KTH Computer Science
and Communication

Hardness of Constraint Satisfaction and Hypergraph Coloring

Constructions of Probabilistically Checkable Proofs with Perfect Completeness

SANGXIA HUANG

Doctoral Thesis
Stockholm, Sweden 2015

TRITA-CSC-A-2015:13
ISSN-1653-5723
ISRN-KTH/CSC/A--15/13-SE
ISBN 978-91-7595-633-6

KTH CSC
Skolan för datavetenskap och kommunikation
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi fredagen den 4 september 2015 klockan 14.00 i F3, Lindstedtsvägen 26, Kungliga Tekniska högskolan, Stockholm.

Tryck: Universitetsservice US AB

Abstract

A Probabilistically Checkable Proof (PCP) of a mathematical statement is a proof written in a special manner that allows for efficient probabilistic verification. The celebrated PCP Theorem states that for every family of statements in NP, there is a probabilistic verification procedure that checks the validity of a PCP proof by reading only 3 bits from it. This landmark theorem, and the works leading up to it, laid the foundation for many subsequent works in computational complexity theory, the most prominent among them being the study of inapproximability of combinatorial optimization problems.

This thesis focuses on a broad class of combinatorial optimization problems called Constraint Satisfaction Problems (CSPs). In an instance of a CSP problem of *arity* k, we are given a set of variables taking values from some finite domain, and a set of constraints each involving a subset of at most k variables. The goal is to find an assignment that simultaneously satisfies as many constraints as possible. An alternative formulation of the goal that is commonly used is Gap-CSP, where the goal is to decide whether a CSP instance is *satisfiable* or *far from satisfiable*, where the exact meaning of being far from satisfiable varies depending on the problems.

We first study Boolean CSPs, where the domain of the variables is $\{0, 1\}$. The main question we study is the hardness of distinguishing satisfiable Boolean CSP instances from those for which no assignment satisfies more than some ε fraction of the constraints. Intuitively, as the arity increases, the CSP gets more complex and thus the hardness parameter ε should decrease. We show that for Boolean CSPs of arity k, it is NP-hard to distinguish satisfiable instances from those that are at most $2^{\widetilde{O}(k^{1/3})}/2^k$-satisfiable.

We also study coloring of graphs and hypergraphs. Given a graph or a hypergraph, a coloring is an assignment of colors to vertices, such that all edges or hyperedges are non-monochromatic. The gap problem is to distinguish instances that are colorable with a small number of colors, from those that require a large number of colors. For graphs, we prove that there exists a constant $K_0 > 0$, such that for any $K \geq K_0$, it is NP-hard to distinguish K-colorable graphs from those that require $2^{\Omega(K^{1/3})}$ colors. For hypergraphs, we prove that it is quasi-NP-hard to distinguish 2-colorable 8-uniform hypergraphs of size N from those that require $2^{(\log N)^{1/4-o(1)}}$ colors.

In terms of techniques, all these results are based on constructions of PCPs with perfect completeness, that is, PCPs where the probabilistic proof verification procedure always accepts a correct proof. Not only is this a very natural property for proofs, but it can also be an essential requirement in many applications. It has always been particularly challenging to construct PCPs with perfect completeness for NP statements due to limitations in techniques. Our improved hardness results build on and extend many of the current approaches. Our Boolean CSP result and GraphColoring result were proved by adapting the Direct Sum of PCPs idea by Siu On Chan to the perfect completeness setting. Our proof for hypergraph coloring hardness improves and simplifies the recent work by Khot and Saket, in which they proposed the notion of superposition complexity of CSPs.

Sammanfattning

Ett probabilistiskt verifierbart bevis (eng: Probabilistically Checkable Proof, PCP) av en matematisk sats är ett bevis skrivet på ett speciellt sätt vilket möjliggör en effektiv probabilistisk verifiering. Den berömda PCP-satsen säger att för varje familj av påståenden i NP finns det en probabilistisk verifierare som kontrollerar om en PCP bevis är giltigt genom att läsa endast 3 bitar från det. Denna banbrytande sats, och arbetena som ledde fram till det, lade grunden för många senare arbeten inom komplexitetsteorin, framförallt inom studiet av approximerbarhet av kombinatoriska optimeringsproblem.

I denna avhandling fokuserar vi på en bred klass av optimeringsproblem i form av villkorsuppfyllningsproblem (engelska "Constraint Satisfaction Problems" CSPs). En instans av ett CSP av *aritet* k ges av en mängd variabler som tar värden från någon ändlig domän, och ett antal villkor som vart och ett beror på en delmängd av högst k variabler. Målet är att hitta ett tilldelning av variablerna som samtidigt uppfyller så många som möjligt av villkoren. En alternativ formulering av målet som ofta används är Gap-CSP, där målet är att avgöra om en CSP-instans är *satisfierbar* eller *långt ifrån satisfierbar*, där den exakta innebörden av att vara "långt ifrån satisfierbar" varierar beroende på problemet.

Först studerar vi booleska CSPer, där domänen är $\{0, 1\}$. Den fråga vi studerar är svårigheten av att särskilja satisfierbara boolesk CSP-instanser från instanser där den bästa tilldelningen satisfierar högst en andel ε av villkoren. Intuitivt, när ariten ökar blir CSP mer komplexa och därmed bör svårighetsparametern ε avta med ökande aritet. Detta visar sig vara sant och ett första resultat är att för booleska CSP av aritet k är det NP-svårt att särskilja satisfierbara instanser från dem som är högst $2^{\widetilde{O}(k^{1/3})}/2^k$-satisfierbara.

Vidare studerar vi färgläggning av grafer och hypergrafer. Givet en graf eller en hypergraf, är en färgläggning en tilldelning av färger till noderna, så att ingen kant eller hyperkant är monokromatisk. Problemet vi analyserar är att särskilja instanser som är färgbara med ett litet antal färger från dem som behöver många färger. För grafer visar vi att det finns en konstant $K_0 > 0$, så att för alla $K \geq K_0$ är det NP-svårt att särskilja grafer som är K-färgbara från dem som kräver minst $2^{\Omega(K^{1/3})}$ färger. För hypergrafer visar vi att det är kvasi-NP-svårt att särskilja 2-färgbara 8-likformiga hypergrafer som har N noder från dem som kräv minst $2^{(\log N)^{1/4-o(1)}}$ färger.

Samtliga dessa resultat bygger på konstruktioner av PCPer med perfekt fullständighet. Det vill säga PCPer där verifieraren alltid accepterar ett korrekt bevis. Inte bara är detta en mycket naturlig egenskap för PCPer, men det kan också vara ett nödvändigt krav för vissa tillämpningar. Konstruktionen av PCPer med perfekt fullständighet för NP-påståenden ger tekniska komplikationer och kräver delvis utvecklande av nya metoder. Vårt booleska CSPer resultat och vårt GraphColoring resultat bevisas genom att anpassa "Direktsumman-metoden" introducerad av Siu On Chan till fallet med perfekt fullständighet. Vårt bevis för hypergraffärgningssvårighet förbättrar och förenklar ett färskt resultat av Khot och Saket, där de föreslog begreppet superpositionskomplexitet av CSP.

Acknowledgments

I am greatly indebted to my advisor Johan Håstad. Despite his very busy schedules, Johan always seemed to have an infinite amount of time to listen to my buggy ideas, give me insightful comments, help me understand difficult concepts, and encourage me to go after hard problems. It has been an incredible experience learning from Johan, and I couldn't have asked for a better *handledare*.

During my studies, I had the opportunity of visiting different institutes and collaborating with many amazing researchers. I want to thank Ryan O'Donnell for hosting me at CMU during the fall of 2013, and later at Boğaziçi in Istanbul. It has been a great pleasure getting to know him, and it is hard not to be influenced by his passion. I am also grateful to many other people at CMU with whom I spent much time discussing research questions and also enjoying Pittsburgh: Anupam Gupta, Venkatesan Guruswami, Aravindan Vijayaraghavan, David Witmer, and John Wright. There was this once we went to PNC Park to watch a Pirates' game, and I was greatly amused by the part of the game when people dressed in potato, onion, bacon (there were probably a few more things, a burger maybe?) racing around the field and trying to knock each other out. I consider that a vivid lesson on American baseball. I am also thankful to Madhur Tulsiani for an enjoyable internship at Toyota Technological Institute at Chicago.

From the Theory Group at KTH, I wish to thank past and present members of the algorithm and approximability group for many valuable discussions: Per Austrin, Adam Schill Collberg, Johan Håstad, Michael Lampis, Rajsekar Manokaran, Tobias Mömke, Lukáš Poláček, Ola Svensson, Li-Yang Tan, and Cenny Wenner. I have also had a great amount of fun sharing room 1445 with Gunnar Kreitz and Lukáš Poláček. Many thanks also to Musard Balliu, Emma Enström, Pedro de Carvalho Gomes, Benjamin Greschbach, Massimo Lauria, Mladen Mikša, Hamed Nemati, Jakob Nordström, Oliver Schwarz, Siavash Soleimanifard and Marc Vinyals — for all the lunch discussions, cycling and hiking trips, fikas, football games, and many more.

I am very grateful to Per Austrin and Johan Håstad for taking the time to look at the preliminary versions of this thesis, and suggesting numerous improvements to the English and Swedish parts of this thesis.

And above all, I thank mum and dad for their unconditional support over the years, without which none of this would have been possible.

Contents

Part I

Introduction

见一叶落　而知岁之将暮
睹瓶中之冰　而知天下之寒

—— 淮南子·说山训

Chapter 1

Overview

Let us start by introducing one of the recurring computation problems in this thesis — GraphColoring: given a natural number $k \geq 2$, and a graph $G = (V, E)$, where V is the set of vertices, E is the set of edges, and we want to color each node with a color. Is it possible to color G with at most k colors such that every edge in E has its two endpoints colored with different colors?

When k is considered to be a constant rather than part of the input, we usually denote the decision problem by k-Coloring.

Problems related to GraphColoring have been studied for centuries. The Four-Color-Problem, one of the most famous mathematical problems, was posed in the mid-19$^{\text{th}}$ century.[1] The original statement of the problem asks that given a map — which we can think of as a plane separated into regions — whether it is possible to color the regions with at most 4 colors, such that no two adjacent regions have the same color, where by "adjacent" we mean that two regions share a non-trivial segment of border. Stated in modern graph theory terminology, the Four-Color-Problem asks whether any planar graph is 4-colorable.

There have been numerous attempts at solving the Four-Color-Problem. In 1976, Kenneth Appel and Wolfgang Haken announced that a proof was found with the help of a computer asserting that the answer to the Four-Color-Problem is "Yes", making it the first major theorem proved using a computer. A number of flaws were subsequently found and fixed. It was highly controversial at the time especially since the proof was impossible for human to verify, and sparked the debate around the question of *"what is a mathematical proof"*. In 1997, Robertson, Sanders, Seymour and Thomas gave a simpler computer proof [94], and in 2005, Georges Gonthier managed to prove the theorem with Coq, a general purpose theorem proving tool.

Despite its origin, the Four-Color-Problem, and GraphColoring in gen-

[1] Möbius mentioned the problem as early as 1840. The first written record of the problem seems to be in a letter from de Morgan in 1852, in which he mentioned that one of his students named Guthrie asked the question.

eral, is probably of little interest to cartographers. Nevertheless, many practical computational problems can be modeled as GraphColoring, making it a very interesting problem for many areas of computer science. We give a simple example.

Example. *Consider a cellular network in a certain region, with a number of base stations that are connected to each other. A pair of base stations are* interfering *if they are close enough that their signals interfere while using the same frequency. Therefore when allocating frequencies, we need to make sure that interfering base stations use disjoint sets of frequencies.*

Mobile devices connect to a base station close to them using a certain frequency channel dynamically assigned by the base station. This enables mobile devices to communicate with each other through base stations.

Each base station also has a specified traffic demand, which is the minimum number of simultaneous connections the base station need to be able to handle. Such requirements may be different for different stations, for instance, it may be higher in dense urban areas and lower in sparsely populated areas.

Suppose the entire available spectrum is divided into k frequency channels, and our task is to allocate sets of frequencies to the base stations. Each base station then assigns frequencies to mobile connections from the set of frequencies allocated to it. Naturally, we would like to know whether there is an allocation that satisfies the connection requirements without creating interference.

