limited amount of noise, we can say that $f$ behaves as a low degree function as long as it does not have a large fourier mass in some interval i.e the fourier mass corresponding to $\hat{f}(T)^2$ such that $|T| \in (s, S)$ for some constant sized interval $(s, S)$ independent of $n$. We handle this obstacle by designing a family of distributions $\mu_1, \mu_2, \ldots, \mu_r$ for large enough $r$ such that the intervals that we cannot handle for different $\mu_i$'s are disjoint. Also, each $\mu_i$ has the same support and is almost pairwise independent. We then let our final test distribution as first selecting $i \in [r]$ u.a.r and then doing the test with the corresponding distribution $\mu_i$. Since the total fourier mass of a $-1/+1$ function is bounded by $1$ and $f$ was fixed before running the test it is very unlikely that $f$ has a large fourier mass in the interval corresponding to the selected distribution $\mu_i$. Hence, we can conclude that for this overall distribution, $f$ behaves as a low degree function. We note that this approach of using family of distributions was used in [Hås14] to construct a 3-bit PCP with perfect completeness. There it was used in the composition step.

To finish the soundness analysis, let $\tilde{f}$ be the low degree part of $f$. The argument in the previous paragraph concludes that $E_S \approx |\,\mathbf{E}[\prod_{i \in S} \tilde{f}(\boldsymbol{x}_i)]|$. As in the previous work, we can now apply invariance principle to claim that $E_S \approx |\,\mathbf{E}[\prod_{i \in S} \tilde{f}(\boldsymbol{g}_i)]|$ where the $i^{th}$ coordinate $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k)_i$ is distributed according to $\xi$ which is almost pairwise independent. We can no longer bring the expectation inside as our distribution lacks independence. To our rescue, we have that the degree of $\tilde{f}$ is bounded by some constant independent of $n$. We then prove that low degree functions are robust w.r.t slight perturbation in the inputs on average. This lets us conclude $\mathbf{E}[\prod_{i \in S} \tilde{f}(\boldsymbol{g}_i)] \approx \mathbf{E}[\prod_{i \in S} \tilde{f}(\boldsymbol{h}_i)]$ where $(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_k)_i$ is pairwise independent. We now use the property of independence of Gaussian distribution and bring the expectation inside to conclude that $E_S \approx |\,\mathbf{E}[\prod_{i \in S} \tilde{f}(\boldsymbol{h}_i)]| = |\prod_{i \in S} \mathbf{E}[\tilde{f}(\boldsymbol{h}_i)]| = 0$.

## 5.3 Query efficient Dictatorship Test

We are now ready to describe our dictatorship test. The test queries a function at $k$ locations and based on the $k$ bits received decides if the function is a dictator or far from it. The check on the received $k$ bits is based on a predicate with few accepting inputs which we describe next.

### 5.3.1 The Predicate

Let $k = 2^m - 1$ for some $m > 2$. Let the coordinates of the predicate is indexed by elements of $\mathbf{F}_2^m \backslash \mathbf{0} =: \{w_1, w_2, \ldots, w_{2^m - 1}\}$. The Hadamard predicate $H_k$ has following satisfying assignments:

$$H_k = \{x \in \{0,1\}^k | \exists a \in \mathbf{F}_2^m \backslash \mathbf{0} \text{ s.t } \forall i \in [k], x_i = a \cdot w_i\}$$

We will identify the set of satisfying assignments in $H_k$ with the variables $h_1$, $h_2$, $\ldots$, $h_k$.

Our final predicate $\mathcal{P}_k$ is the above predicate along with few more satisfying assignments. More precisely, we add all the assignments which are at a hamming distance at most $1$ from $0^k$ i.e. $\mathcal{P}_k = H_k \cup_{i=1}^k e_i \cup 0^k$.

### 5.3.2 The Distribution $\mathcal{D}_{k,\varepsilon}$

For $0 < \varepsilon \leqslant \frac{1}{k^2}$, consider the following distribution $\mathcal{D}_{k,\varepsilon}$ on the set of satisfying assignments of $\mathcal{P}_k$ where $\alpha := (k-1)\varepsilon$.

$$\begin{array}{rl}
\text{Probabilities} & \text{Assignments} \\
\mathcal{D}_{k,\varepsilon} \leftarrow & \left\{ \begin{array}{cccc} x_1 & x_2 & \cdots\cdots & x_k \end{array} \right. \\
\dfrac{1}{1-\alpha}\left(\dfrac{1}{k+1} - \alpha\right) \leftarrow & \left\{ \begin{array}{cccc} 0 & 0 & \cdots\cdots & 0 \end{array} \right. \\
\dfrac{1}{1-\alpha}\left(\dfrac{1}{k+1} - \varepsilon\right) \leftarrow & \left\{ \begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_k \end{array} \right. \\
\dfrac{\varepsilon}{1-\alpha} \leftarrow & \left\{ \begin{array}{cccc} 1 & 0 & \cdots\cdots & 0 \\ 0 & 1 & \cdots\cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots\cdots & 1, \end{array} \right.
\end{array}$$

where each $h_i$ gets a probability mass $\frac{1}{1-\alpha}(\frac{1}{k+1} - \varepsilon)$ and each $e_i$ gets weight $\frac{\varepsilon}{1-\alpha}$. The reasoning behind choosing this distribution is as follows: An uniform distribution on $H_k \cup 0^k$ has a property that it is uniform on every single coordinate and also pairwise independent. These two properties are very useful proving the soundness guarantee. One more property which we require is that the distribution has to be *connected*. In order to achieve this, we add $k$ extra assignment $\{e_1, e_2, \ldots, e_k\}$ and force the distribution to be supported on all $H_k \cup_{i=1}^{k} e_i \cup 0^k$. Even though by adding extra assignments, we loose the pairwise independent property we make sure that the final distribution is *almost* pairwise independent.

We now list down the properties of this distribution which we will use in analyzing the dictatorship test. This is proved in Section 5.5.2.

**Observation 5.3.1** *The distribution $\mathcal{D}_{k,\varepsilon}$ above has the following properties:*

1. *$\mathcal{D}_{k,\varepsilon}$ is supported on $\mathcal{P}_k$.*
2. *Marginal on every single coordinate is uniform.*
3. *For $i \neq j$, covariance of two variables $x_i, x_j$ sampled form above distribution is: $\texttt{Cov}[x_i, x_j] = -\frac{\varepsilon}{2(1-\alpha)}$.*
4. *If we view $\mathcal{D}_{k,\varepsilon}$ as a joint distribution on space $\prod_{i=1}^{k} \mathcal{X}^{(i)}$ where each $\mathcal{X}^{(i)} = \{0,1\}$, then for all $i \in [k]$, $\rho\left(\mathcal{X}^{(i)}, \prod_{j\in[k]\setminus\{i\}} \mathcal{X}^{(j)}; \mathcal{D}_{k,\varepsilon}\right) \leqslant 1 - \frac{\varepsilon^2}{2(1-\alpha)^2}$. (See Definition 5.5.1 for the definition of $\rho$.)*

### 5.3.3 Dictatorship Test

We will switch the notations from $\{0,1\}$ to $\{+1,-1\}$ where we identify $+1$ as $0$ and $-1$ as $1$. Let $f : \{-1,+1\}^n \to \{-1,+1\}$ be a given boolean function. We also assume that $f$ is folded i.e. for every $\boldsymbol{x} \in \{-1,+1\}^n$, $f(\boldsymbol{x}) = -f(-\boldsymbol{x})$. We think of $\mathcal{P}_k$ as a function $\mathcal{P}_k : \{-1,+1\}^k \to \{0,1\}$ such that $P_k(z) = 1$ iff $z \in \mathcal{P}_k$. Consider the following dictatorship test:

**Test** $\mathcal{T}_{k,\delta}$

1. Sample $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k \in \{-1, +1\}^n$ as follows:

    (a) For each $i \in [n]$, independently sample $((\boldsymbol{x}_1)_i, (\boldsymbol{x}_2)_i, \cdots, (\boldsymbol{x}_k)_i)$ according to the distribution $\mathcal{D}_{k,\delta}$.

2. Check if $(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_k)) \in \mathcal{P}_k$.

The final test distribution is basically the above test where the parameter $\delta$ is chosen from an appropriate distribution. For a given $\frac{1}{k^2} \geqslant \varepsilon > 0$, let $\mathsf{err} = \frac{\varepsilon/5}{2^k}$ and define the following quantities : $\varepsilon_0 = \varepsilon$ and for $j \geqslant 0$, $\varepsilon_{j+1} = \mathsf{err} \cdot 2^{-\left(\frac{k^{10}}{\mathsf{err}^3 \varepsilon_j}\right)^k}$.

**Test** $\mathcal{T}'_{k,\varepsilon}$

1. Set $r = \left(\frac{k}{\mathsf{err}}\right)^2$
2. Select $j$ from $\{1, 2, \ldots, r\}$ uniformly at random.
3. Set $\delta = \varepsilon_j$
4. Run test $\mathcal{T}_{k,\delta}$.