To formulate this as a GraphColoring *problem, let us number the base stations with $1, 2, \ldots, N$. For station i, let d_i be the minimum number of simultaneous channels required. We construct a graph with $\sum_{i=1}^{N} d_i$ vertices. Station i corresponds to d_i vertices that are connected with each other. For $i \neq j$, if signals from station i and station j interfere, then we add $d_i \cdot d_j$ edges between all pairs of vertices corresponding to station i and station j. A k-coloring of this graph corresponds to a way to allocate the k frequency channels without introducing interference.*

1.1 Computational Complexity

A straightforward procedure to solve GraphColoring is simply to try out all possibilities. This *brute force search* solution is unfortunately not very practical. Even in the extremely moderate setting where we have 100 vertices and 3 colors, we still potentially need to try $3^{100} \approx 5 \times 10^{47}$ possibilities. It was estimated in [56] that, as of 2011, the total computing power of the whole world is about 6.4×10^{18} instructions per second.[2] This means that it will take around 2.5×10^{21} years to figure out by brute force search whether a single instance of a graph with 100 vertices is 3-colorable.

GraphColoring is only one example of a very broad class of computational problems called *Constraint Satisfaction Problems (*CSP*s)*. Roughly speaking, in a

[2]The authors of [56] arrived at this number by estimating the installation numbers of supercomputers, PCs, mobile devices, and so on. Human brain power or other potential computing power was not included.

CSP, we have some variables, such as the colors of the vertices, and some simple constraints that involve a small number of variables, such as the colors of the two endpoints of an edge being different. Many problems we encounter are, at their core, CSPs.

There has been remarkable progress in solving different kinds of CSPs, and powerful tools such as convex optimization are now available. However, fundamental problems such as GraphColoring still seems beyond reach. The ultimate goal of research in computational complexity is to understand the *intrinsic hardness*[3] of computational problems in terms of the amount of resources needed to solve them. The complexity measures we are interested in include time, storage space, randomness, query access, and many others, and the goal is to understand how the resource requirement increases as the size of the instance increases. The focus of this thesis is time complexity.

1.2 Reduction, Complexity Classes, and NP

Researchers in computational complexity have been quite successful in *classifying* computational problems into complexity classes according to their hardness. The central notion here is *reduction*.[4] An *efficient*[5] reduction from computational problem A to computational problem B is an algorithm that solves A by calling a *small* number of times a subroutine that solves problem B, and spends a *small* amount of time outside those subroutines. Therefore, if there is an efficient algorithm that solves B, then we automatically get an efficient algorithm that solves A.

A problem A is *complete* for a class $\mathcal{C}$ of problems, or $\mathcal{C}$-complete, if A is in $\mathcal{C}$, and all problems in $\mathcal{C}$ can be reduced to A.

Among all complexity classes, the most notable ones are undoubtedly P and NP. The class P contains all computational problems for which there is a polynomial time algorithm that correctly decides the answer. The class NP contains all computational problems for which there is a polynomial-time verifiable *proof*. In other words, a problem A is in NP if and only if there exists a polynomial time algorithm V that we refer to as the *verifier*, such that

- If an instance x is a "Yes" instance, then there is a string π, such that V answers "Yes" on input (x, π). In this case, we say that π is a *proof*[6] of x being a "Yes" instance of problem A.
- If an instance x is a "No" instance, then for any π, V always answers "No" given x and π as input.

[3]The word "intrinsic" refers to those measures that are not qualitatively affected by either the computational model or the way problem instances are represented, so long as they are "reasonable". The computational model considered in this thesis are Turing machines.

[4]Turing reduction, to be precise.

[5]The exact meaning of "efficient" depends on the settings of the problems, so we do not go into details here. The same applies to "a small amount of time" later in the text.

[6]Also commonly known as witness or certificate.

The verifier V that expects π to be a 3-coloring of the input graph, and checks if it is a valid 3-coloring is a verifier that satisfies the above requirement, and thus 3-Coloring is in NP. It is also known that 3-Coloring is NP-complete.

Clearly every problem in P is also in NP. The P vs. NP question asks whether every problem in NP is in P. If indeed P = NP, then every problem that has efficiently verifiable proofs also has efficient algorithms. The general consensus is that P $\neq$ NP, since it always seems much harder to come up with proofs for mathematical statements than to verify whether given proofs are correct.

1.3 Probabilistically Checkable Proofs

Observe that in order to verify the certificate π for 3-Coloring that we give above, it is necessary to check the entirety of π. This is also the case in proofs in mathematics: to verify whether a proof is correct, we need to go through the proof line by line, because an erroneous inference in a single step invalidates the whole proof. Although this can be tedious at times, we do have the strong guarantee that if a proof is correct, then it will *always* be accepted, and if a proof is not correct, then it will *never* be accepted.

What if we allow the verifier to accept an incorrect proof with some small probability? This may sound outrageous at first, but in light of the proof of the Four-Color-Theorem, and other theorems proved with computer assistance such as the classification of all finite simple groups, it seems that allowing some probability of error might offer a way out, if in exchange we are able to do proof verification more efficiently. Goldwasser, Micali and Rackoff [41], and independently Babai [11], introduced randomness and interaction to the procedure of proof verification and started the study of interactive proofs.

Again let's take 3-Coloring as an example. There are two parties in this interactive setting — the prover and the verifier, and there is some pre-designed verification protocol. Given any graph G, the almighty and unscrupulous prover always attempts to convince the verifier that G is 3-colorable. The verifier usually only has limited computational resources, so the only hope not to get fooled too often is to design a "robust" protocol.

This turns out to be a very important generalization. Subsequent works in this area demonstrated the power of interactive proofs, fundamentally changed our understanding of the notion of a proof, and gave rise to fascinating discoveries in computational complexity, such as the PCP Theorem, IP = PSpace, and zero-knowledge proofs.

The idea behind all results in this thesis is to construct Probabilistically Checkable Proofs, or PCPs. In this setting, the verifier is given a proof of some statement — say, some graph is 3-colorable — and is only allowed to decide whether to accept or reject the proof by looking at a very small portion of the proof. The verifier's access to the original instance is not limited. The celebrated PCP Theorem says that for every problem A in NP, there is a probabilistic polynomial time verifier V

that takes a proof π that is a string of 0/1 bits whose length is polynomial in the size of the instances, reads 3 bits from it, always answers "YES" if the instance is a YES instance, and answers "NO" with constant probability if the instance is a NO instance.

Such PCPs can be viewed naturally as an optimization problem: for some problem A in NP and a verifier V, given an instance x, find a proof π that maximizes the probability that V accepts (x, π). Indeed, the notion of Probabilistically Checkable Proofs as well as the PCP Theorem has become an indispensable tool in the study of the limit of efficient approximation algorithms for combinatorial optimization problems.

Below we describe some parameters of PCPs that are important for applications in hardness of approximation.

Randomness The number of random bits used by V. This is also directly related to the size of the proof that is given to V.

Alphabet Size The alphabet the proof π is written in. The proof π is not always limited to 0/1 strings. When the alphabet size is larger, each symbol in the proof potentially gives the verifier more information, therefore a larger alphabet size could be helpful when the goal is to optimize other parameters.

Completeness The probability that V accepts a correct proof. If V always accepts a correct proof, then we say that it has *perfect completeness*.

Soundness The probability that V accepts an incorrect proof.

Query Complexity The number of symbols V needs to read from a proof. The requirement depends on the actual applications, but is typically either a constant or some function that grows slowly as instance size grows.

Amortized Query Complexity Let c and s be the completeness and the soundness parameters, and let q be the query complexity. The amortized query complexity is $q/\log(c/s)$. When $c = 1$, this measures how much on average each additional query decreases the probability that the verifier accepts a wrong proof.

We refer to the work by Bellare, Goldreich and Sudan [16] for the importance of these parameters.

1.4 Contribution of This Thesis

In this thesis, we construct PCPs that have perfect completeness. Having perfect completeness makes it easier to compose a PCP with other reductions, and thus is a desirable — sometimes even necessary — property in many applications.

This thesis has two main parts. Part II focuses on CSPs on Boolean variables, and we construct several PCPs with Boolean alphabet. In Chapter 4, we construct

PCPs for NP, assuming the d-to-1 Conjectures. For any integer $k \geq 4$, we give a PCP that uses Boolean alphabet, has query complexity k, perfect completeness and soundness $\frac{1}{2} + \frac{1}{2^k}$. In Chapter 5, the focus is on the relation between soundness and query complexity. For a k-query non-adaptive PCP, the best soundness we could hope for is $O(k)/2^k$. We give a construction that achieves $2^{\widetilde{O}(k^{1/3})}/2^k$, improving the best previous soundness of $2^{O(k^{1/2})}/2^k$ by Håstad and Khot [55].

Part III studies graph and hypergraph coloring. The alphabet size of the PCPs are exactly the number of colors we can use, the query complexity corresponds to the size of the edges in the graphs or hypergraphs. Chapter 7 gives improved hardness for graph coloring, and Chapter 8 shows new hardness results for 2-coloring 8-uniform hypergraphs.

We review some mathematical tools in Chapter 2, and give a more detailed description of CSP, PCP and hardness of approximation in Chapter 3.

Some of the results in this thesis have appeared previously in different forms. Chapter 5 is based on the paper "Approximation Resistance on Satisfiable Instances for Sparse Predicates" [62] in the journal Theory of Computing, and the conference version of this paper [60] appeared in the 45^{th} ACM Symposium on the Theory of Computing, in 2013. Chapter 7 is based on the paper "Improved Hardness of Approximating Chromatic Number" [61], which appeared in APPROX 2013.

Chapter 4 is based on a technical report on ECCC [59], and Chapter 8 is more recent and has not yet been published else-where.

Chapter 2

Mathematical Background

We introduce some notations, and review the mathematical tools that are useful for the rest of the thesis. We recall some standard notations and conventions in Section 2.1. This is followed by a review of basic probability theory in Section 2.2, some algebra in Section 2.3, and an introduction to discrete Fourier analysis in Section 2.4. We conclude this chapter with a quick overview of Long-Code, Low-Degree-Long-Code and Hadamard-Code in Section 2.5.

2.1 Basic Notations

We use the following notations for sets that are frequently used.

$\mathbb{Z}$	The integers
$\mathbb{N}$	The natural numbers $\{n \in \mathbb{Z} \mid n \geq 0\}$
$\mathbb{N}^+$	The natural numbers $\{n \in \mathbb{Z} \mid n \geq 1\}$
$[n]$	The set of integers from 1 to n $\{1, 2, \ldots, n\}$
$\mathbb{R}$	The real numbers
$\mathbb{C}$	The complex numbers
$\mathbb{F}_q$	The finite field with q elements, for some prime power q

For two sets X and Y, we use X^Y to denote the set of all vectors with elements from X indexed by Y. There is a bijection between vectors in X^Y and functions $f : Y \to X$, and we make no distinction between these two notation. When $X = [2]$, the set X^Y can also be viewed as the family of all subsets of Y, also known as the power set of Y, denoted as $\mathcal{P}(Y)$. When $Y = [n]$, we write X^n instead of $X^{[n]}$.

For a vector $v \in X^Y$ and a set $S \subseteq Y$, we use $v|_S \in X^S$ to denote the restriction of v on S.

For a statement P, the Iverson bracket notation is defined as

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true;} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathbb{F}$ be a field, X an index set, and $u, v \in \mathbb{F}^X$ be two vectors. The *dot product* $\langle u, v \rangle$ defined as

$$\langle u, v \rangle = \sum_{x \in X} u_x \times v_x \,.$$

If the vectors are over a Euclidean space, the dot product is also called *inner product.*

2.2 Probability Theory

In this thesis, we work exclusively with probability spaces with finite sample spaces.

Let (Ω, μ) be a probability space. The support of the space $\text{supp}(\Omega, \mu) = \{x \in \Omega \mid \mu(x) > 0\}$ contains all samples from Ω with non-zero probability under μ.

A random variable over (Ω, μ) is a function $f : \Omega \to \mathbb{R}$. We define the expectation of function f over (Ω, μ) as

$$\mathbf{E}[f] = \sum_{x \in \Omega} f(x)\mu(x)\,,$$

and the variance of f as

$$\mathbf{Var}[f] = \mathbf{E}[f^2] - \mathbf{E}[f]^2\,.$$

For $1 \leq p < \infty$, the l_p norm of f is defined as

$$\|f\|_p = \left(\mathbf{E}[|f|^p]\right)^{1/p}\,.$$

We define the infinity norm l_∞ as

$$\|f\|_\infty = \max_{x : \mu(x) > 0} |f(x)|\,.$$

Example 2.1. *Let (Ω, μ) be the probability space where $\Omega = [N]$ for some integer N, and μ is the uniform distribution over Ω.*

Let $f : \Omega \to \mathbb{R}$ be the function that evaluates to 1 everywhere on Ω. Then

$$\|f\|_1 = 1, \quad \|f\|_2 = 1, \quad \|f\|_4 = 1, \ \ldots$$

Define $g : \Omega \to \mathbb{R}$ as follows

$$g(x) = \begin{cases} N & \text{if } x = 1; \\ 0 & \text{otherwise.} \end{cases}$$

Then g has norms

$$\|g\|_1 = 1, \quad \|g\|_2 = \sqrt{N}, \quad \|g\|_4 = N^{3/4}, \ \ldots$$

We can see that although f and g has the same l_1 norm, their l_2, l_4 and all other norms behave very differently.