We would like to make a remark that this particular setting of $\varepsilon_{j+1}$ is not very important. For our analysis, we need a sequence of $\varepsilon_j$'s such that each subsequent $\varepsilon_j$ is sufficiently small compared to $\varepsilon_{j-1}$.

## 5.4 Analysis of the Dictatorship Test

**Notation:** We can view $f : \{-1,+1\}^n \rightarrow \{-1,+1\}$ as a function over $n$-fold product set $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$ where each $\mathcal{X}_i = \{-1,+1\}^{\{i\}}$. In the test distribution $\mathcal{T}_{k,\delta}$, we can think of $\boldsymbol{x}_i$ sampled from the product distribution on $\mathcal{X}_1^{(i)} \times \mathcal{X}_2^{(i)} \times \cdots \times \mathcal{X}_n^{(i)}$. With these notations in hand, the overall distribution on $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k)$, from the test $\mathcal{T}_{k,\delta}$, is a $n$-fold product distribution from the space

$$\prod_{j=1}^{n} \left( \prod_{i=1}^{k} \mathcal{X}_j^{(i)} \right).$$

where we think of $\prod_{i=1}^{k} \mathcal{X}_j^{(i)}$ as correlated space. We define the parameters for the sake of notational convenience:

1. $\beta_j := \frac{\varepsilon_j}{1-(k-1)\varepsilon_j}$ be the minimum probability of an atom in the distribution $\mathcal{D}_{k,\varepsilon_j}$.

2. $s_{j+1} := \log(\frac{k}{\mathsf{err}})\frac{1}{\varepsilon_j^2}$ and $S_j = s_{j+1}$ for $0 \leqslant j \leqslant r$.

3. $\alpha_j := (k-1)\varepsilon_j$ for $j \in [r]$,

### 5.4.1 Completeness

Completeness is trivial, if $f$ is say $ith$ dictator then the test will be checking the following condition

$$((\boldsymbol{x}_1)_i, (\boldsymbol{x}_2)_i, \cdots, (\boldsymbol{x}_k)_i) \in \mathcal{P}_k$$

Using Observation 5.3.1(1), the distribution is supported on only strings in $\mathcal{P}_k$. Therefore, the test accepts with probability 1.

### 5.4.2 Soundness

**Lemma 5.4.1** *For every $\frac{1}{k^2} \geqslant \varepsilon > 0$ there exists $0 < \tau < 1, d \in \mathbf{N}^+$ such that the following holds: Suppose $f$ is such that for all $i \in [n]$, $\mathtt{inf}_i^{\leqslant d}(f) \leqslant \tau$, then the test $\mathcal{T}'_{k,\varepsilon}$ accepts with probability at most $\frac{2k+1}{2^k} + \varepsilon$. (Note: One can take $\tau$ such that $\tau^{\Omega_k(\mathsf{err}/10 s_r \log(1/\beta_r))} \leqslant \mathsf{err}$ and $d = \frac{\log(1/\tau)}{\log(1/\beta_r)}$.)*

**Proof:** The acceptance probability of the test is given by the following expression:

$$\Pr[\text{Test accepts } f] = \mathop{\mathbf{E}}_{\mathcal{T}'_{k,\varepsilon}} [\mathcal{P}_k(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_k))]$$

After expanding $P_k$ in terms of its Fourier expansion, we get

$$\begin{aligned}
\Pr[\text{Test accepts } f] &= \frac{2k+1}{2^k} + \mathop{\mathbf{E}}_{\mathcal{T}'_{k,\varepsilon}}\left[\sum_{S\subseteq[k], S\neq\varnothing} \hat{\mathcal{P}}_k(S) \prod_{i\in S} f(\boldsymbol{x}_i)\right] \\
&= \frac{2k+1}{2^k} + \sum_{S\subseteq[k], S\neq\varnothing} \hat{\mathcal{P}}_k(S) \mathop{\mathbf{E}}_{\mathcal{T}'_{k,\varepsilon}}\left[\prod_{i\in S} f(\boldsymbol{x}_i)\right] \\
&\leqslant \frac{2k+1}{2^k} + \sum_{S\subseteq[k], S\neq\varnothing} \left|\mathop{\mathbf{E}}_{\mathcal{T}'_{k,\varepsilon}}\left[\prod_{i\in S} f(\boldsymbol{x}_i)\right]\right| \qquad (|\hat{\mathcal{P}}_k(S)| \leqslant 1) \\
&= \frac{2k+1}{2^k} + \sum_{S\subseteq[k], |S|\geqslant 2} \left|\mathop{\mathbf{E}}_{\mathcal{T}'_{k,\varepsilon}}\left[\prod_{i\in S} f(\boldsymbol{x}_i)\right]\right|.
\end{aligned}$$

In the last equality, we used the fact that each $\boldsymbol{x}_i$ is distributed uniformly in $\{-1,+1\}^n$ and hence when $S = \{i\}$, $\mathbf{E}[f(\boldsymbol{x}_i)] = \hat{f}(\varnothing) = 0$. Thus, to prove the lemma it is enough to show that for all $S \subseteq [k]$ such that $|S| \geqslant 2$, $\mathbf{E}\left[\prod_{i\in S} f(\boldsymbol{x}_i)\right] \leqslant \frac{\varepsilon}{2^k}$. This follows from Lemma 5.4.2. ■

**Lemma 5.4.2** *For any $S \subseteq [k]$ such that $|S| \geqslant 2$,*

$$\left| \mathop{\mathbf{E}}_{j\in[r]} \left[ \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i\in S} f(\boldsymbol{x}_i) \right] \right] \right| \leqslant \frac{\varepsilon}{2^k}$$

The proof of this follows from the following Lemmas 5.4.3 , 5.4.4, 5.4.5.

**Lemma 5.4.3** *For any $j \in [r]$ and for any $S \subseteq [k]$, $|S| \geqslant 2$ such that $S = \{\ell_1, \ell_2, \dots, \ell_t\}$,*

$$\left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i\in S} f(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i\in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{x}_{\ell_i}) \right] \right| \leqslant 2\cdot \mathsf{err} + k\sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2}.$$

*where $\gamma_j = \frac{\mathsf{err}}{ks_j}$ and $d_{j,i}$ is a sequence given by $d_{j,1} = \frac{2k^2\cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right)$ and $d_{j,i} = (d_{j,1})^i$ for $1 < i \leqslant t$.*

**Lemma 5.4.4** *Let $j \in [r]$ and $\nu_j$ be a distribution on jointly distributed standard Gaussian variables with same covariance matrix as that of $\mathcal{D}_{k,\varepsilon_j}$. Then for any $S \subseteq [k]$, $|S| \geqslant 2$ such that $S = \{\ell_1, \ell_2, \dots, \ell_t\}$,*

$$\left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i\in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{(\boldsymbol{g}_1,\boldsymbol{g}_2,\dots,\boldsymbol{g}_k)\sim\nu_j^{\otimes n}} \left[ \prod_{\ell_i\in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{g}_i) \right] \right| \leqslant \mathsf{err}_2$$

*where $d_{j,i}$ from Lemma 5.4.3 and $\mathsf{err}_2 = \tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}$ (Note: $\Omega(.)$ hides a constant depending on $k$).*

**Lemma 5.4.5** *Let $k \geqslant 2$ and $S \subseteq [k]$ such that $|S| \geqslant 2$ and let $f : \mathbf{R}^n \to \mathbf{R}$ be a multilinear polynomial of degree $D \geqslant 1$ such that $\|f\|_2 \leqslant 1$. If $\mathcal{G}$ be a joint distribution* *on $k$ standard gaussian random variable with a covariance matrix $(1+\delta)\mathbf{I} - \delta\mathbf{J}$ and $\mathcal{H}$ be a distribution on $k$ independent standard gaussian then it holds that*

$$\left| \mathop{\mathbf{E}}_{\mathcal{G}^{\otimes n}} \left[ \prod_{i \in S} f(\boldsymbol{g}_i) \right] - \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} \left[ \prod_{i \in S} f(\boldsymbol{h}_i) \right] \right| \leqslant \delta \cdot (2k)^{2kD}$$

Proofs of Lemma 5.4.3 , 5.4.4, 5.4.5 appear in Section 5.5.1. We now prove Lemma 5.4.2 using the above three claims.

**Proof of Lemma 5.4.2:** Let $S = \{\ell_1, \ell_2, \ldots, \ell_t\}$. We are interested in getting an upper bound for the following expectation:

$$\left| \mathop{\mathbf{E}}_{j \in [r]} \left[ \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] \right] \right| \leqslant \mathop{\mathbf{E}}_{j \in [r]} \left[ \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] \right| \right].$$

Let us look at the inner expectation first. Let $\gamma_j = \frac{\mathsf{err}}{ks_j}$ and the sequence $d_{j,i}$ be from Lemma 5.4.3. We can upper bound the inner expectation as follows:

$$\left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] \right| \leqslant \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}} (\boldsymbol{x}_{\ell_i}) \right] \right| + 2 \cdot \mathsf{err} + k \sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2}$$

(by Lemma 5.4.3)

$$\leqslant \left| \mathop{\mathbf{E}}_{(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}} (\boldsymbol{g}_i) \right] \right| + \quad \text{(by Lemma 5.4.4)}$$

$$\mathsf{err}_2 + 2 \cdot \mathsf{err} + k \sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2} \qquad (5.4.1)$$

where $\mathsf{err}_2 = \tau^{\Omega_k(\gamma_j / \log(1/\beta_j))}$ and $\nu_j$ has the same covariance matrix as $\mathcal{D}_{k,\varepsilon_j}$. If we let $\delta_j = \frac{2\varepsilon_j}{1-\alpha_j}$ then using Observation 5.3.1(3), the covariance matrix is precisely $(1+\delta_j)\mathbf{I} - \delta_j \mathbf{J}$ (note that we switched from $0/1$ to $-1/+1$ which changes the covaraince by a factor of $4$). Each of the functions $(T_{1-\gamma_j} f)^{\leqslant d_{j,i}}$ has $\ell_2$ norm upper bounded by $1$ and degree at most $d_{j,t}$. We can now apply Lemma 5.4.5 to conclude that

$$\left|\mathop{\mathbf{E}}_{(\boldsymbol{g}_1,\boldsymbol{g}_2,\dots,\boldsymbol{g}_k)\sim\nu_j^{\otimes n}}\left[\prod_{\ell_i\in S}(T_{1-\gamma_j}f)^{\leqslant d_{j,i}}(\boldsymbol{g}_i)\right]\right| \leqslant \\ \left|\mathop{\mathbf{E}}_{(\boldsymbol{h}_1,\boldsymbol{h}_2,\dots,\boldsymbol{h}_k)}\left[\prod_{\ell_i\in S}(T_{1-\gamma_j}f)^{\leqslant d_{j,i}}(\boldsymbol{h}_i)\right]\right| + \delta_j\cdot(2k)^{2kd_{j,t}}, \tag{5.4.2}$$

where $\boldsymbol{h}_i$'s are independent and each $\boldsymbol{h}_i$ is distributed according to $\mathcal{N}(0,1)^n$. Thus,

$$\mathop{\mathbf{E}}_{(\boldsymbol{h}_1,\boldsymbol{h}_2,\dots,\boldsymbol{h}_k)}\left[\prod_{\ell_i\in S}(T_{1-\gamma_j}f)^{\leqslant d_{j,i}}(\boldsymbol{h}_i)\right] = \prod_{\ell_i\in S}\mathop{\mathbf{E}}_{\boldsymbol{h}_i}\left[(T_{1-\gamma_j}f)^{\leqslant d_{j,i}}(\boldsymbol{h}_i)\right] \\ = \left(\widehat{(T_{1-\gamma_j}f)}^{\leqslant d_{j,i}}(\varnothing)\right)^t = (\hat{f}(\varnothing))^t = 0, \tag{5.4.3}$$

where we used the fact that $f$ is a folded function in the last step. Combining (5.4.1), (5.4.2) and (5.4.3), we get

$$\left|\mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}}\left[\prod_{\ell_i\in S}f(\boldsymbol{x}_{\ell_i})\right]\right| \leqslant \\ \left(\delta_j\cdot(2k)^{2kd_{j,t}}\right) + \left(\tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}\right) + 2\cdot\mathsf{err} + k\sqrt{\sum_{s_j\leqslant|T|\leqslant S_j}\hat{f}(T)^2} \tag{5.4.4}$$

We now upper bound the first term. For this, we use a very generous upper bounds $d_{j,1} \leqslant \frac{k^5}{\mathsf{err}^3} \frac{1}{\varepsilon_{j-1}^2}$ and $\delta_j \leqslant 4\varepsilon_j$.

$$\begin{aligned}
\delta_j \cdot (2k)^{2kd_{j,t}} &\leqslant \left(4\varepsilon_j \cdot (2k)^{2\mathbf{d}_{j,k} k}\right) \\
&\leqslant \varepsilon_j \cdot 2^{\left(\frac{k^{10}}{\mathsf{err}^3 \varepsilon_{j-1}}\right)^k} \\
&\leqslant \mathsf{err}. && \left(\text{using } \varepsilon_j = \mathsf{err} \cdot 2^{-\left(\frac{k^{10}}{\mathsf{err}^3 \varepsilon_{j-1}}\right)^k}\right)
\end{aligned}$$

The second term in (5.4.4) can also be upper bounded by $\mathsf{err}$ by choosing small enough $\tau$.

$$\max_j \{\left(\tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}\right)\} \leqslant \left(\tau^{\Omega_k(\gamma_r/\log(1/\beta_r))}\right) \leqslant \mathsf{err}.$$

Finally, taking the outer expectation of (5.4.4), we get

$$\mathop{\mathbf{E}}_{j \in [r]} \left[ \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] \right| \right] \leqslant 4 \cdot \mathsf{err} + k \mathop{\mathbf{E}}_{j \in r} \left[ \sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2} \right].$$

Using Cauchy-Schwartz inequality,

$$\mathop{\mathbf{E}}_{j \in [r]} \left[ \sqrt{\sum_{s_j < |T| < S_j} \hat{f}(T)^2} \right] \leqslant \sqrt{\mathop{\mathbf{E}}_{j \in [r]} \left[ \sum_{s_j < |T| < S_j} \hat{f}(T)^2 \right]} \leqslant \frac{1}{\sqrt{r}},$$

where the last inequality uses the fact that the intervals $(s_j, S_j)$ are disjoint for

$j \in [r]$ and $\|f\|_2^2 = \sum_T \hat{f}(T)^2 \leqslant 1$. The final bound we get is

$$\begin{aligned}\left|\mathop{\mathbf{E}}_{j\in[r]}\left[\mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}}\left[\prod_{\ell_i\in S} f(\boldsymbol{x}_{\ell_i})\right]\right]\right| &\leqslant \mathop{\mathbf{E}}_{j\in[r]}\left[\left|\mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}}\left[\prod_{\ell_i\in S} f(\boldsymbol{x}_{\ell_i})\right]\right|\right] \\ &\leqslant 4\cdot \mathsf{err} + \frac{k}{\sqrt{r}} \\ &\leqslant 5.\mathsf{err} \\ &\leqslant \frac{\varepsilon}{2^k}\end{aligned}$$

as required. ■

## 5.5 Deferred proofs

### 5.5.1 Proofs of Lemma 5.4.3 , 5.4.4 & 5.4.5

In this section, we provide proofs of three crucial lemmas which we used in proving the soundness analysis of our dictatorship test. We start with some more preliminaries.

#### 5.5.1.1 Correlated Spaces

Let $\Omega_1 \times \Omega_2$ be two correlated spaces and $\mu$ denotes the joint distribution. Let $\mu_1$ and $\mu_2$ denote the marginal of $\mu$ on space $\Omega_1$ and $\Omega_2$ respectively. The correlated space $\rho(\Omega_1 \times \Omega_2; \mu)$ can be represented as a bipartite graph on $(\Omega_1, \Omega_2)$ where $x \in \Omega_1$ is connected to $y \in \Omega_2$ iff $\mu(x, y) > 0$. We say that the correlated spaces is $connected$ if this underlying graph is connected.

We need a few definitions and lemmas related to correlated spaces defined by Mossel [Mos08].

**Definition 5.5.1** *Let $(\Omega_1 \times \Omega_2, \mu)$ be a finite correlated space, the correlation between $\Omega_1$ and $\Omega_2$ with respect to $\mu$ us defined as*

$$\rho(\Omega_1, \Omega_2; \mu) := \max_{\substack{f:\Omega_1 \to \mathbf{R}, \mathbf{E}[f]=0, \mathbf{E}[f^2] \leqslant 1 \\ g:\Omega_2 \to \mathbf{R}, \mathbf{E}[g]=0, \mathbf{E}[g^2] \leqslant 1}} \mathop{\mathbf{E}}_{(x,y)\sim\mu} [|f(x)g(y)|].$$

The following result (from [Mos08]) provides a way to upper bound correlation of a correlated spaces.

**Lemma 5.5.2** *Let $(\Omega_1 \times \Omega_2, \mu)$ be a finite correlated space such that the probability of the smallest atom in $\Omega_1 \times \Omega_2$ is at least $\alpha > 0$ and the correlated space is connected then*

$$\rho(\Omega_1, \Omega_2; \mu) \leqslant 1 - \alpha^2/2$$

**Definition 5.5.3 (Markov Operator)** *Let $(\Omega_1 \times \Omega_2, \mu)$ be a finite correlated space, the Markov operator, associated with this space, denoted by $U$, maps a function $g : \Omega_2 \to \mathbf{R}$ to functions $Ug : \Omega_1 \to \mathbf{R}$ by the following map:*

$$(Ug)(x) := \mathop{\mathbf{E}}_{(X,Y)\sim\mu} [g(Y) \mid X = x].$$

In the soundness analysis of our dictatorship test, we will need to understand the Efron-Stein decomposition of $Ug$ in terms of the decomposition of $g$. The following proposition gives a way to relate these two decompositions.