We use $L^2(\Omega, \mu)$ to denote the set of all functions $f : \Omega \to \mathbb{R}$ such that $\|f\|_2 < \infty$. Since in our setting Ω is finite, $L^2(\Omega, \mu)$ is simply the set of all functions $f : \Omega \to \mathbb{R}$. The inner-product on $L^2(\Omega, \mu)$ is defined as

$$\langle f, g \rangle_\mu = \mathop{\mathbf{E}}_{x \in (\Omega, \mu)} [f(x) \cdot g(x)] .$$

Theorem 2.2 (Hölder's Inequality)**.** *Let $1 \leq p \leq q \leq \infty$ be such that $1/p + 1/q = 1$ (we adopt the convention that when $q = \infty$, $1/q = 0$). Then for every $f, g \in L^2(\Omega, \mu)$*

$$\langle f, g \rangle \leq \|f\|_p \cdot \|g\|_q .$$

Taking $p = q = 2$, we have the following Cauchy-Schwarz Inequality.

Theorem 2.3 (Cauchy-Schwarz' Inequality)**.** *For every $f, g \in L^2(\Omega, \mu)$, we have*

$$\langle f, g \rangle \leq \|f\|_2 \cdot \|g\|_2 .$$

The following generalization of Hölder's Inequality is also useful in some cases, and can be proved using mathematical induction and Hölder's Inequaltiy.

Theorem 2.4 (Generalized Hölder's Inequality)**.** *Assume that $1 \leq p_1, \ldots, p_t \leq \infty$, such that*

$$\sum_{i=1}^{t} \frac{1}{p_i} = 1 .$$

Then for all $f_1, \ldots, f_t \in L^2(\Omega, \mu)$

$$\left\| \prod_{i=1}^{t} f_i \right\|_1 \leq \prod_{i=1}^{t} \|f_i\|_{p_i} .$$

In many parts of this thesis, we deal with the sample spaces that have the form Ω^n, and the distribution is a product distribution $\mu = \bigotimes_{i=1}^{n} \mu_i$, where for each i, μ_i is a distribution over Ω_i. We call these *product spaces.*

We also study correlated probability spaces. Let $(\Omega \times \Psi, \mu)$ be a probability space. We say that Ω and Ψ are correlated spaces. The notion of correlation for correlated probability spaces was introduced by Mossel [86].

Definition 2.5 ([86])**.** *Let $(\Omega \times \Psi, \mu)$ be a correlated probability space, μ is a distribution on the finite product set $\Omega \times \Psi$ and that the marginals of μ on Ω and Ψ have full support. Define the correlation between Ω and Ψ to be*

$$\rho(\Omega, \Psi; \mu) = \max_{\substack{f:\Omega \to \mathbb{R} \\ g:\Psi \to \mathbb{R}}} \{ |\mathbf{E}[fg]| \mid \mathbf{E}[f] = 0, \mathbf{E}[f^2] \leq 1, \mathbf{E}[g] = 0, \mathbf{E}[g^2] \leq 1 \} ,$$

where the expectation $\mathbf{E}[fg]$ is under μ, and $\mathbf{E}[f]$, $\mathbf{E}[f^2]$, $\mathbf{E}[g]$ and $\mathbf{E}[g^2]$ are under marginals of μ on corresponding spaces.

A useful fact for bounding correlation of probability spaces from [86] is that the correlation of a product of correlated probability space is equal to the maximum correlation among the individual correlated spaces (excluding empty components).

Lemma 2.6 ([86])**.** *Let* $\{(\Omega_i \times \Psi, \mu_i)\}$ *be a set of correlated probability spaces, then*

$$\rho\left(\prod_i \Omega_i, \prod_i \Psi_i; \prod_i \mu_i\right) \leq \max_i \rho(\Omega_i, \Psi_i; \mu_i)\,.$$

The following is a useful condition for the correlation being strictly smaller than 1 and is from [86].

Lemma 2.7 ([86])**.** *Let* $\{\Omega \times \Psi, \mu\}$ *be two correlated spaces such that the smallest probability in* $\mathrm{supp}(\mu)$ *is at least* $\alpha > 0$*. Define a bipartite graph* $G = (\Omega, \Psi, E)$ *where* $(a, b) \in \Omega \times \Psi$ *satisfies* $(a, b) \in E$ *if* $\mu(a, b) > 0$*. If* G *is connected, then*

$$\rho(\Omega, \Psi; \mu) \leq 1 - \alpha^2/2\,.$$

We also need the following lemma when analyzing correlations. Intuitively, if we can decompose μ into a convex combination of two distributions and we can bound the correlation between Ω and Ψ in both sub-distributions by some constant c, then barring special cases it seems reasonable that the correlation $\rho(\Omega, \Psi; \mu)$ should also be bounded by some function of c. More formally, we have the following lemma.

Lemma 2.8 ([104])**.** *Let* $(\Omega \times \Psi, \delta\nu + (1-\delta)\nu')$ *be a correlated space such that the marginal distribution of at least one of* Ω *and* Ψ *is identical on both* ν *and* ν'*. Then*

$$\rho(\Omega, \Psi; \delta\nu + (1-\delta)\nu') \leq \sqrt{\delta\rho(\Omega, \Psi; \nu)^2 + (1-\delta)\rho(\Omega, \Psi; \nu')^2}\,.$$

Finally, we recall the notion of k-wise independence.

Definition 2.9. *Let* η *be some probability distribution over* Ω*. A product space* (Ω^n, μ) *is* k-wise independent with marginals η*, if for every subset* $S \subseteq [n]$*, where* $|S| \leq k$*, we have that* $\mu|_S = \eta^{\otimes S}$*.*

If η *is the uniform distribution over* Ω*, we say that* (Ω^n, μ) *is* balanced k-wise independent*.*

We say that a set $P \subseteq \Omega^n$ supports a balanced k-wise independent distribution *if there exists some* k*-wise independent distribution* ν *on* Ω^n*, such that* $\mathrm{supp}(\nu) \subseteq P$*.*

2.3 Algebra

In this section, we review some algebraic tools that are useful. We assume some familiarity with elementary linear algebra.

Denote by I_m the identity matrix of order m. When the size is clear from context, we drop the subscript and simply write I.

A real matrix Q is *orthogonal* if $Q^TQ = QQ^T = I$. A set of vectors $\{v_1, \dots, v_n\}$ is *orthonormal* if for all $i, j \in [n]$, $\langle v_i, v_j \rangle = [i = j]$.

Let $A \in \mathbb{R}^{m \times m}$ be a matrix. We say that vector $v \in \mathbb{R}^m$, $v \neq 0$ is an *eigenvector* of A if there exists $\lambda \in \mathbb{R}$, such that $Av = \lambda v$. In this case, we say that λ is an *eigenvalue* of A. The multiset of the eigenvalues of A is called its *spectrum*.

Most matrices we work with in this thesis are symmetric. We have the following theorem regarding the spectrum of symmetric matrices.

Theorem 2.10. *Let $A \in \mathbb{R}^{m \times m}$ be a real symmetric matrix, and let $\lambda_1, \dots, \lambda_m$ be its eigenvalues. Then there exists an orthonormal set of vectors $v_1, \dots, v_m \in \mathbb{R}^m$, such that for all $i \in [m]$, $Av_i = \lambda_i v_i$.*

Fact 2.11. *If $A^T \cdot A = cI$ for some $c \geq 0$, then all eigenvalues of A have absolute value $\sqrt{c}$.*

The following is a standard fact about quadratic forms.

Claim 2.12. *Let $A \in \mathbb{R}^{m \times m}$ be a symmetric real matrix with eigenvalues $\lambda_1, \dots, \lambda_m$. Let $\lambda_{max} = \max_i |\lambda_i|$. Note that by definition $\lambda_{max} \geq 0$. Then for any $u \in \mathbb{R}^m$*

$$\left|u^T A u\right| \leq \lambda_{max} \|u\|_2^2 .$$

Proof. Let $v_1, \dots, v_m$ be a set of orthonormal eigenvectors of A, with corresponding eigenvalues $\lambda_1, \dots, \lambda_m$. Express u under this basis as $v = \sum_{i \in [m]} c_i v_i$. Then using the orthonormality of $v_1, \dots, v_m$, we have

$$\begin{aligned}
u^T A u &= \left(\sum_{i \in [m]} c_i v_i^T \right) A \left(\sum_{i \in [m]} c_i v_i \right) \\
&= \left(\sum_{i \in [m]} c_i v_i^T \right) \left(\sum_{i \in [m]} c_i \lambda_i v_i \right) \\
&= \sum_{i,j \in [m]} \lambda_j c_i c_j v_i^T v_j \\
&= \sum_{j \in [m]} \lambda_j c_j^2 \\
&\leq \lambda_{max} \sum_{j \in [m]} c_j^2 = \lambda_{max} \|u\|_2^2 .
\end{aligned}$$

□

We now turn to polynomials over $\mathbb{F}_2$. For a positive integer m, denote by P_m the vector space of m-variable functions $\mathbb{F}_2^m \to \mathbb{F}_2$. For $f, g \in \mathsf{P}_m$, let $\Delta(f, g)$ be the Hamming distance between f and g. For a subset of functions $\mathcal{F} \subseteq \mathsf{P}_m$, the distance between g and $\mathcal{F}$ is defined as $\Delta(g, \mathcal{F}) = \min_{f \in \mathcal{F}} \Delta(f, g)$.

Definition 2.13. *For $f, g \in \mathsf{P}_m$, their dot product on P_m is defined as $\langle f, g\rangle = \sum_{x\in\mathbb{F}_2^m} f(x)g(x)$.*

Let $\mathcal{A}$ be a subspace of P_m. The *dual space* is defined as $\mathcal{A}^\perp := \{f \in \mathsf{P}_m \mid \forall g \in \mathcal{A}, \langle f, g\rangle = 0\}$.

Denote by $\mathsf{P}_{m,d}$ the space of functions with degree at most d. The following is a well known fact about the dual of $\mathsf{P}_{m,d}$.

Fact 2.14. $\mathsf{P}_{m,d}^\perp = \mathsf{P}_{m,m-d-1}$.

For $\beta \in \mathsf{P}_m$, denote by $\mathrm{supp}(\beta)$ the support of β, that is $\mathrm{supp}(\beta) = \{x \mid \beta(x) = 1\}$. Define $\mathrm{wt}(\beta) = |\mathrm{supp}(\beta)|$. Define the *character function* $\chi_\beta : \mathsf{P}_{m,d} \to \mathbb{R}$ as $\chi_\beta(f) = (-1)^{\langle\beta,f\rangle}$.

Fact 2.14 gives a method of testing whether $\beta \in \mathsf{P}_{m,d}^\perp$ for any degree d: pick a random $g \sim \mathsf{P}_{m,d}$ and evaluate $\chi_\beta(g)$. If $\beta \in \mathsf{P}_{m,d}^\perp$, then the result is always 1, otherwise, the result is 1 half of the time, and -1 the other half of the time.

In [29], Dinur and Guruswami proved that if β is far from $\mathsf{P}_{m,d}^\perp$, then it suffices to pick g in a *pseudo-random* way.

Theorem 2.15 ([29])**.** *Let d be a multiple of 4. If $\beta \in \mathsf{P}_m$ is such that $\Delta(\beta, \mathsf{P}_{m,d}) \geq 2^{d/2}$, then*

$$\mathop{\mathbf{E}}_{g\sim\mathsf{P}_{m,d/4}}\left[\left|\mathop{\mathbf{E}}_{h\sim\mathsf{P}_{m,3d/4}}[\chi_\beta(gh)]\right|\right] \leq 2^{-4\cdot 2^{d/4}}.$$

Note that $\chi_\beta(gh) = (-1)^{\langle\beta g,\beta h\rangle}$. We now prove a generalization of the above theorem that is useful for later applications in this thesis.