**Proposition 5.5.4 ([Mos08, Proposition 2.11])** *Consider a product correlated space $(\prod_{i=1}^n \Omega_i^{(1)} \times \prod_{i=1}^n \Omega_i^{(2)}, \prod_{i=1}^n \mu_i)$. Let $g : \prod_{i=1}^n \Omega_i^{(2)} \to \mathbf{R}$ be a function and $U$ be the Markov operator mapping functions form space $\prod_{i=1}^n \Omega_i^{(2)}$ to the functions on space $\prod_{i=1}^n \Omega_i^{(1)}$. If $g = \sum_{S\subseteq[n]} g_S$ and $Ug = \sum_{S\subseteq[n]} (Ug)_S$ be the Efron-Stein decomposition* *of $g$ and $Ug$ respectively then,*

$$(Ug)_S = U(g_S)$$

*i.e. the Efron-Stein decomposition commutes with Markov operators.*

Finally, the following proposition says that if the correlation between two spaces is bounded away from 1 then *higher order* terms in the Efron-Stein decomposition of $Ug$ has a very small $\ell_2$ norm compared to the $\ell_2$ norm of the corresponding higher order terms in the Efron-Stein decomposition of $g$.

**Proposition 5.5.5 ([Mos08, Proposition 2.12])** *Assume the same setting as that of Proposition 5.5.4 and furthermore assume that $\rho(\Omega_i^{(1)}, \Omega_i^{(2)}; \mu_i) \leqslant \rho$ for all $i \in [n]$, then for all $g$ it holds that*

$$\|U(g_S)\|_2 \leqslant \rho^{|S|}\|g_S\|_2.$$

#### 5.5.1.2 Hypercontractivity

**Definition 5.5.6** *A random variable $r$ is said to be $(p, q, \eta)$-hypercontractive if it satisfies*

$$\|a + \eta r\|_q \leqslant \|a + r\|_p$$

*for all $a \in \mathbf{R}$.*

We note down the hypercontractive parameters for Rademacher random variable (uniform over $\pm 1$) and standard gaussian random variable.

**Theorem 5.5.7 ([Wol07][Ole03])** *Let $X$ denote either a uniformly random $\pm 1$ bit, a standard one-dimensional Gaussian. Then $X$ is $\left(2, q, \frac{1}{\sqrt{q-1}}\right)$-hypercontractive.*

The following proposition says that the higher norm of a low degree function w.r.t hypercontractive sequence of ensembles is bounded above by its second norm.

**Proposition 5.5.8 ([MOO05])** *Let $\boldsymbol{x}$ be a $(2, q, \eta)$-hypercontractive sequence of ensembles and $Q$ be a multilinear polynomial of degree $d$. Then*

$$\|Q(\boldsymbol{x})\|_q \leqslant \eta^{-d} \|Q(\boldsymbol{x})\|_2$$

#### 5.5.1.3 Invariance Principle

Let $\mu$ be any distribution on $\{-1, +1\}^k$. Consider the following distribution on $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in \{-1, +1\}^n$ such that independently for each $i \in [n]$, $((\boldsymbol{x}_1)_i$, $(\boldsymbol{x}_2)_i$, $\ldots$, $(\boldsymbol{x}_k)_i)$ is sampled from $\mu$. We will denote this distribution as $\mu^{\otimes n}$. We are interested in evaluation of a multilinear polynomial $f : \mathbf{R}^n \to \mathbf{R}$ on $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k)$ sampled as above.

Invariance principle shows the closeness between two different distributions w.r.t some quantity of interest. We are now ready to state the version of the invariance principle from [Mos08] that we need.

**Theorem 5.5.9 ([Mos08])** *For any $\alpha > 0, \varepsilon > 0, k \in \mathbf{N}^+$ there are $d, \tau > 0$ such that the following holds: Let $\mu$ be the distribution on $\{+1, -1\}^k$ satisfying*

1. $\mathbf{E}_{x\sim\mu}[x_i] = 0$ *for every* $i \in [k]$
2. $\mu(x) \geqslant \alpha$ *for every* $x \in \{-1, +1\}^k$ *such that* $\mu(x) \neq 0$

*Let $\nu$ be a distribution on standard jointly distributed Gaussian variables with the same covariance matrix as distribution $\mu$. Then, for every set of $k$ $(d, \tau)$-quasirandom* *multilinear polynomials $f_i : \mathbf{R}^n \to \mathbf{R}$, and suppose* $\mathsf{Var}[f_i^{>d}] \leqslant (1-\gamma)^{2d}$ *for $0 < \gamma < 1$ it holds that*

$$\left| \mathop{\mathbf{E}}_{(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_k)\sim\mu^{\otimes n}} \left[ \prod_{i=1}^{k} f_i(\boldsymbol{x}_i) \right] - \mathop{\mathbf{E}}_{(\boldsymbol{g}_1,\boldsymbol{g}_2,\ldots,\boldsymbol{g}_k)\sim\nu^{\otimes n}} \left[ \prod_{i=1}^{k} f_i(\boldsymbol{g}_i) \right] \right| \leqslant \varepsilon$$

*(Note: one can take $d = \frac{\log(1/\tau)}{\log(1/\alpha)}$ and $\tau$ such that $\varepsilon = \tau^{\Omega(\gamma/\log(1/\alpha))}$, where $\Omega(.)$ hides constant depending only on $k$.)*

#### 5.5.1.4 Moving to a low degree function

The following lemma, at a very high level, says that if change $f$ to its low degree *noisy version* then the loss we incur in the expected quantity is small.

**Lemma 5.5.10 (Restatement of Lemma 5.4.3)** *For any $j \in [r]$ and for any $S \subseteq [k]$, $|S| \geqslant 2$ such that $S = \{\ell_1, \ell_2, \ldots, \ell_t\}$,*

$$\left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{x}_{\ell_i}) \right] \right| \leqslant 2 \cdot \mathsf{err} + k \sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2}.$$

*where $\gamma_j = \frac{\mathsf{err}}{ks_j}$ and $d_{j,i}$ is a sequence given by $d_{j,1} = \frac{2k^2 \cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right)$ and $d_{j,i} = (d_{j,1})^i$ for $1 < i \leqslant t$.*

**Proof:** The proof is presented in two parts. We first prove an upper bound on

$$\Gamma_1 := \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] \right| \leqslant \mathsf{err} + k \sqrt{\sum_{s_j \leqslant |T| \leqslant S_j} \hat{f}(T)^2} \tag{5.5.1}$$

and then an upper bound on

$$\Gamma_2 := \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{x}_{\ell_i}) \right] \right| \leqslant \mathsf{err}. \tag{5.5.2}$$

Note that both these upper bounds are enough to prove the lemma.

**Upper Bounding** $\Gamma_1$: The following analysis is very similar to the one in [TY15], we reproduce it here for the sake of completeness. The first upper bound is obtained by getting the upper bound for the following, for every $a \in [t]$.

$$\Gamma_{1,a} := \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \geqslant a} f(\boldsymbol{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i > a} f(\boldsymbol{x}_{\ell_i}) \prod_{i \leqslant a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] \right| \tag{5.5.3}$$

Note that by triangle inequality, $\Gamma_1 \leqslant \sum_{a \in [t]} \Gamma_{1,a}$.

$$\begin{aligned} (5.5.3) &= \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( f(\boldsymbol{x}_{\ell_a}) - T_{1-\gamma_j} f(\boldsymbol{x}_{\ell_a}) \right) \prod_{i > a} f(\boldsymbol{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] \right| \\ &= \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( id - T_{1-\gamma_j} \right) f(\boldsymbol{x}_{\ell_a}) \prod_{i > a} f(\boldsymbol{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] \right| \\ &= \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ U \left( \left( id - T_{1-\gamma_j} \right) f \right) (\boldsymbol{x}_{\{\ell_i : i \in [t] \setminus \{a\}\}}) \prod_{i > a} f(\boldsymbol{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \right] \right| \end{aligned} \tag{5.5.4}$$

where $U$ is the Markov operator for the correlated probability space which maps functions from the space $\mathcal{X}^{(\ell_a)}$ to the space $\prod_{i \in [t] \setminus \{a\}} \mathcal{X}^{(\ell_i)}$. We can look at the above expression as a product of two functions, $F = \prod_{i > a} f \prod_{i < a} (T_{1-\gamma_j} f)$ and $G = U(id - T_{1-\gamma_j}) f)$. From Observation 5.3.1( 4), the correlation between spaces $\left( \mathcal{X}^{(\ell_a)}, \prod_{i \in [t] \setminus \{a\}} \mathcal{X}^{(\ell_i)} \right)$ is upper bounded by $1 - \left( \frac{\varepsilon_j}{1-\alpha_j} \right)^2 \leqslant 1 - \varepsilon_j^2 =: \rho_j$. Taking the Efron-Stein decomposition with respect to the product distribution, we have the following because of orthogonality of the Efron-Stein decomposition,

$$\begin{aligned}(5.5.4) = \left|\mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}}[G \times F]\right| &= \left|\sum_{T\subseteq[n]} \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}}[G_T \times F_T]\right| \\ \text{(by Cauchy-Schwartz)} &\leqslant \sqrt{\sum_{T\subseteq[n]} \|F_T\|_2^2}\sqrt{\sum_{T\subseteq[n]} \|G_T\|_2^2} \qquad (5.5.5)\end{aligned}$$

where the norms are with respect to $\mathcal{D}_{k,\varepsilon_j}^{\otimes n}$'s marginal distribution on the product distribution $\prod_{i\in[t]\setminus\{a\}} \mathcal{X}^{(\ell_i)}$. By orthogonality, the quantity $\sqrt{\sum_{T\subseteq[n]} \|F_T\|_2^2}$ is just $\|F\|_2$. As $F$ is product of function whose range is $[-1,+1]$, rane of $F$ is also $[-1,+1]$ and hence $\|F\|_2$ is at most 1. Therefore,

$$(5.5.5) \leqslant \sqrt{\sum_{T\subseteq[n]} \|G_T\|_2^2} \tag{5.5.6}$$

We have $G_T = (UG')_T$, where $G' = (id - T_{1-\gamma_j})f$. In $G'_T$, the Efron-Stein decomposition is with respect to the marginal distribution of $\mathcal{D}_{k,\varepsilon_j}^{\otimes n}$ on $\mathcal{X}^{(\ell_a)}$, which is just uniform (by Observation 5.3.1(2)). Using Proposition 5.5.4, we have $G_T = UG'_T = U(id - T_{1-\gamma_j})f_T$. Substituting in (5.5.6), we get

$$(5.5.6) = \sqrt{\sum_{T\subseteq[n]} \|U(if - T_{1-\gamma_j})f_T)\|_2^2} \tag{5.5.7}$$

We also have that the correlation is upper bounded by $\rho_j$. We can therefore apply Proposition 5.5.5, and conclude that for each $T \subseteq [n]$,

$$\|U(id - T_{1-\gamma_j})f_T\|_2 \leqslant \rho_j^{|T|}\|(id - T_{1-\gamma_j})f_T\|_2$$

where the norm on the right is with respect to the uniform distribution. Observe that

$$\|(id - T_{1-\gamma_j})f_T\|_2^2 = (1-(1-\gamma_j)^{|T|})^2 \hat{f}(T)^2$$

Substituting back into (5.5.7), we get

$$(5.5.7) \leqslant \sqrt{\sum_{T \subseteq [n]} \underbrace{\rho_j^{2|T|}(1-(1-\gamma_j)^{|T|})^2 \hat{f}(T)^2}_{\mathsf{Term}(\varepsilon_j, \gamma_j, T)}} \tag{5.5.8}$$

We will now break the above summation into three different parts and bound each part separately.

$$\Theta_1 := \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j}} \mathsf{Term}(\varepsilon_j, \gamma_j, T) \qquad \Theta_2 := \sum_{\substack{T \subseteq [n], \\ s_j < |T| < S_j}} \mathsf{Term}(\varepsilon_j, \gamma_j, T)$$

$$\Theta_3 := \sum_{\substack{T \subseteq [n], \\ |T| \geqslant S_j}} \mathsf{Term}(\varepsilon_j, \gamma_j, T)$$

- **Upper bounding $\Theta_1$:**

$$\begin{aligned}
\Theta_1 &= \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j}} \mathsf{Term}(\varepsilon_j, \gamma_j, T) \\
&= \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j}} \rho_j^{2|T|}(1-(1-\gamma_j)^{|T|})^2 \hat{f}(T)^2 \\
&\leqslant \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j}} (1-(1-\gamma_j)^{|T|})^2 \hat{f}(T)^2.
\end{aligned}$$

For every $|T| \leqslant s_j$ we have $1 - (1-\gamma_j)^{|T|} \leqslant \mathsf{err}_1/k$. Thus,

$$\Theta_1 \leqslant \left(\frac{\mathsf{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j}} \hat{f}(T)^2.$$

- **Upper bounding $\Theta_3$:**

$$\begin{aligned}
\Theta_3 &= \sum_{\substack{T \subseteq [n], \\ |T| \geqslant S_j}} \mathsf{Term}(\varepsilon_j, \gamma_j, T) \\
&= \sum_{\substack{T \subseteq [n], \\ |T| \geqslant S_j}} \rho_j^{2|T|}(1-(1-\gamma_j)^{|T|})^2 \hat{f}(T)^2 \\
&\leqslant \sum_{\substack{T \subseteq [n], \\ |T| \geqslant S_j}} \rho_j^{2|T|} \hat{f}(T)^2.
\end{aligned}$$

For every $|T| \geqslant S_j$ we have $\rho_j^{|T|} \leqslant (1-\varepsilon_j^2)^{|T|} \leqslant \mathsf{err}_1/k$. Thus,

$$\Theta_3 \leqslant \left(\frac{\mathsf{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \geqslant S_j}} \hat{f}(T)^2.$$

Substituting these upper bounds in (5.5.8),

$$\begin{aligned}
\Gamma_{1,a} &\leqslant \sqrt{\left(\frac{\mathsf{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \leqslant s_j or |T| \geqslant S_j}} \hat{f}(T)^2 + \sum_{\substack{T \subseteq [n], \\ s_j < |T| < S_j}} \hat{f}(T)^2} \\
&\leqslant \sqrt{\left(\frac{\mathsf{err}_1}{k}\right)^2 + \sum_{s_j < |T| < S_j} \hat{f}(T)^2} && \text{(since } \textstyle\sum_T \hat{f}(T)^2 \leqslant 1\text{)} \\
&\leqslant \frac{\mathsf{err}_1}{k} + \sqrt{\sum_{s_j < |T| < S_j} \hat{f}(T)^2}. && \text{(using concavity)}
\end{aligned}$$

The required upper bound on $\Gamma_1$ follows by using $\Gamma_1 \leqslant \sum_{a\in[t]} \Gamma_{1,a}$ and the above bound.

**Upper Bounding** $\Gamma_2$: We will now show an upper bound on $\Gamma_2$. The approach is similar to the previous case, we upper bound the following quantity for every $a \in [t]$

$$\begin{aligned}
&\Gamma_{2,a} \\
&:= \Bigg| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i\geqslant a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \prod_{i<a} (T_{1-\gamma_j} f^{\leqslant d_{j,i}})(\boldsymbol{x}_{\ell_i}) \right] \\
&\qquad - \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i>a} (T_{1-\gamma_j} f)(\boldsymbol{x}_{\ell_i}) \prod_{i\leqslant a} (T_{1-\gamma_j} f^{\leqslant d_{j,i}})(\boldsymbol{x}_{\ell_i}) \right] \Bigg| \\
&= \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( T_{1-\gamma_j} f(\boldsymbol{x}_{\ell_a}) - T_{1-\gamma_j} f^{\leqslant d_{j,a}}(\boldsymbol{x}_{\ell_a}) \right) \prod_{i>a} T_{1-\gamma_j} f(\boldsymbol{x}_{\ell_i}) \prod_{i<a} (T_{1-\gamma_j} f^{\leqslant d_{j,i}})(\boldsymbol{x}_{\ell_i}) \right] \right| \\
&= \left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( T_{1-\gamma_j} f^{> d_{j,a}}(\boldsymbol{x}_{\ell_a}) \right) \prod_{i>a} T_{1-\gamma_j} f(\boldsymbol{x}_{\ell_i}) \prod_{i<a} (T_{1-\gamma_j} f^{\leqslant d_{j,i}})(\boldsymbol{x}_{\ell_i}) \right] \right| \qquad (5.5.9)
\end{aligned}$$

By using Holder's inequality we can upper bound (5.5.9) as:

$$(5.5.9) \leqslant \|T_{1-\gamma_j} f^{>d_{j,a}}\|_2 \prod_{i>a} \|T_{1-\gamma_j} f\|_{2(t-1)} \prod_{i<a} \|T_{1-\gamma_j} f^{\leqslant d_{j,i}}\|_{2(t-1)}, \qquad (5.5.10)$$

where each norm is w.r.t the uniform distribution as marginal of each $\boldsymbol{x}_{\ell_i}$ is uniform in $\{+1,-1\}^n$. Now, $\|T_{1-\gamma_j} f\|_{2(t-1)} \leqslant 1$ as the range if $T_{1-\gamma_j} f$ is in $[-1,+1]$. To upper bound $\|T_{1-\gamma_j} f^{\leqslant d_{j,i}}\|_{2(t-1)}$, we use Proposition 5.5.8 and using the fact that $\{-1,+1\}$ uniform random variable is $(2, q, 1/\sqrt{q-1})$ hypercontractive (Theorem 5.5.7) to get

$$\|T_{1-\gamma_j} f^{\leqslant d_{j,i}}\|_{2(t-1)} \leqslant (2t-3)^{d_{j,i}} \|T_{1-\gamma_j} f^{\leqslant d_{j,i}}\|_2 \leqslant (2t)^{d_{j,i}}.$$