The setting is that we have an additional t functions $A_1, \ldots, A_t : \mathsf{P}_{m,d} \to \mathbb{F}_2$. We show that as long as t is small compared to $2^{d/2}$, the expectation

$$\mathop{\mathbf{E}}_{g,h}\left[(-1)^{\langle\beta g,\beta h\rangle+\sum_{i=1}^t A_i(g)A_i(h)}\right]$$

is still close to 0 for arbitrary $A_1, \ldots, A_t$.

We use some of the notions and results from [29].

Definition 2.16. *For β and $k \leq d$, define*

$$B_{d,k}^{(m)}(\beta) := \left\{g \in \mathsf{P}_{m,k} \mid \beta g \in \mathsf{P}_{m,m-d-1+k}\right\}.$$

Note that $B_{d,k}^{(m)}(\beta)$ is a subspace of $\mathsf{P}_{m,k}$.

For positive integers d, k, define $\Phi_{d,k} : \mathbb{N}^+ \to \mathbb{N}$ as follows: if $d < k$, then $\Phi_{d,k}$ is identically 0, otherwise

$$\Phi_{d,k}(D) = \min_{\substack{m>d \\ \beta\in\mathsf{P}_m:\Delta(\beta,P(m,m-d-1))\geq D}} \left\{\dim(P(m,k)) - \dim(B_{d,k}^{(m)}(\beta))\right\}.$$

The following two claims are from [29], which serve as the basis step and induction step for their lower-bound for $\Phi_{d,k}(D)$.

Claim 2.17. *For $d \geq k$ and $D \geq 1$, $\Phi_{d,k}(D) \geq 1$.*

Claim 2.18. *For all $d \geq k$ and $40 < D < 2^d$, $\Phi_{d,k}(D) \geq \Phi_{d-1,k}(D/4) + \phi_{d-1,k-1}(D/4)$.*

For $D = 2^{d-4} = 4^{d/2-2}$ and $k = d/2$, applying the above for a depth of $d/2-4$, reducing D from $4^{d/2-2}$ to 16, we have $\Phi_{d,d/2}(2^{d-4}) \geq 2^{d/2-4}$. This gives the following theorem.

Theorem 2.19. *For all integers m, d such that $m > d > 0$ and $4|d$, if $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ has distance more than 2^{d-4} from $\mathsf{P}_{m,m-d-1}$, then the subspace $B^{(m)}_{d,d/2}(\beta)$ (as a subspace of $\mathsf{P}_{m,d/2}$) has codimension at least $2^{d/2-4}$.*

We remark that Dinur and Guruswami used different degree parameters in [29] for their application. Otherwise, the above theorem is the same as in [29].

Theorem 2.20. *Let $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ be a polynomial with distance more than 2^{d-4} from $\mathsf{P}_{m,m-d-1}$. Let $t \in \mathbb{N}^+$ and $A_1, \ldots, A_t : \mathsf{P}_{m,d/2} \to \mathbb{F}_2$ be some arbitrary t functions. Let μ be the uniform distribution on $\mathsf{P}_{m,d/2}$. Then*

$$\begin{aligned} &\mathop{\mathbf{E}}_{g,h\sim\mu}\left[\chi_\beta(gh)\cdot(-1)^{\sum_{i=1}^t A_i(g)A_i(h)}\right] \\ =\ &\mathop{\mathbf{E}}_{g,h\sim\mu}\left[(-1)^{\langle\beta g,\beta h\rangle+\sum_{i=1}^t A_i(g)A_i(h)}\right] \leq 2^{-(2^{d/2-4}-t)/2}. \end{aligned}$$

Proof. Denote by $\mathcal{W}$ the quotient space $\mathsf{P}_{m,d/2}\Big/B^{(m)}_{d,d/2}(\beta)$. By Theorem 2.19, we have $w := \dim(\mathcal{W}) = \operatorname{codim}(B^{(m)}_{d,d/2}(\beta)) \geq 2^{d/2-4}$.

The expectation we are considering can be written as

$$\mathop{\mathbf{E}}_{g_0,h_0\sim\mathcal{W}}\ \mathop{\mathbf{E}}_{\substack{g:g-g_0\in B^{(m)}_{d,d/2}(\beta)\\ h:h-h_0\in B^{(m)}_{d,d/2}(\beta)}}\left[(-1)^{\langle\beta g,\beta h\rangle+\sum_{i=1}^t A_i(g)A_i(h)}\right]. \tag{2.1}$$

Consider $f \in \mathsf{P}_{m,d/2}$ and $g \in B^{(m)}_{d,d/2}(\beta)$. We have $\langle\beta f, \beta g\rangle = \langle\beta g, f\rangle = 0$, because $f \in \mathsf{P}_{m,d/2}$ and $\beta g \in \mathsf{P}_{m,m-d/2-1} = \mathsf{P}^\perp_{m,d/2}$. This allows us to define "dot product" between elements in $\mathcal{W}$. In particular, for any $f, f', g, g' \in \mathsf{P}_{m,d/2}$ such that $f - f', g - g' \in B^{(m)}_{d,d/2}(\beta)$, we have

$$\begin{aligned} &\langle\beta f', \beta g'\rangle \\ =\ &\langle\beta f', \beta g'\rangle + \langle\beta(f-f'), \beta g'\rangle + \langle\beta f', \beta(g-g')\rangle + \langle\beta(f-f'), \beta(g-g')\rangle \\ =\ &\langle\beta f, \beta g\rangle. \end{aligned}$$

This means that taking any representative from $\mathcal{W}$ will give the same result for this "dot product".

We can thus further rewrite the expectation as

$$(2.1) = \mathop{\mathbf{E}}_{g_0, h_0 \sim \mathcal{W}} \left[(-1)^{\langle \beta g_0, \beta h_0 \rangle} \mathop{\mathbf{E}}_{\substack{g : g - g_0 \in B^{(m)}_{d,d/2}(\beta) \\ h : h - h_0 \in B^{(m)}_{d,d/2}(\beta)}} \left[(-1)^{\sum_{i=1}^{t} A_i(g) A_i(h)} \right] \right] . \quad (2.2)$$

Consider the matrix $M \in \mathbb{R}^{2^{w+t} \times 2^{w+t}}$, where the rows and columns are indexed by a pair (f_0, a) where $f_0 \in \mathcal{W}$ and $a \in \mathbb{F}_2^t$, and the entries are

$$M_{(f_0,a),(g_0,b)} = (-1)^{\langle \beta f_0, \beta g_0 \rangle + \sum_{i=1}^{t} a_i b_i} .$$

Define vector $u \in \mathbb{R}^{2^{w+t}}$ as

$$u_{f_0,a} = \Pr_{g \sim \mathsf{P}_{m,d/2}} \left[g - f_0 \in B^{(m)}_{d,d/2}(\beta) \wedge \forall i \in [t], A_i((g)) = a_i \right] .$$

Since in (2.2), g and h are sampled independently, we can verify that the expectation in (2.2) is exactly $u^T M u$. Moreover, since g is chosen uniformly random from $\mathsf{P}_{m,d/2}$, the probability that $g - f_0 \in B^{(m)}_{d,d/2}(\beta)$ is exactly 2^{-w}, thus all entries in u has absolute value at most 2^{-w}, and therefore $\|u\|_2 \leq 2^{-w/2}$.

We finish the proof by studying the spectrum of M. Observe that M can be written as the tensor product of a $2^w \times 2^w$ matrix and a $2^t \times 2^t$ matrix as follows. Define $W \in \mathbb{R}^{2^w \times 2^w}$ as

$$W_{f_0,g_0} = (-1)^{\langle \beta f_0, \beta g_0 \rangle} ,$$

for $f_0, g_0 \in \mathcal{W}$. Define $H \in \mathbb{R}^{2^t \times 2^t}$ as

$$H_{a,b} = (-1)^{\sum_{i=1}^{t} a_i b_i} .$$

We can easily verify that $M = W \otimes H$.

The matrix H satisfies $HH^T = 2^t \cdot I$, where I is the identity matrix, therefore we have that the eigenvalues of H all have absolute value exactly $2^{t/2}$. For the spectrum of W, let $f_0, g_0 \in \mathcal{W}$ be two rows of W. Consider the dot product of row f_0 and g_0 of matrix W

$$W_{f_0}^T W_{g_0} = \sum_{h_0 \in \mathcal{W}} (-1)^{\langle \beta(f_0 + g_0), \beta h_0 \rangle} = \sum_{h_0 \in \mathcal{W}} (-1)^{\langle \beta(f_0 + g_0), h_0 \rangle} .$$

The above sum is 2^w if $\beta(f_0 + g_0) \in \mathsf{P}_{m,m-d/2-1}$, or in other words f_0 and g_0 belong to the same coset in $\mathcal{W}$, and otherwise the sum is 0. Hence we have $WW^T = 2^w \cdot I$, and thus the eigenvalues of W all have absolute value $2^{w/2}$. We conclude that the tensor product matrix $M = W \otimes H$ has eigenvalues with absolute value $2^{(w+t)/2}$.

Using Claim 2.12, we can now upper-bound the absolute value of the expectation by $|u^T M u| \leq 2^{(w+t)/2} \cdot \|u\|_2^2 = 2^{-(w-t)/2}$. □

2.4 Fourier Analysis

Let (Ω, μ) be a finite probability space with $|\Omega| = q$, and we assume that for every $x \in \Omega$, $\mu(x) > 0$. Let $(\Omega^n, \mu^{\otimes n})$ be a product space. In this thesis, we will study functions over $(\Omega^n, \mu^{\otimes n})$, and in most cases we have $\Omega = \mathbb{F}_2$. We now introduce some analytical tools that help us understand the structures of these functions. An excellent resource for more details of many of the results presented here can be found in the book by Ryan O'Donnell [88].

2.4.1 Fourier Decomposition

Let $\chi_0, \ldots, \chi_{q-1} : \Omega \to \mathbb{R}$ be an orthonormal basis for the space $L^2(\Omega, \mu)$ with respect to the inner-product $\langle \cdot, \cdot \rangle_\mu$. Let this basis be such that $\chi_0 = \mathbf{1}$, where $\mathbf{1}$ is the identically one function.

When $\Omega = \mathbb{F}_2$ and μ is the uniform distribution on $\mathbb{F}_2$, we use the following as the basis:

$$\chi_r(x) = (-1)^{rx}, \quad r = 0, 1\,.$$

For $\sigma \in \Omega^n$, define

$$\chi_\sigma(x_1, \ldots, x_n) = \prod_{i=1}^{n} \chi_{\sigma_i}(x_i)\,.$$

Then $\{\chi_\sigma\}_{\sigma \in \Omega^n}$ forms an orthonormal basis for $L^2(\Omega^n, \mu^{\otimes n})$, and every function $f \in L^2(\Omega^n, \mu^{\otimes n})$ can be written as

$$f(x) = \sum_{\sigma \in \Omega^n} \hat{f}_\sigma \chi_\sigma(x)\,,$$

where the Fourier coefficients $\left\{\hat{f}_\sigma : \Omega^n \to \mathbb{R}\right\}_{\sigma \in \Omega^n}$ are defined by $\hat{f}_\sigma = \langle f, \chi_\sigma \rangle_{\mu^{\otimes n}}$. We summarize the basic facts about $\hat{f}$ below.

Fact 2.21. *Let* $\mathbf{0}$ *be the all-*0 *vector. Let* $f \in L^2(\Omega^n, \mu^{\otimes n})$*. Then*

$$\mathbf{E}[f] = \hat{f}_{\mathbf{0}}, \quad \mathbf{Var}[f] = \sum_{\sigma \neq \mathbf{0}} \hat{f}_\sigma^2\,.$$

The following is known as Parseval's identity.

Fact 2.22. *Let* $f \in L^2(\Omega^n, \mu^{\otimes n})$*. Then*

$$\|f\|_2^2 = \mathop{\mathbf{E}}_x \left[f(x)^2\right] = \sum_{\sigma \in \Omega^n} \hat{f}_\sigma^2\,.$$

We also make extensive use of the following Efron-Stein decomposition.