Plugging this in (5.5.10), we get

$$\begin{aligned}
(5.5.10) &\leqslant \|T_{1-\gamma_j} f^{>d_{j,a}}\|_2 \prod_{i<a} (2t)^{d_{j,i}} \leqslant (1-\gamma_j)^{d_{j,a}} \cdot \prod_{i<a} (2t)^{d_{j,i}} \\
&\leqslant e^{-\gamma_j d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} \\
&\leqslant e^{-\frac{\mathsf{err}}{k s_j} \cdot d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} \qquad (5.5.11)
\end{aligned}$$

Now,

$$\begin{aligned}
d_{j,1} \cdot d_{j,a-1} &= d_{j,a} \\
\frac{2k^2 \cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right) \cdot d_{j,a-1} &= d_{j,a} \\
\frac{k^2 \cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right) + \frac{k^2 \cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right) \cdot d_{j,a-1} &\leqslant d_{j,a} \\
\frac{k \cdot s_j}{\mathsf{err}} \log\left(\frac{k}{\mathsf{err}}\right) + \frac{k^2 \cdot s_j}{\mathsf{err}} \cdot \log(2k) \cdot d_{j,a-1} &\leqslant d_{j,a} \\
\frac{k \cdot s_j}{\mathsf{err}} \cdot \left(\log\left(\frac{k}{\mathsf{err}}\right) + k \cdot d_{j,a-1} \log(2k)\right) &= d_{j,a} \\
\frac{k \cdot s_j}{\mathsf{err}} \cdot \log\left(\frac{k}{\mathsf{err}} (2k)^{k \cdot d_{j,a-1}}\right) &= d_{j,a}
\end{aligned}$$

This implies

$$\begin{aligned}
\log\left(\frac{k}{\mathsf{err}} (2k)^{k \cdot d_{j,a-1}}\right) &= \frac{\mathsf{err}}{k s_j} \cdot d_{j,a} \\
\Rightarrow \frac{k}{\mathsf{err}} (2k)^{k \cdot d_{j,a-1}} &= e^{\frac{\mathsf{err}}{k s_j} \cdot d_{j,a}} \\
\Rightarrow e^{-\frac{\mathsf{err}}{k s_j} \cdot d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} &= \frac{\mathsf{err}}{k}.
\end{aligned}$$

Thus from (5.5.11), we have $\Gamma_{2,a} \leqslant \frac{\mathsf{err}}{k}$. To conclude the proof, by triangle inequality we have $\Gamma_2 \leqslant \sum_{a \in [t]} \Gamma_{2,a} \leqslant \mathsf{err}$. ■

#### 5.5.1.5 Moving to the Gaussian setting

We are now in the setting of *low degree* polynomials because of Lemma 5.4.3. The following lemma let us switch from our test distribution to a Gaussian distribution with the same first two moments.

**Lemma 5.5.11 (Restatement of Lemma 5.4.4)** *Let $j \in [r]$ and $\nu_j$ be a distribution on jointly distributed standard Gaussian variables with same covariance matrix as that of $\mathcal{D}_{k,\varepsilon_j}$. Then for any $S \subseteq [k]$, $|S| \geqslant 2$ such that $S = \{\ell_1, \ell_2, \ldots, \ell_t\}$,*

$$\left| \mathop{\mathbf{E}}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{x}_{\ell_i}) \right] - \mathop{\mathbf{E}}_{(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leqslant d_{j,i}}(\boldsymbol{g}_i) \right] \right| \leqslant \mathsf{err}_2$$

*where $d_{j,i}$ from Lemma 5.4.3 and $\mathsf{err}_2 = \tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}$ (Note: $\Omega(.)$ hides a constant depending on $k$).*

**Proof:** Using the definition of $(d, \tau)$-quasirandom function and Fact 2.2.6, if $f$ is $(d, \tau)$- quasirandom then so is $T_{1-\gamma} f$ for any $0 \leqslant \gamma \leqslant 1$. Also, $T_{1-\gamma} f$ satisfies

$$\mathsf{Var}[T_{1-\gamma} f^{>d}] = \sum_{\substack{T \subseteq [n] \\ |T| > d}} (1-\gamma)^{2|T|} \hat{f}(T)^2 \leqslant (1-\gamma)^{2d} \cdot \sum_{\substack{T \subseteq [n] \\ |T| > d}} \hat{f}(T)^2 \leqslant (1-\gamma)^{2d}.$$

The lemma follows from a direct application of Theorem 5.5.9. ■

#### 5.5.1.6 Making Gaussian variables independent

Our final lemma allows us to make the Gaussian variables independent. Here we crucially need the property that the polynomials we are dealing with are low degree polynomials. Before proving Lemma 5.4.5, we need the following lemma which says that low degree functions are robust to small perturbations in the input on average.

**Lemma 5.5.12** *Let $f : \mathbf{R}^n \to \mathbf{R}$ be a multilinear polynomial of degree $d$ such that $\|f\|_2 \leqslant 1$ suppose $\boldsymbol{x}, \boldsymbol{z} \sim \mathcal{N}(0,1)^n$ be $n$-dimensional standard gaussian vectors such that $\mathbf{E}[x_i z_i] \geqslant 1 - \delta$ for all $i \in [n]$. Then*

$$\mathbf{E}[(f(\boldsymbol{x}) - f(\boldsymbol{z}))^2] \leqslant 2\delta d.$$

**Proof:** For $T \subseteq [n]$, we have

$$\mathbf{E}[\chi_T(\boldsymbol{x})\chi_T(\boldsymbol{z})] = \prod_{i \in T} \mathbf{E}[x_i z_i] \geqslant \prod_{i \in T}(1-\delta) \geqslant (1-\delta)^{|T|}$$

We now bound the following expression,

$$\begin{aligned}
\mathbf{E}[(f(\boldsymbol{x}) - f(\boldsymbol{z}))^2] &= \mathbf{E}[f(\boldsymbol{x})^2 + f(\boldsymbol{z})^2 - 2f(\boldsymbol{x})z(\boldsymbol{x})] \\
&= \sum_{T \subseteq [n], |T| \leqslant d} \hat{f}(T)^2 (2 - 2\,\mathbf{E}[\chi_T(\boldsymbol{x})\chi_T(\boldsymbol{z})]) \\
&\leqslant 2 \cdot \sum_{T \subseteq [n], |T| \leqslant d} \hat{f}(T)^2 (1 - (1-\delta)^{|T|}) \\
&\leqslant 2 \cdot \sum_{T \subseteq [n], |T| \leqslant d} \hat{f}(T)^2 \delta |T| \\
&\leqslant 2\delta d \cdot \sum_{T \subseteq [n], |T| \leqslant d} \hat{f}(T)^2 \leqslant 2\delta d,
\end{aligned}$$

where the last inequality uses $\|f\|_2 \leqslant 1$. ■

We are now ready to prove Lemma 5.4.5.

**Lemma 5.5.13 (Restatement of Lemma 5.4.5)** *Let $k \geqslant 2$ and $2 \leqslant t \leqslant k$ and let*

*$f : \mathbf{R}^n \to \mathbf{R}$ be a multilinear polynomial of degree $D \geqslant 1$ such that $\|f\|_2 \leqslant 1$. If $\mathcal{G}$ be a joint distribution on $k$ standard gaussian random variable with covariance matrix $(1+\delta)\mathbf{I} - \delta\mathbf{J}$ and $\mathcal{H}$ be a distribution on $k$ independent standard gaussian then it holds that*

$$\left| \mathop{\mathbf{E}}_{\mathcal{G}^{\otimes n}}\left[\prod_{i\in[t]} f(\boldsymbol{g}_i)\right] - \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}}\left[\prod_{i\in[t]} f(\boldsymbol{h}_i)\right]\right| \leqslant \delta \cdot (2k)^{2Dk}.$$

**Proof:** Let $\mathbf{\Sigma} = (1+\delta)\mathbf{I} - \delta\mathbf{J}$ be the covariance matrix. Let $\mathbf{M} = (1-\delta')((1+\beta)\mathbf{I} - \beta\mathbf{J})$ be a matrix such that $\mathrm{M}^{\mathbf{2}} = \mathbf{\Sigma}$. There are multiple $\mathrm{M}$ which satisfy $\mathrm{M}^{\mathbf{2}} = \mathbf{\Sigma}$. We chose the $\mathrm{M}$ stated above to make the analysis simpler. From the way we chose $\mathrm{M}$ and using the condition $\mathrm{M}^{\mathbf{2}} = \mathbf{\Sigma}$, it is easy to observe that $\beta$ and $\delta'$ should satisfy the following two conditions:

$$1 - \delta' = \frac{1}{\sqrt{1+(k-1)\beta^2}} \quad \text{and} \quad \frac{(k-2)\beta^2 - 2\beta}{1+(k-1)\beta^2} = -\delta.$$

Since $\mathcal{H}$ is a distribution of $k$ independent standard gaussians, we can generate a sample $x \sim \mathcal{G}$ by sampling $y \sim \mathcal{H}$ and setting $x = \mathrm{M}y$. In what follows, we stick to the following notation: $(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_k) \sim \mathcal{H}^{\otimes n}$ and $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k)_j = \mathrm{M}(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_k)_j$ for each $j \in [n]$.

Because of the way we chose to generate $g_i's$, we have for all $i \in [k]$ and $j \in [n]$, $\mathbf{E}[(\boldsymbol{g}_i)_j(\boldsymbol{h}_i)_j] = 1-\delta' \geqslant 1-k\beta^2$. To get an upper bound on $\beta$, notice that $\beta$ is a root of the quadratic equation $(k+\delta k-\delta-2)\beta^2 - 2\beta + \delta = 0$. Let $k' = (k+\delta k - \delta - 2)$, if $\beta_1, \beta_2$ are the roots of the equation then they satisfy: $k'\beta_1 + k'\beta_2 = 2$ and $(k'\beta_1)(k'\beta_2) = \delta k'$ and $\beta_1, \beta_2 > 0$. Thus, we have $\min\{k'\beta_1, k'\beta_2\} \leqslant \delta k'$ and hence, we can take $\beta$ such that $\beta \leqslant \delta$.

We wish to upper bound the following expression:

$$\Gamma := \left| \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} \left[ \prod_{i\in[t]} f(\boldsymbol{g}_i) - \prod_{i\in[t]} f(\boldsymbol{h}_i) \right] \right| .$$

Define the following quantity

$$\Gamma_i := \left| \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} \left[ \prod_{j=1}^{i-1} f(\boldsymbol{h}_j) \prod_{j=i}^{t} f(\boldsymbol{g}_j) - \prod_{j=1}^{i} f(\boldsymbol{h}_j) \prod_{j=i+1}^{t} f(\boldsymbol{g}_j) \right] \right| .$$

By triangle inequality, we have $\Gamma \leqslant \sum_{i\in[t]} \Gamma_i$. We now proceed with upper bounding $\Gamma_i$ for a given $i \in [t]$.

$$\begin{aligned}
\Gamma_i &= \left| \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} \left[ \prod_{j=1}^{i-1} f(\boldsymbol{h}_j) \prod_{j=i}^{t} f(\boldsymbol{g}_j) - \prod_{j=1}^{i} f(\boldsymbol{h}_j) \prod_{j=i+1}^{t} f(\boldsymbol{g}_j) \right] \right| \\
&= \left| \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} \left[ (f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i)) \cdot \prod_{j=1}^{i-1} f(\boldsymbol{h}_j) \prod_{j=i+1}^{t} f(\boldsymbol{g}_j) \right] \right| \\
&\leqslant \sqrt{\mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [(f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i))^2]} \cdot \prod_{j=1}^{i-1} \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [f(\boldsymbol{h}_j)^{2(t-1)}]^{\frac{1}{2(t-1)}} \prod_{j=i+1}^{t} \mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [f(\boldsymbol{g}_j)^{2(t-1)}]^{\frac{1}{2(t-1)}} ,
\end{aligned}$$

where the last step uses Holder's Inequality. Now, the marginal distribution on each $h_j$ and $g_j$ is identical which is $\mathcal{N}(0,1)^n$, we have

$$\begin{aligned}
\Gamma_i &\leqslant \sqrt{\mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [(f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i))^2]} \cdot \prod_{j=1}^{i-1} \|f\|_{2(t-1)} \prod_{j=i+1}^{t} \|f\|_{2(t-1)} \\
&\leqslant \sqrt{\mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [(f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i))^2]} \cdot (\|f\|_{2(t-1)})^{t-1}
\end{aligned}$$

Since a standard one dimensional Gaussian is $(2, q, 1/\sqrt{q-1})$-hypercontractive (Theorem 5.5.7), from Proposition 5.5.8 , $\|f\|_{2(t-1)} \leqslant (\sqrt{2t-3})^D \|f\|_2 \leqslant (\sqrt{2t-3})^D < (2t)^{D/2}$. Thus,

$$\Gamma_i \leqslant (2t)^{D(t-1)/2} \cdot \sqrt{\mathop{\mathbf{E}}_{\mathcal{H}^{\otimes n}} [(f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i))^2]}$$

Now, each $\boldsymbol{g}_i, \boldsymbol{h}_i$ are such that such that $\mathbf{E}[(\boldsymbol{g}_i)_j \cdot (\boldsymbol{h}_i)_j] = 1 - \delta' \geqslant 1 - k\delta^2$ for every $j \in [n]$. We can apply Lemma 5.5.12 to get $\mathbf{E}_{\mathcal{H}^{\otimes n}}[(f(\boldsymbol{g}_i) - f(\boldsymbol{h}_i))^2] \leqslant 2k\delta^2 D$. Hence, we can safely upper bound $\Gamma_i$ as

$$\Gamma_i \leqslant (2t)^{D(t-1)/2} \cdot 2k\delta D.$$

Therefore, $\Gamma \leqslant \sum_i \Gamma_i \leqslant t \cdot (2t)^{D(t-1)/2} \cdot 2k\delta D$ which is at most $2k^2\delta D \cdot (2k)^{Dk/2} \leqslant \delta \cdot (2k)^{2Dk}$ as required. ■

### 5.5.2 Proof of Observation 5.3.1

**Observation 5.5.1** *(Restatement of Observation 5.3.1) The distribution $\mathcal{D}_{k,\varepsilon}$ above has the following properties:*

1. *$\mathcal{D}_{k,\varepsilon}$ is supported on $\mathcal{P}_k$.*

2. *Marginal on every single coordinate is uniform.*

3. *For $i \neq j$, covariance of two variables $x_i, x_j$ sampled form above distribution is: $\texttt{Cov}[x_i, x_j] = -\frac{\varepsilon}{2(1-\alpha)}$.*

4. *If we view $\mathcal{D}_{k,\varepsilon}$ as a joint distribution on space $\prod_{i=1}^k \mathcal{X}^{(i)}$ where each $\mathcal{X}^{(i)} = \{0,1\}$, then for all $i \in [k]$, $\rho\left(\mathcal{X}^{(i)}, \prod_{j\in[k]\setminus\{i\}} \mathcal{X}^{(j)}; \mathcal{D}_{k,\varepsilon}\right) \leqslant 1 - \frac{\varepsilon^2}{2(1-\alpha)^2}$. (See Definition 5.5.1 for the definition of $\rho$.)*

**Proof:** We prove each of the observations about the distribution. The first property is straight-forward. To prove (2), we compute $\mathbf{E}[x_i]$ as follows.

$$\begin{aligned}\mathbf{E}[x_i] &= (k+1)\cdot\frac{1}{1-\alpha}\left(\frac{1}{k+1}-\varepsilon\right)\cdot\frac{1}{2}+\frac{\varepsilon}{1-\alpha}\\ &= \frac{1-\varepsilon(k+1)+2\varepsilon}{2(1-\alpha)}\\ &= \frac{1}{2}\end{aligned}$$

Consider the quantity $\underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_i x_j]$. If $x$ is sampled from $0$'s or $e_i$'s, the value is $0$. Moreover, we know that if it is sampled uniformly from $H_k \cup 0^k$, it is $1/4$ because of pairwise independence and the above fact. Therefore, we can write

$$\underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_i x_j] = (k+1)\frac{1}{1-\alpha}\left(\frac{1}{k+1}-\varepsilon\right)\frac{1}{4}$$

We know that $\underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_i] = \underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_j] = 1/2$. Therefore,

$$\begin{aligned}\texttt{Cov}[x_i, x_j] &= \underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_i x_j] - \underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_i]\underset{\mathcal{D}_{k,\varepsilon}}{\mathbf{E}}[x_j]\\ &= \frac{1}{4(1-\alpha)} - \frac{\varepsilon(k+1)}{4(1-\alpha)} - \frac{1}{4}\\ &= \frac{-\varepsilon}{2(1-\alpha)}\end{aligned}$$

We now show that the bi-partite graph $G\left(\mathcal{X}^{(i)}, \prod_{j\in[k]\setminus\{i\}}\mathcal{X}^{(j)}, E\right)$ where $(a,b) \in \mathcal{X}^{(i)} \times \prod_{j\in[k]\setminus\{i\}}\mathcal{X}^{(j)}$ is an edge iff $\Pr(a,b) > 0$, is connected. To see that the graph is connected, note that for both $0$ and $1$ on the left hand side, $0^{k-1}$ is a neighbor on the right hand side as the distribution's support includes $e_i$ for all $i$, and $0^k$. From the distribution, we see that the smallest atom is at least $\frac{\varepsilon}{1-\alpha}$, since $\varepsilon \leqslant 1/k^2$. We now use Lemma 5.5.2 to get the required result. ■

# Bibliography

[ABG06] Eric Angel, Evripidis Bampis, and Laurent Gourvès. *Approximation algorithms for the bi-criteria weighted max-cut problem*. Discrete Applied Mathematics, 154(12):1685 – 1692, 2006. doi:http://dx.doi.org/10.1016/j.dam.2006.02.008. 15