Theorem 2.23 ([35, 86]). *Any function $f \in L^2(\Omega^n, \mu^{\otimes n})$ can be uniquely decomposed as*

$$f(x) = \sum_{S \subseteq [n]} f_S(x)\,,$$

where

- *the function $f_S(x)$ depends only on $x_S = \{x_i \mid i \in S\}$;*
- *for every $S, T \subseteq [n]$, $S - T \neq \emptyset$, $x' \in \Omega^n$, it holds that*

$$\mathbf{E}[f_S(x) \mid x_T = x'_T] = 0\,.$$

For $\sigma \in \Omega^n$, let $\mathrm{Set}(\sigma) = \{i \mid \sigma_i \neq 0\}$, and let $|\sigma| = |\mathrm{Set}(\sigma)|$. It is easily verified that the Efron-Stein decomposition is related to the Fourier decomposition as follows

$$f_S(x) = \sum_{\substack{\sigma \in \Omega^n \\ \mathrm{Set}(\sigma) = S}} \hat{f}_\sigma \chi_\sigma(x)\,.$$

2.4.2 Influences and Noise

The notion of *influence* of a coordinate on a function has proved to be influential in combinatorics and theoretical computer science.

Definition 2.24. *For $f \in L^2(\Omega^n, \mu^{\otimes n})$, $i \in [n]$, the influence of i on f is defined as*

$$\mathrm{Inf}_i(f) = \mathop{\mathbf{E}}_{x_{[n]-\{i\}}} [\mathop{\mathbf{Var}}_{x_i}[f(x)]]\,.$$

Note that when we refer to influence, it is always with respect to the underlying probability space $(\Omega^n, \mu^{\otimes n})$. We have the following characterization of influence in terms of Fourier decomposition and Efron-Stein decomposition.

Proposition 2.25. *For $f \in L^2(\Omega^n, \mu^{\otimes n})$ and $i \in [n]$,*

$$\mathrm{Inf}_i(f) = \sum_{\substack{\sigma \in \Omega^n \\ i \in \mathrm{Set}(\sigma)}} \hat{f}_\sigma^2 = \sum_{S \ni i} \mathbf{E}[f_S^2]\,.$$

Let the total influence $\mathrm{Inf}(f) = \sum_{i \in [n]} \mathrm{Inf}_i(f)$ be the sum of influences of all coordinates on f.

The above analytical definition of influence can be generalized to the influence of a set of coordinates. The following is defined in [90]

Definition 2.26. *For a function $f \in L^2(\Omega^n, \mu^{\otimes n})$ and a set of coordinates $S \subseteq [n]$, we define the influence of S on f to be*

$$\mathrm{Inf}_S(f) = \sum_{\substack{\sigma \in \Omega^n \\ S \subseteq \mathrm{Set}(\sigma)}} \hat{f}_\sigma^2\,.$$

The Bonami-Beckner operator, also known as noise operator, is defined as follows.

Definition 2.27. *Let $0 \leq \rho \leq 1$. The Bonami-Beckner operator T_ρ is a linear operator mapping $f \in L^2(\Omega^n, \mu^{\otimes n})$ to $T_\rho f$ as follows*

$$(T_\rho f)(x) = \mathop{\mathbf{E}}_{y \sim_\rho x}[f(y)],$$

where y is sampled by setting each bit independently to $y_i = x_i$ with probability ρ, and otherwise sampled according to μ with probability $1 - \rho$.

Again we have the following Fourier/Efron-Stein characterization of T_ρ.

Proposition 2.28. *For any $f \in L^2(\Omega^n, \mu^{\otimes n})$ and $0 \leq \rho \leq 1$,*

$$T_\rho f = \sum_{\sigma \in \Omega^n} \rho^{|\sigma|} \widehat{f}_\sigma \chi_\sigma .$$

Fact 2.29. *For any $0 \leq \rho, \rho' \leq 1$ and $f \in L^2(\Omega^n, \mu^{\otimes n})$, $T_\rho T_{\rho'} f = T_{\rho\rho'} f$.*

The following Hypercontractivity Theorem proved by Bonami shows that T_ρ "smoothens" the random variable f.

Theorem 2.30 ([22])**.** *Let $\Omega = \mathbb{F}_2$ and μ be the uniform distribution. Then for any $f \in L^2(\Omega^n, \mu^{\otimes n})$, $1 \leq p \leq q \leq \infty$, and $0 \leq \rho \leq \sqrt{(p-1)/(q-1)}$, we have*

$$\left\|T_\rho f\right\|_q \leq \left\|f\right\|_p .$$

Note that for any $p \geq 1$, we naturally have $\|T_\rho f\|_p \leq \|f\|_p$ by Jensen's Inequality. The operator T_ρ is "hypercontractive" in the sense that we can even bound the l_q norm of $T_\rho f$ by the l_p norm of f. Example 2.1 provides some intuition why we consider such random variables as smoother and better-behaved.

We define noisy influence as $\mathrm{Inf}_S^{(\rho)}(f) = \mathrm{Inf}_S(T_\rho f)$ for all $S \subseteq [n]$, and similarly $\mathrm{Inf}^{(\rho)}(f) = \sum \mathrm{Inf}_i^{(\rho)}(f)$. The following bound for the total noisy influence of functions with range $[-1, 1]$ appeared in [90, Lemma 5.9].

Proposition 2.31 ([90])**.** *For any $f : \Omega^n \to [-1, 1]$ and $0 < \rho \leq 1$, we have*

$$\mathrm{Inf}^{(\rho)}(f) = \sum_{i \in [n]} \mathrm{Inf}_i^{(\rho)}(f) \leq (1 - \rho)^{-1} .$$

More generally, for $1/2 \leq \rho < 1$, we have the following upper-bound in terms of influence of sets of coordinates

$$\sum_{S \subseteq [n], |S| \leq m} \mathrm{Inf}_S^{(\rho)}(f) \leq (m/2(1-\rho))^m .$$

The following concept of lifted functions is useful in the context of projection games, which we will describe in more detail in Section 3.3.

We say that a mapping $\pi : R \to L$ is d-to-1 if for all l, $|\pi^{-1}(l)| = d$.

Definition 2.32. *Let $|L| = n$, and $|R| = nd$, and let $\pi : R \to L$ be a d-to-1 mapping. Given function $f : \Omega^R \to \mathbb{R}$, define the lifted version of f naturally induced by π, denoted as $\overline{f}^{\pi} : (\Omega^d)^L \to \mathbb{R}$, where*

$$\overline{f}^{\pi}(\overline{x}^{\pi}) = f(x)\,,$$

where $\overline{x}^{\pi} \in (\Omega^d)^L$ is defined as $\overline{x}^{\pi}_{r,t} = x_{(r,t)}$ for $r \in L, t \in [d]$.

In terms of influence, we have the following relation between f and $\overline{f}$, due to Wenner [104].

Proposition 2.33 ([104])**.** *For any r, we have*

$$\mathrm{Inf}_r(\overline{f}) \leq \sum_{r' : \pi(r') = r} \mathrm{Inf}_{r'}(f)\,.$$

Proof. The claim follows by applying Proposition 2.25 and comparing the terms. □

2.4.3 Conditional Expectation Operator

Let $(\Omega \times \Psi, \mu)$ be two correlated probability spaces, and $f \in L^2(\Psi, \mu)$ be a function on Ψ. Sometimes, we may only have control over some variables in Ω, and we would like to understand the expected behavior of f given observations of Ω. Formally, we define the following conditional expectation operator.

Definition 2.34. *Let $(\Omega \times \Psi, \mu)$ be two correlated spaces. The conditional expectation operator $\mathcal{U}$ associated with (Ω, Ψ) is the operator mapping $f \in L^2(\Psi, \mu)$ to $\mathcal{U}f \in L^2(\Omega, \mu)$ by*

$$(\mathcal{U}f)(x) = \mathbf{E}[f(Y) \mid X = x]\,,$$

for $x \in \Omega$ and $(X, Y) \in \Omega \times \Psi$ is distributed according to μ.

An important property we need in the analysis, due to Mossel [85], is that the Efron-Stein decomposition commutes with the conditional expectation operator.

Proposition 2.35 ([85])**.** *Let $(\Omega \times \Psi, \mu) := (\prod \Omega_i \times \prod \Psi_i, \bigotimes \mu_i)$ be correlated space and let $\mathcal{U} := \bigotimes \mathcal{U}_i$ be the conditional expectation operator associated with Ω and Ψ. Suppose $f \in L^2(\Psi, \mu)$ has Efron-Stein decomposition $f(x) = \sum_{S \subseteq [n]} f_S(x_S)$. Then the Efron-Stein decomposition of $\mathcal{U}f$ satisfies $(\mathcal{U}f)_S = \mathcal{U}(f_S)$ for $S \subseteq [n]$.*

The following result, due to Mossel [85], shows that in the above setting, if the correlations between all Ω and Ψ are less than 1, then the L^2 norms of the high-degree terms of $\mathcal{U}f$ are small.

Proposition 2.36 ([85])**.** *Assume the setting of Proposition 2.35 and that for all i, we have*

$$\rho(\Omega_i, \Psi_i; \mu_i) \leq \rho_i \,.$$

Then for all f, we have

$$\|\mathcal{U}(f_S)\|_2 \leq \left(\prod_{i \in S} \rho_i\right) \|f_S\|_2 \,.$$

2.4.4 The Invariance Principle

The Berry-Esseen Central Limit Theorem says that suppose we have n independent random variables $X_1, \ldots, X_n$, with $\mathbf{E}[X_i] = 0$ and $\sum_i \mathbf{E}[X_i^2] = 1$, then the random variable $S = \sum_i X_i$ and the standard Gaussian $Z \sim \mathcal{N}(0,1)$ are close, or more precisely, for all $u \in \mathbb{R}$

$$|\Pr[S \leq u] - \Pr[Z \leq u]| \leq c\gamma \,,$$

where $\gamma = \sum_i \|X_i\|_3^3$. Thus the Central Limit Theorem allows us to apply our knowledge from Gaussian geometry to prove properties about random variables arising from other applications. Moving between probability distributions that are *similar* is a powerful paradigm, and has been used in many areas of mathematics and theoretical computer science.

The Invariance Principle can be viewed as a generalization of the above Central Limit Theorem. It was first proved in [87] for low degree multilinear polynomials, and later generalized to product of functions on correlated spaces in [86]. The main theorem in [87] studies $\mathbf{E}\left[\Phi(Q(X_1, \ldots, X_n))\right]$, where $\Phi : \mathbb{R} \to \mathbb{R}$ is a smooth "test" function, and Q is a multi-linear polynomial. The Invariance Principle says that if the random variables $X_1, \ldots, X_n$ are "reasonable" — using terminology from [88] — and the function Q does not have coordinates with large influence, then $X_1, \ldots, X_n$ can be replaced with standard Gaussians and the expectation does not change much. The main theorem in [86] deals with product of functions, and the requirement is that the functions do not share a coordinate that has large influence. Since we do not use the original forms of these theorems in this thesis, we refer to [87] and [86] for the precise statements of the results.

Both theorems have led to many exciting discoveries in hardness of approximation. The result in [87] implies the Majority is Stablest Theorem, which, combined with [71] shows that the Goemans-Williamson algorithm for Max-Cut is optimal under the Unique Games Conjecture. The general result in [86] was used by Raghavendra [91] to show that certain generic algorithm is optimal for every Max-CSP assuming the Unique Games Conjecture. Another example is a sufficient condition for a wide class of Max-CSP to be hard to approximate, given by Austrin and Mossel [10],

Since its discovery, the Invariance Principle has found numerous applications, and has been refined and improved in different ways. For examples of their adap-

tation in hardness of approximation based on Label-Cover hardness, see for instance [89], [24], [104].

2.5 A Short Introduction to Long-Code

We now apply the tools presented earlier in the chapter to some encoding problem. The basic setting is that we want to encode a string $a \in \mathbb{F}_2^m$ into some n-bit string $b \in \mathbb{F}_2^n$, and we would like to test whether some given n-bit string is *close to* a codeword by reading from the string as few bits as possible. Such codes are known as *Locally Testable Codes*. Since this is not the main focus of this thesis, we do not formally define what those are and go into much details. We refer to the survey by Oded Goldreich [40] (and revisions on his website) for more information.

We describe the constructions of Long-Code, Hadamard-Code, and also Low-Degree-Long-Code. The materials included here are mostly straightforward applications of definitions and facts in the previous sections. Long-Code was introduced in the pioneering work of Bellare, Goldreich and Sudan [16]. They also presented the basic structure of most of the recent inapproximability results — the composition of Long-Code and Label-Cover. The Fourier analytic method of analyzing Long-Code test was first introduced in [15], and used to prove hardness of approximation results in the breakthrough paper by Håstad [51]. Application of Low-Degree-Long-Code in the context of proving inapproximability results was first studied in [29].

In many applications, we have some further restrictions on what strings in $\mathbb{F}_2^m$ are valid. This could be specified by a subset of $\mathbb{F}_2^m$, by a set of polynomials on m bits, or by linear or affine subspaces (note that this is a special case of specifying a subset of $\mathbb{F}_2^m$ by the set of common roots of a set of polynomials).

Throughout this section, the length of the codeword n is a power of 2. It is therefore more convenient to think of the encoding scheme as mapping words $a \in \mathbb{F}_2^m$ to functions on some r bits $\mathbb{F}_2^r \to \mathbb{F}_2$, where $n = 2^r$. Also, it is usually more convenient to think of functions on r bits as $\mathbb{F}_2^r \to \{-1, 1\}$ by mapping elements in $\mathbb{F}_2$ to elements in $\{-1, 1\}$ according to $0 \mapsto -1$, and $1 \mapsto -1$.

Recall that P_m is the linear space of all polynomials on m bits, and $\mathsf{P}_{m,d}$ is the subspace of polynomials on m bits of degree at most d.

Let us start with Long-Code.

Definition 2.37. *For $a \in \mathbb{F}_2^m$, its* Long-Code *encoding is defined by*

$$\begin{aligned} \mathsf{LC}_a : \mathsf{P}_m &\to \{-1, 1\} \\ f &\mapsto (-1)^{f(a)}. \end{aligned}$$

The Long-Code uses 2^{2^m} bits to encode m bits. A more efficient way of encoding is to only evaluate a with functions in $\mathsf{P}_{m,d}$ for some small d. This gives the Low-Degree-Long-Code.

Definition 2.38. *Let $1 \leq d \leq m$. For $a \in \mathbb{F}_2^m$, its* Low-Degree-Long-Code $_d$ *encoding is*

$$\begin{aligned} \mathsf{SC}_{d,a} : \mathsf{P}_{m,d} &\to \{-1,1\} \\ f &\mapsto (-1)^{f(a)}. \end{aligned}$$

Taking $d = 1$ in the above definition gives the Hadamard-Code. For function $f \in \mathsf{P}_{m,1}$, defined by $f(x) = \sum_{i=1}^{m} a_i x_i$, we can write the evaluation of f at x as the dot product $f(x) = \langle a, x \rangle$.

Definition 2.39. *For $a \in \mathbb{F}_2^m$, its* Hadamard-Code *encoding is*

$$\begin{aligned} \mathsf{HD}_a : \mathbb{F}_2^m &\to \{-1,1\} \\ x &\mapsto (-1)^{\langle a,x \rangle}. \end{aligned}$$

The Hadamard-Code code has length 2^m, and the set of all codewords are exactly the set of character functions $\{\chi_\sigma\}_{\sigma \in \mathbb{F}_2^m}$.

We use $\delta(f,g)$ to denote the fraction of inputs on which f and g differ. For a set of functions $\mathcal{F}$, we say that f is ε-close to $\mathcal{F}$ if there exists some $g \in \mathcal{F}$, such that $\delta(f,g) \leq \varepsilon$.

Theorem 2.40. *For an arbitrary function $f : \mathbb{F}_2^m \to \{-1,1\}$ and a character χ_σ, we have $\delta(f, \chi_\sigma) = \frac{1}{2}(1 - \hat{f}_\sigma)$.*

Proof. By definition of Fourier coefficient, we have

$$\begin{aligned} \hat{f}_\sigma &= \mathop{\mathbf{E}}_{x \sim \mathbb{F}_2^m} [f(x)\chi_\sigma(x)] \\ &= \Pr_{x \sim \mathbb{F}_2^m} [f(x) = \chi_\sigma(x)] - \Pr_{x \sim \mathbb{F}_2^m} [f(x) \neq \chi_\sigma(x)] \\ &= (1 - \delta(f, \chi_\sigma)) - \delta(f, \chi_\sigma) = 1 - 2\delta(f, \chi_\sigma). \end{aligned}$$

□

To test if a function $f : \mathbb{F}_2^m \to \{-1,1\}$ is close to the Hadamard-Code of some $a \in \mathbb{F}_2^m$, we use the following famous BLR test, named after Blum, Luby and Rubinfeld [21]:

- Choose $x, y \sim \mathbb{F}_2^m$ independently.
- Accept if $f(x)f(y) = f(x+y)$.

Bellare, Coppersmith, Håstad, Kiwi and Sudan [15] analyzed the test using Fourier analysis and gave a complete description of the relationship between the probability that the BLR test succeeds and the distance between f and some linear function.

We now describe the key step in their analysis. The acceptance probability of the BLR test can be written as

$$\begin{aligned}
&\frac{1}{2} + \frac{1}{2} \mathop{\mathbf{E}}_{x,y} [f(x)f(y)f(x+y)] \\
&= \frac{1}{2} + \frac{1}{2} \sum_{\alpha,\beta,\gamma} \hat{f}_\alpha \hat{f}_\beta \hat{f}_\gamma \mathop{\mathbf{E}}_{x,y} \left[\chi_\alpha(x)\chi_\beta(y)\chi_\gamma(x+y)\right] \\
&= \frac{1}{2} + \frac{1}{2} \sum_{\alpha} \hat{f}_\alpha^3 \\
&\leq \frac{1}{2} + \frac{1}{2} \max_\alpha \left|\hat{f}_\alpha\right| .
\end{aligned}$$

Note that in the last step we used Parseval's Identity. Therefore if f passes the BLR test with probability $\frac{1}{2} + \varepsilon$, then there exists $\alpha \in \mathbb{F}_2^m$, such that f is $\frac{1}{2} - \varepsilon$ close to either χ_α or $-\chi_\alpha$. □

In hardness of approximation applications, we not only want to test if a given string is close to the HADAMARD-CODE of some $a \in \mathbb{F}_2^m$, we also want to make sure that a satisfies some constraint, for instance, that a is in some linear subspace. Let $\mathcal{A} \subseteq \mathbb{F}_2^m$ be a linear subspace. As defined in Section 2.3, we denote the dual space of $\mathcal{A}$ as $\mathcal{A}^\perp$.

Definition 2.41. *A function is* conditioned over $\mathcal{A}$ *if for any* $x \in \mathbb{F}_2^m$ *and* $a \in \mathcal{A}^\perp$, *we have* $f(x+a) = f(x)$.

This is enforced by choosing a representative for each coset of $\mathcal{A}^\perp$, and returning the value of f on the representative for all inputs in the same coset.

Claim 2.42. *If f is conditioned over a linear subspace $\mathcal{A} \subseteq \mathbb{F}_2^m$, and $\sigma \notin \mathcal{A}$, then $\hat{f}_\sigma = 0$.*

Proof. Using the definition of Fourier expansion, we have

$$\begin{aligned}
\hat{f}_\sigma &= \mathop{\mathbf{E}}_{x} [f(x)\chi_\sigma(x)] \\
&= \mathop{\mathbf{E}}_{a \in \mathcal{A}^\perp} \mathop{\mathbf{E}}_{x} [f(x+a)\chi_\sigma(x+a)] \\
&= \mathop{\mathbf{E}}_{a \in \mathcal{A}^\perp} \mathop{\mathbf{E}}_{x} [f(x)\chi_\sigma(x+a)] \\
&= \mathop{\mathbf{E}}_{x} \left[f(x)\chi_\sigma(x) \mathop{\mathbf{E}}_{a \in \mathcal{A}^\perp} [\chi_\sigma(a)] \right] \\
&= \hat{f}_\sigma \mathop{\mathbf{E}}_{a \in \mathcal{A}^\perp} [\chi_\sigma(a)] .
\end{aligned}$$

Since $\sigma \notin \mathcal{A}$, we have that $\mathbf{E}_{a \in \mathcal{A}^\perp}[\chi_\sigma(a)] = 0$, thus $\hat{f}_\sigma = 0$. □

We now turn to LONG-CODE. Observe that there is a bijection between $\mathbb{F}_2^{\mathbb{F}_2^m}$ and P_m. In many applications where LONG-CODE is used, we in fact view the codewords as function $f : \mathbb{F}_2^{\mathbb{F}_2^m} \to \{-1, 1\}$. Such a function has Fourier decomposition

$$\begin{aligned} f(x) &= \sum_{\alpha \in \mathbb{F}_2^{\mathbb{F}_2^m}} \hat{f}_\alpha \chi_\alpha(x) \\ &= \sum_{\alpha \subseteq \mathbb{F}_2^m} \hat{f}_\alpha \prod_{a \in \alpha} (-1)^{x_a} . \end{aligned}$$

The LONG-CODE encoding LC_a is sometimes also known as a *dictator function*, because the value of the function depends only on the a-th coordinate.

When testing and decoding LONG-CODE, we usually get some set $\sigma \subseteq \mathbb{F}_2^m$ of m-bit strings that the function "depends" on. The following folding makes sure that the sets σ with $\hat{f}_\sigma \neq 0$ is non-empty.

Definition 2.43. *Function f is* folded over constant*, or sometimes simply known as* odd*, if for all $x \in \mathbb{F}_2^{\mathbb{F}_2^m}$, we have $f(x+1) = -f(x)$, where $x+1$ denotes the string where we negate all bits of x.*

To actually enforce this, we can ask for an evaluation table of size 2^{2^m-1} containing the value of all points with $x_{\mathbf{0}} = 0$, where the subscript $\mathbf{0}$ of x denotes the m-bit string with all entries being 0. Then, whenever we want to read $f(x)$, we simply read entry $(x - x_{\mathbf{0}})$ from the table and return $(-1)^{x_{\mathbf{0}}} \cdot f(x - x_{\mathbf{0}})$.

Claim 2.44. *Suppose f is folded over constant. Then for any $\sigma \subseteq \mathbb{F}_2^m$ such that $|\sigma|$ is even, we have $\hat{f}_\sigma = 0$.*

Proof. Let $\sigma \subseteq \mathbb{F}_2^m$. By definition of Fourier coefficients, we have

$$\begin{aligned} \hat{f}_\sigma &= \mathop{\mathbf{E}}_x[f(x)\chi_\sigma(x)] = \mathop{\mathbf{E}}_x[f(1+x)\chi_\sigma(1+x)] \\ &= \mathop{\mathbf{E}}_x[-f(x)\chi_\sigma(x)\chi_\sigma(1)] \\ &= (-1)^{1+|\sigma|}\hat{f}_\sigma . \end{aligned}$$

This means that either $|\sigma_i|$ is odd, or $\hat{f}_\sigma = 0$. □

For any subset $S \subseteq \mathbb{F}_2^m$, the following folding trick ensures that the set σ with $\hat{f}_\sigma \neq 0$ is a subset of S.

Definition 2.45. *Let $S \subseteq \mathbb{F}_2^m$ be a subset of indices. For $x \in \mathbb{F}_2^{\mathbb{F}_2^m}$, define $x \wedge S$ to be the vector where $(x \wedge S)_a = x_a$ for $a \in S$, and $(x \wedge S)_a = 0$ for $a \in \mathbb{F}_2^m - S$. We say that a function is* conditioned on S*, if for all $x \in \mathbb{F}_2^{\mathbb{F}_2^m}$, we have $f(x) = f(x \wedge S)$. In other words, the function only depends on inputs in S.*

To enforce conditioning, we simply return $f(x \wedge S)$ for any x.

Claim 2.46. *If f is conditioned on S and $\sigma \subseteq \mathbb{F}_2^m$ is such that $\sigma \nsubseteq S$. Then $\hat{f}_\sigma = 0$.*

Proof. Fix the input inside S to be some arbitrary $x_0 \in \mathbb{F}_2^S$. Then

$$\mathop{\mathbf{E}}_{x}[f(x)\chi_\sigma(x) \mid x|_S = x_0] \;= f(x_0)\,\mathbf{E}[\chi_\sigma(x) \mid x|_S = x_0] = 0\,.$$

Note that $\hat{f}_\sigma$ is just the expectation of the left hand side of the above equation over x_0. Therefore $\hat{f}_\sigma = 0$. □

Finally, we study the behavior of Fourier characters under coordinate projection.

Definition 2.47. *Let $\pi : \mathbb{F}_2^m \to \mathbb{F}_2^{m'}$ be a coordinate projection. That is, there is some $S \subseteq [m]$, $|S| = m'$, such that $\pi(x) = x|_S$. For $f : \mathbb{F}_2^{\mathbb{F}_2^{m'}} \to \{-1,1\}$ and $\pi : \mathbb{F}_2^m \to \mathbb{F}_2^{m'}$, define $(f \circ \pi) : \mathbb{F}_2^{\mathbb{F}_2^m} \to \{-1,1\}$ by $(f \circ \pi)(x) = f(x|_S)$.*

The following is easy to verify.