[ABG12] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. *Better balance by being biased: A 0.8776-approximation for max bisection*. CoRR, abs/1205.0458, 2012. 16, 20, 21, 39, 53

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems*. J. ACM, 45(3):501–555, May 1998. (Preliminary version in *33rd FOCS*, 1992). eccc:TR98-008, doi:10.1145/278298.278306. 3, 63, 92

[AM09] Per Austrin and Elchanan Mossel. *Approximation resistant predicates from pairwise independence*. computational complexity, 18(2):249–271, Jun 2009. doi:10.1007/s00037-009-0272-6. 96, 98

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of np*. J. ACM, 45(1):70–122, January 1998. doi:10.1145/273865.273901. 3, 63, 92

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. *Free bits, PCPs, and nonapproximability—towards tight results*. SIAM J. Computing, 27(3):804–915, June 1998. (Preliminary version in *36th FOCS*, 1995). eccc:TR95-024, doi:10.1137/S0097539796302531. 92, 95

[BKS15] Amey Bhangale, Swastik Kopparty, and Sushant Sachdeva. *Simultaneous approximation of constraint satisfaction problems*. In *International Colloquium on Automata, Languages, and Programming*, pages 193–205. Springer, 2015. 6, 15, 16, 17, 19, 20, 21, 22

[Cha13] Siu On Chan. *Approximation resistance from pairwise independent subgroups*. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 447–456. ACM, New York, NY, USA, 2013. doi:10.1145/2488608.2488665. 96

[DH13] Irit Dinur and Prahladh Harsha. *Composition of low-error 2-query PCPs using decodable PCPs*. SIAM J. Computing, 42(6):2452—-2486, 2013. (Preliminary version in *51st FOCS*, 2009). eccc:TR09-042, doi:10.1137/100788161. 6, 66, 67, 71, 74, 76, 77, 78, 80, 81, 82, 84

[DS14] Irit Dinur and David Steurer. *Analytical approach to parallel repetition*. In *Proc.* 46*th ACM Symp. on Theory of Computing (STOC)*, pages 624–633. 2014. arXiv:1305.1979. 64, 65, 68, 85

[EH08] Lars Engebretsen and Jonas Holmerin. *More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP*. Random Structures & Algorithms, 33(4):497–514, 2008. 95

[FL92] Uriel Feige and László Lovász. *Two-prover one-round proof systems: Their power and their problems (extended abstract)*. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 733–744. ACM, New York, NY, USA, 1992. doi:10.1145/129712.129783. 14

[Gol11] Oded Goldreich. *A sample of samplers: A computational perspective on sampling*. In Oded Goldreich, ed., *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *LNCS*, pages 302–332. Springer, 2011. eccc:1997/TR97-020. 78

[GRW11] Christian Glasser, Christian Reitwiessner, and Maximilian Witek. *Applications of discrepancy theory in multiobjective approximation*. CoRR, abs/1107.0634, 2011. 22

[GW95] MICHEL X. GOEMANS and DAVID P. WILLIAMSON. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*. J. ACM, 42(6):1115–1145, November 1995. doi:10.1145/227683.227684. 6, 14, 15

[Hås01] JOHAN HÅSTAD. *Some optimal inapproximability results*. J. ACM, 48(4):798–859, July 2001. (Preliminary version in *29th STOC*, 1997). doi:10.1145/502090.502098. 6, 14, 62, 63, 92, 95

[Hås14] JOHAN HÅSTAD. *On the NP-hardness of Max-Not-2*. SIAM Journal on Computing, 43(1):179–193, 2014. 95, 100

[HK05] JOHAN HÅSTAD and SUBHASH KHOT. *Query Efficient PCPs with Perfect Completeness.* Theory of Computing, 1(1):119–148, 2005. 95

[Hua13] SANGXIA HUANG. *Approximation resistance on satisfiable instances for predicates with few accepting inputs*. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 457–466. ACM, 2013. 93, 96

[HV02] JOHAN HÅSTAD and SISTLA VENKATESH. *On the advantage over a random assignment*. Random Structures and Algorithms, 25:117 – 149, 05 2002. doi:10.1002/rsa.20031. 61

[HW03] JOHAN HÅSTAD and AVI WIGDERSON. *Simple analysis of graph tests for linearity and PCP*. Random Structures & Algorithms, 22(2):139–160, 2003. 95

[Kho01] SUBHASH KHOT. *Improved inapproximability results for maxclique, chromatic number and approximate graph coloring*. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 600–609. 2001. doi:10.1109/SFCS.2001.959936. 64, 67, 86

[Kho02] SUBHASH KHOT. *On the power of unique 2-prover 1-round games*. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 767–775. ACM, New York, NY, USA, 2002. doi:10.1145/509907.510017. 92, 96

[Kho10] SUBHASH KHOT. *Inapproximability results for computational problems on lattices*, volume 10 of *Information Security and Cryptography*, pages 453–473. Springer International Publishing, 2010. doi:10.1007/978-3-642-02295-1_14. 65

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. *Optimal inapproximability results for max-cut and other 2-variable csps?* SIAM J. Comput., 37(1):319–357, April 2007. doi:10.1137/S0097539705447372. 14, 15

[KN07] Subhash Khot and Assaf Naor. *Linear equations modulo 2 and the L1 diameter of convex bodies*. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 318–328. 2007. doi:10.1109/FOCS.2007.20. 62

[KS06] Subhash Khot and Rishi Saket. *A 3-query non-adaptive PCP with perfect completeness*. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 11–pp. IEEE, 2006. 95

[KTW14] Subhash Khot, Madhur Tulsiani, and Pratik Worah. *A characterization of strong approximation resistance*. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 634–643. ACM, 2014. 96

[MOO05] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. *Noise stability of functions with low influences: invariance and optimality*. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 21–30. IEEE, 2005. 14, 114

[Mos08] Elchanan Mossel. *Gaussian bounds for noise correlation of functions and tight analysis of long codes*. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 156–165. Oct 2008. doi:10.1109/FOCS.2008.44. 111, 112, 113, 114

[MR08] D. Moshkovitz and R. Raz. *Two query pcp with sub-constant error*. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 314–323. Oct 2008. doi:10.1109/FOCS.2008.60. 6, 63, 64, 65, 66, 68

[Ole03] Krzysztof Oleszkiewicz. *On a nonsymmetric version of the khinchine-kahane inequality*. In *Stochastic inequalities and applications*, pages 157–168. Springer, 2003. 113

[OW08] Ryan O'Donnell and Yi Wu. *An optimal sdp algorithm for max-cut, and equally optimal long code tests*. In *Proceedings of the Fortieth Annual ACM Symposium on*

*Theory of Computing*, STOC '08, pages 335–344. ACM, New York, NY, USA, 2008. doi:10.1145/1374376.1374425. 14

[OW09a] Ryan O'Donnell and Yi Wu. *3-Bit Dictator Testing: 1 vs. 5/8*, pages 365–374. Society for Industrial and Applied Mathematics, 2009. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611973068.41, doi:10.1137/1.9781611973068.41. 95, 98

[OW09b] Ryan O'Donnell and Yi Wu. *Conditional hardness for satisfiable 3-CSPs*. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 493–502. ACM, 2009. 95

[PRS02] Michal Parnas, Dana Ron, and Alex Samorodnitsky. *Testing basic Boolean Formulae*. SIAM Journal on Discrete Mathematics, 16(1):20–46, 2002. 92, 95

[Rag08] Prasad Raghavendra. *Optimal algorithms and inapproximability results for every CSP?* In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 245–254. ACM, New York, NY, USA, 2008. doi:10.1145/1374376.1374414. 96

[Raz98] Ran Raz. *A parallel repetition theorem*. SIAM J. Comput., 27(3):763–803, June 1998. doi:10.1137/S0097539795280895. 63, 92

[RT12] Prasad Raghavendra and Ning Tan. *Approximating csps with global cardinality constraints using sdp hierarchies*. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012. 16, 18, 20, 34, 37, 38, 39, 40

[Sjo09] Henrik Sjogren. *Rigorous analysis of approximation algorithms for max 2-csp*. Master's thesis, 2009. 21

[ST00] Alex Samorodnitsky and Luca Trevisan. *A PCP characterization of NP with optimal amortized query complexity*. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 191–199. ACM, 2000. 95

[ST09] Alex Samorodnitsky and Luca Trevisan. *Gowers uniformity, influence of variables, and PCPs*. SIAM Journal on Computing, 39(1):323–360, 2009. 95

[TY15] Suguru Tamaki and Yuichi Yoshida. *A query efficient non-adaptive long code test with perfect completeness*. Random Structures & Algorithms, 47(2):386–406, 2015. 7, 94, 96, 98, 99, 116

[Wol07] Pawel Wolff. *Hypercontractivity of simple random variables*. Studia Mathematica, 180(3):219–236, 2007. 113

[Zwi97] Uri Zwick. *Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables per Constraint*. In *In Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1997. 95