Claim 2.48. *For any $f : \mathbb{F}_2^{\mathbb{F}_2^{m'}} \to \{-1,1\}$, $\pi : \mathbb{F}_2^m \to \mathbb{F}_2^{m'}$ defined as above, and $\sigma \subseteq \mathbb{F}_2^m$, we have $\chi_\sigma(f \circ \pi) = \chi_{\pi_2(\sigma)}(f)$, where $y \in \pi_2(\sigma) \subseteq \mathbb{F}_2^{m'}$ iff there exists an odd number of $x \in \sigma$ with $x|_S = y$.*

The above folding and conditioning over constraint set techniques also extend to Low-Degree-Long-Code. We first describe Fourier analysis for functions in $\mathsf{P}_{m,d} \to \{-1,1\}$.

Recall that in Section 2.3, we define the *character function* χ_β corresponding to $\beta \in \mathsf{P}_m$ as $\chi_\beta(f) = (-1)^{\langle \beta, f \rangle}$.

Definition 2.49 (Character Set)**.** *Define the* character set $\Lambda_{m,d}$ *to be the set of functions $\beta \in \mathsf{P}_m$ which are minimum weight functions in the cosets of $\mathsf{P}_m \Big/ \mathsf{P}^\perp_{m,d}$, where ties are broken arbitrarily.*

We have the following result about the character set and the "Fourier decomposition" for functions $\mathsf{P}_{m,d} \to \mathbb{R}$.

Lemma 2.50 ([29])**.** *The following are true:*

- *For any $\beta, \beta' \in \mathsf{P}_m$, $\chi_\beta = \chi_{\beta'}$ if and only if $\beta - \beta' \in \mathsf{P}^\perp_{m,d}$.*
- *For $\beta \in \mathsf{P}^\perp_{m,d}$, χ_β is the constant 1 function.*
- *For any β, there exists β', such that $\beta - \beta' \in \mathsf{P}^\perp_{m,d}$, and $|\mathrm{supp}(\beta')| = \Delta(\beta, \mathsf{P}^\perp_{m,d})$. We call such β' the minimum support function for the coset $\beta + \mathsf{P}^\perp_{m,d}$.*

- *The characters in the character set $\Lambda_{m,d}$ form an orthonormal basis under the inner product $\langle A, B\rangle = \mathbf{E}_{f\in \mathsf{P}_{m,d}}[A(f)B(f)]$.*
- *Any function $A : \mathsf{P}_{m,d} \to \mathbb{R}$ can be uniquely decomposed as*
$$A(g) = \sum_{\beta\in\Lambda_{m,d}} \widehat{A}_\beta \chi_\beta(g)\,.$$
- *Parseval's identity: For any $A : \mathsf{P}_{m,d} \to \mathbb{R}$, $\sum_{\beta\in\Lambda_{m,d}} \widehat{A}_\beta^2 = \mathbf{E}_{f\sim \mathsf{P}_{m,d}}[A(f)^2]$.*

The following lemma relates characters from different domains related by coordinate projections.

Lemma 2.51 ([29])**.** *Let $m' \le m$, and $S \subseteq [m]$ with $|S| = m'$, and let $\pi : \mathbb{F}_2^m \to \mathbb{F}_2^{m'}$ be a projection, mapping $x \in \mathbb{F}_2^m$ to $x|_S \in \mathbb{F}_2^{m'}$. Then for $f \in \mathsf{P}_{m',d}$ and $\beta \in \mathsf{P}_m$, we have*
$$\chi_\beta(f\circ\pi) = \chi_{\pi_2(\beta)}(f)\,,$$
where $\pi_2(\beta)(y) = \sum_{x\in\pi^{-1}(y)} \beta(x)$.

The following properties of the Fourier coefficients of folded functions were also studied in [29].

Definition 2.52. *A function $f : \mathsf{P}_{m,d} \to \mathbb{R}$ is folded over constant if for any $p \in \mathsf{P}_{m,d}$, we have $f(p+1) = -f(p)$.*

Lemma 2.53 ([29])**.** *If $f : \mathsf{P}_{m,d} \to \mathbb{R}$ is folded over constant, then for any α such that $\hat{f}_\alpha \neq 0$, we have $\sum_{x\in\mathbb{F}_2^m} \alpha(x) = 1$. In particular, we have* $\mathrm{supp}(\alpha) \neq \emptyset$.

As for conditioning over constraints, in the case of Low-Degree-Long-Code, we cannot condition on any set $S \subseteq \mathbb{F}_2^m$. Instead, the set S is defined by low-degree polynomials. For our application, it suffices to have degree 3.

Definition 2.54 ([29])**.** *Let $q_1, \ldots, q_k \in \mathsf{P}_{m,3}$, and let*
$$J(q_1,\ldots,q_k) := \left\{ \sum_i r_i q_i \mid r_i \in \mathsf{P}_{m,d-3} \right\}.$$
We say that a function $f : \mathsf{P}_{m,d} \to \mathbb{R}$ is folded over *J if f is constant over cosets of J in $\mathsf{P}_{m,d}$.*

The following lemma shows that a function folded over J does not have weight on small support characters that are non-zero on J.

Lemma 2.55 ([29])**.** *Let $\beta \in \mathsf{P}_m$ be such that* $\mathrm{wt}(\beta) < 2^{d-3}$*, and there exists some $i \in [k]$ and $x \in \mathrm{supp}(\beta)$ with $q_i(x) \neq 0$. Then if $f : \mathsf{P}_{m,d} \to \mathbb{R}$ is folded over J, then $\hat{f}_\beta = 0$.*

In the actual reduction, $q_1, \ldots, q_k$ will be the set of functions associated with vertices in the Label-Cover instance, as described in Theorem 3.10.

Chapter 3

Constraint Satisfaction Problems

In a Constraint Satisfaction Problem (CSP) we are given some variables that may take value from some domain, and some restrictions on what combination of values are allowed. The goal is to find a way to assign values to the variables without violating any restrictions.

In this chapter, we give an introduction to combinatorial CSPs. We assume some familiarity with the basics of the theory of computing, complexity classes, and so on. See [2] for a modern treatment. We start with a general formulation of CSP in Section 3.1, and continue onto special classes of CSP that this thesis focuses on. We mainly consider CSP from an optimization perspective. In this introductory part, we also present a simple algorithm for CSP optimization with *surprisingly good* performance.

In Section 3.2, we look at CSP_q, in which the variables have domain $\mathbb{F}_q$. This includes many classical examples studied in computational complexity, such as E3-Sat and GraphColoring.

Section 3.3 is about Label-Cover, one of the basic building blocks of almost all hardness of approximation results in recent years. Many variants of Label-Cover are constructed and used in this thesis. We describe, in some detail, the motivation of these variants, the key parameters that are important for proving inapproximability results, as well as how they are constructed.

3.1 A General Framework of CSP

Let us start with the definition of CSP in its full generality.

Definition 3.1. *Let V be a set of variables, taking values from a domain D of finite size. A* CSP *instance Ψ consists of tuple $\Psi = (\mathcal{C}, \text{wt})$, where $\mathcal{C}$ is a set of constraints, and $\text{wt} : \mathcal{C} \to [0, 1]$ assigns a weight to each constraint.*

The constraint set has the form $\mathcal{C} = \{(k_i, \vec{s_i}, R_i)\}_{i=1}^{r}$, where $k_i \in \mathbb{N}^+$ is the arity *of the i-th constraint, $\vec{s_i} \in V^{k_i}$ is a tuple of k_i variables involved in the i-th constraint, and $R_i \subseteq D^{\vec{s_i}}$ is the combination of values that are allowed.*

Given an assignment $\phi : V \to D$, a constraint $(k, \vec{s}, R)$ is satisfied *if and only if $\phi(\vec{s}) \in R$. Assignment ϕ is a* satisfying assignment *if all constraints in $\mathcal{C}$ are satisfied.*

The value *of an assignment ϕ is the total weight of all satisfied constraints*

$$\mathrm{Val}_\Psi(\phi) = \sum_{(k,\vec{s},R)\in\mathcal{C}} [\phi(\vec{s}) \in R] \cdot \mathrm{wt}((k, \vec{s}, R)) .$$

The optimum *of Ψ is the maximum value of any assignment*

$$\mathrm{Opt}_\Psi = \max_{\phi:V\to D} \mathrm{Val}_\Psi(\phi) .$$

Problems considered in this thesis are either unweighted — or equivalently, assign the same weight to all constraints — or can easily be transformed into unweighted instances with little change in value by duplicating constraints and rounding weights. Therefore, we do not make distinction between these two cases in this thesis.

Also, it is without loss of generality to assume that the total weight of all constraints in an instance sums to 1. For unweighted problems, the value of an assignment $\mathrm{Val}_\Psi(\phi)$ is then simply the fraction of the constraints in Ψ that are satisfied by ϕ.

A class $\mathcal{P}$ of CSPs is simply a set of CSP instances. Many fundamental computational problems can be expressed as classes of CSP problems.

Example 3.2. *In* E3-SAT*, we have V as the set of variables. The variables take Boolean values, therefore $D = \{0, 1\}$. Each clause contains three literals of distinct variables ("E3" in* E3-SAT *stands for "exactly three"), and the clause is satisfied by some assignment if the value of the literals are not all* 0*. This can be translated into a constraint $(k, \vec{s}, R)$ where:*

- *The arity $k = 3$.*
- *The set of variables $\vec{s_i}$ contains three distinct variables.*
- *The set R_i contains assignments to variables in $\vec{s_i}$ that satisfy the clause. Therefore $R_i \subseteq \{0, 1\}^3$ and $|R_i| = 7$.*

Example 3.3. *To express* 3-COLORING *as* CSP*, let V be the set of vertices, $D = \{1, 2, 3\}$ be the set of colors, and for each edge $\{u, v\}$ in the graph, we have a constraint $(2, (u, v), D^2 - \{(1, 1), (2, 2), (3, 3)\})$.*

3.1.1 Decision versus Optimization

Given a CSP problem $\mathcal{P}$, the first question we would like to answer is whether we could decide if there is an assignment that satisfies all constraints. A more refined question would be to find an assignment with as large a value as possible. We use

Max-CSP to denote this optimization variant of CSP, or Max-$\mathcal{P}$ for some specific class $\mathcal{P}$ of CSP.

We say that an instance Ψ is *β-satisfiable* if the optimum $\mathrm{Opt}(\Psi) \geq \beta$, and when $\beta = 1$, we simply refer to it as being *satisfiable*. If we instead have $\mathrm{Opt}(\Psi) \leq \beta$, then we say that Ψ is *at-most-β-satisfiable*.

Much was known about the time complexity of such problems. Many natural decision problems, such as E3-Sat and 3-Coloring, are NP-complete. The classical paper by Schaefer gave a complete characterization for Boolean CSP decision problems [98]. Given the general consensus that $\mathsf{P} \neq \mathsf{NP}$, it seems unlikely that there are efficient algorithms that are able to decide the satisfiability of CSP problems.

CSP problems such as E3-Sat and 3-Coloring are connected to many other computational problems people encounter in both theory and practice. From the perspective of optimization, even though we could not hope to find, in a reasonable amount of time, a 3-coloring of a given graph such that no edge is monochromatic, we may still be happy with a 3-coloring if only 1% of the edges are violated.

Formally, we define the approximation ratio of an algorithm for some Max-CSP problem as the following.

Definition 3.4. *Let $\mathcal{P}$ be a class of* Max-CSP *problems, and $\mathcal{A}$ be an algorithm. We say that $\mathcal{A}$ is an α-approximation algorithm, for some $\alpha \in [0, 1]$, if for any $\Psi \in \mathcal{P}$, $\mathrm{Val}_\Psi(\mathcal{A}(\Psi)) \geq \alpha \,\mathrm{Opt}(\Psi)$.*

A Max-CSP *problem $\mathcal{P}$ is α-approximable if there is an α-approximation algorithm for it.*

For Max-CSP optimization problems, there is a decision variant that might look slightly easier, known as Gap-CSP.

Definition 3.5. *Let $0 \leq s < c \leq 1$, and $\mathcal{P}$ be a class of* Max-CSP*. We call c the* completeness *parameter, and s the* soundness *parameter.*

In the $\textsc{Gap}_{c,s}$-$\mathcal{P}$ problem, we are promised that for any given instance Ψ, either $\mathrm{Opt}(\Psi) \geq c$, or $\mathrm{Opt}(\Psi) \leq s$, and we are asked to decide which is the case. For $c = 1$, we usually drop c and just write $\textsc{Gap}_s$-$\mathcal{P}$.

It is straightforward to see that $\textsc{Gap}_{1,1}$-CSP is the usual CSP decision problem, and $\textsc{Gap}_{1,\beta}$-CSP is a natural generalization of it where we need to distinguish CSP instances that are satisfiable from those that are "far" from satisfiable. For a CSP problem $\mathcal{P}$, if we have an α-approximation algorithm $\mathcal{A}$, then we can solve the gap variant $\textsc{Gap}_{1,\beta}$-$\mathcal{P}$ for any $\beta < \alpha$ by running $\mathcal{A}$ and checking whether the solution given by $\mathcal{A}$ has value at least β. All results in this thesis in fact prove hardness for the gap version of the problems with some gap parameters c and s, which in turn imply that for any $\varepsilon > 0$, no $c/s + \varepsilon$-approximation algorithm exists for those problems unless $\mathsf{P} = \mathsf{NP}$, or some other standard complexity assumption fails.

For GraphColoring, instead of trying to find a 3-coloring that maximizes the fraction of non-monochromatic edges, we could consider the variant where we are

given graphs that are 3-colorable, and the goal is to find a valid coloring that uses as few colors as possible. The gap version of this problem would be to distinguish 3-colorable graphs from graphs that does not have valid coloring that uses a small number of colors. The computational complexity of these minimization problems sometimes behave quite differently, and we take a closer look at them and study their connections with Max-CSP in Part III.

3.1.2 A Simple Approximation Algorithm

Before we conclude this section, let us consider a very simple algorithm for Max-CSP.

Given a Max-CSP instance $\Psi = (V, D, \mathcal{C})$ with weight function $\mathrm{wt}(\cdot)$, the algorithm constructs assignment ϕ by picking a uniformly random value from D for each $v \in V$.

It is easy to analyze the performance of this algorithm. By linearity of expectation, the expected value of this assignment is

$$\frac{1}{|\mathcal{C}|} \sum_{(k, \vec{s}, R) \in \mathcal{C}} \frac{|R|}{|D|^{|\vec{s}|}} .$$

For Max-E3-Sat, the above works out to $\frac{7}{8}$.

Given the extreme mindlessness of the random assignment algorithm, one would hope that more sophisticated algorithms might give us better results. Surprisingly, Håstad proved that for many problems, including Max-E3-Sat, this is indeed the best possible, assuming $\mathsf{P} \neq \mathsf{NP}$ [51]. We discuss this phenomenon in more details in Section 3.2.

Of course, for many other problems, such as MaxCut, GraphColoring, and even many Boolean CSP problems, there are algorithms using convex optimization tools that achieves results better than the naive random assignment algorithm. We review some of them in later chapters.

3.2 Approximability of Max-k-CSP

We now look at Max-k-CSP$_q$ instances. A Max-CSP instance Ψ is a Max-k-CSP$_q$ instance if the domain of the variables has size q, and each constraint in Ψ involves at most k variables. Given a predicate $P : \mathbb{F}_q \to \{0, 1\}$, we define the Max-$P$ problem as follows.

Definition 3.6. *A* Max-k-CSP$_q$ *instance* $\Psi = (V, D, \mathcal{C})$ *is a* Max-P *instance if the following conditions are satisfied:*

- *The domain* $D = \mathbb{F}_q$.

- *For each constraint $(a, \vec{s}, R) \in \mathcal{C}$, we have that the arity $a = k$, the set of variables $\vec{s}$ contains k distinct variables, and there exists $b \in \mathbb{F}_q^k$, such that*

$$\forall x \in \mathbb{F}_q^k, x \in R \Leftrightarrow P(x + b) = 1\,.$$

  The vector b is sometimes called the shift *of constraint (a, s, R).*

Let E3-SAT: $\{0,1\}^3 \to \{0,1\}$ be a predicate that returns 1 unless all 3 input bits are 0. This gives us the MAX-E3-SAT problem mentioned in Section 3.1. The vector $b \in \mathbb{F}_q^3$ in each constraint determines the sign of each literal.

Many other classical problems can be viewed as MAX-P problems for some suitable predicate P:

- Ek-LIN: The predicate Ek-LIN: $\{0,1\}^k \to \{0,1\}$ is defined as

$$k\text{-LIN}(x_1, \ldots, x_k) = \sum_{i=1}^{k} x_i\,.$$

  A Ek-LIN instance is precisely a system of linear equations over $\mathbb{F}_2$, in which each equation contains exactly k distinct variables.

- NOTTWO: The predicate NOTTWO : $\{0,1\}^3 \to \{0,1\}$ accepts input string (x_1, x_2, x_3) iff exactly two of the three bits are 1.

- HADAMARD$_K$: This predicate is defined for $K = 2^k - 1$ for $k \in \mathbb{N}^+$. The predicate is defined on K Boolean variables, indexed by non-empty subsets of $[k]$. The predicate accepts input $\left\{x^{(S)}\right\}_{\emptyset \neq S \subseteq [k]}$ iff for all $S \subseteq [k]$, $|S| \geq 2$, we have

$$x^{(S)} = \sum_{i \in S} x^{(\{i\})}\,.$$

- NOTALLEQUAL$_k$: The predicate NOTALLEQUAL$_k$: $\{0,1\}^k \to \{0,1\}$ returns 1 if the k input bits are not all equal, and 0 otherwise.

- TSA: The Tri-Sum-And predicate[1] is defined on 5 Boolean variables

$$\text{TSA}(x_1, x_2, x_3, x_4, x_5) = 1 + x_1 + x_2 + x_3 + x_4 \cdot x_5\,.$$

For a predicate $P : \mathbb{F}_q^k \to \{0,1\}$, it is natural to view it as a set $P \subseteq \mathbb{F}_q^k$ containing inputs on which P evaluates to 1. We call this the set of *accepting*

[1] Not to be confused with the Transportation Security Administration. The Tri-Sum-And predicate was first studied by Håstad and Khot [55]. The conference version of [55] appeared in FOCS 2001, which was held October 8 – 11.

inputs of P. One important parameter is its *density*, defined as $\rho(P) := |P|/q^k$. For the predicates we have encountered so far, we list their density as follows:

$$\begin{aligned}
\rho(\text{E3-Sat}) &= 7/8\,,, \\
\rho(\text{E}k\text{-Lin}) &= 1/2, \quad \forall k \in \mathbb{N}^+\,, \\
\rho(\text{NotTwo}) &= 5/8\,, \\
\rho(\text{Hadamard}_K) &= (K+1)/2^K, \quad \forall K = 2^k - 1, k \in \mathbb{N}^+\,, \\
\rho(\text{NAE}_k) &= 1 - 2^{1-k}, \quad \forall k \in \mathbb{N}^+\,, \\
\rho(\text{TSA}) &= 1/2\,.
\end{aligned}$$

As mentioned in Section 3.1.2, if we use the random assignment algorithm to solve Max-P, the expected fraction of satisfied constraints is exactly $\rho(P)$. Surprisingly, it turns out that for some predicates, this simple algorithm gives the best approximation guarantee assuming $\mathsf{P} \neq \mathsf{NP}$. We say that a predicate P is *approximation resistant* if it is hard to achieve an approximation ratio strictly larger than $\rho(P)$. In the language of Gap-CSP, predicate P is approximation resistant if $\text{Gap}_{1-\varepsilon,\rho(P)+\varepsilon}$-$P$ is NP-hard. In a celebrated result, Håstad [51] showed that $\text{Gap}_{1-\varepsilon,1/2+\varepsilon}$-E$k$-Lin is NP-hard. That is, it is NP-hard to find an assignment satisfying more than a $1/2+\varepsilon$ fraction of the constraints for any $\varepsilon > 0$, even when the input has an assignment that satisfies $1-\varepsilon$ of them.

There has been much progress in understanding what kinds of predicates are approximation resistant. Most of the results study predicates on domain of size 2, i.e. Boolean predicates. Some of the algorithms and hardness results can be generalized to larger domains, but for clarity of presentation, in the remaining of this section, we focus on the case of Boolean predicates since this is the most well-studied case and serves well to illustrate the key issues of the Max-CSP approximability. Also, we only look at Boolean predicates of arity $k \geq 3$, since none of the predicates with arity less than 3 is approximation resistant [52].

For predicates of small arity (3 or 4), both algorithms and hardness results are studied, and there is a complete characterization of approximability for predicates of arity 3 [51, 106] and an almost complete one for predicates of size 4 thanks to an extensive study by Gustav Hast [49]. For predicates of higher arities, a handful of predicates were shown to be approximation resistant [51, 96, 49, 36].

The scenario where the random assignment algorithm does poorly is when $\rho(P)$ is small, or in other words, the predicate P has few accepting inputs compared to the possible inputs. Intuitively, such CSP instances are very restrictive, therefore if we are given a Max-P instance Ψ with the promise that it is satisfiable (or $(1-\varepsilon)$-satisfiable for some small $\varepsilon > 0$), one might hope that there are smarter algorithms that take advantage of the structure of the predicate P. For satisfiable instances, the algorithm by Trevisan [101] shows that if $|P| < k+1$, then $\text{Gap}_{\rho(P)+\varepsilon}$-$P$ is in P for some $\varepsilon > 0$. Hast [50] proved that even without the guarantee that the instances are satisfiable, $\text{Gap}_{1-\varepsilon,\rho(P)+\varepsilon}$-$P$ is still in P as long as $|P| \leq 2\lfloor k/2 \rfloor + 1$. More generally, Charikar, Makarychev and Makarychev [25] gave a $ck/2^k$-approximation

algorithm for MAX-k-CSP, for some $c > 0.44$. Makarychev and Makarychev [83] later improved the constant c to ≈ 0.62.

These algorithmic results are tight up to multiplicative constant factor as $k \to \infty$. In [97], Samorodnitsky and Trevisan showed the approximation resistance of HADAMARD_K for any $K = 2^k - 1$ assuming the Unique Games Conjecture (UGC). The Unique Games Conjecture was proposed by Khot in 2002 [69], and has become one of the most important open problems in theoretical computer science. We discuss more about the UGC and related problems in Section 3.3.1. Austrin and Mossel [10] proved that assuming the UGC, P is approximation resistant if P contains the support of a pairwise independent distribution. In a recent breakthrough [24], Siu On Chan settled the NP-hardness of MAX-HADAMARD_K (and, up to a constant factor, MAX-k-CSP in general), bypassing the UGC. Together with the result in [53], this shows that almost all Boolean predicates are approximation resistant.

Let us pause for a minute and take another look at approximation resistant predicates such as Ek-LIN and HADAMARD_K. Instances of these problems are just systems of linear equations, and we can decide satisfiability for these problems in polynomial time by Gaussian elimination. However, as we have explained above, for both MAX-Ek-LIN and MAX-HADAMARD_K, even for instances that are *almost* fully satisfiable, Gaussian Elimination is no longer applicable and the best alternative is to just pick a random assignment. In some sense, this says that Gaussian Elimination is not a very robust method. This is in contrast to other MAX-CSP algorithms that are based on Linear Programming (LP) or Semi-definite Programming (SDP), such as the one by [25]. The LP/SDP based algorithms seem to be of a very different nature, and approximation guarantees for most of them does not depend critically on the satisfiability of the input instances.

This also demonstrates that the notion of approximation resistance is still far from capturing the full picture of the computational complexity of MAX-CSP. It is possible that for satisfiable instances, generalizations of Gaussian Elimination or some entirely new techniques are yet to be discovered. The same is true for the hardness of GAP_β-CSP problems. In contrast to our understanding of approximation resistance as demonstrated above, approximation resistance on satisfiable instances is still largely a mystery. There have been only a handful of results, and the best soundness we have for $\text{GAP}_{1,s}$-CSP is still very far from the algorithmic guarantees and what we have for $\text{GAP}_{1-\varepsilon,s}$-CSP. In Part II, we elaborate on the technical challenges of understanding the computational complexity for $\text{GAP}_{\rho(P)}$-P and show some results that partially get around these limitations.

Understanding the computational complexity of GAP-CSP is not just interesting in its own right. Many hardness results for other combinatorial optimization problems use GAP-CSP inapproximability results as a starting point, and for some applications such as GRAPHCOLORING, having perfect completeness is believed to be crucial. Results in Part III provide an example how improved understanding for $\text{GAP}_{1,s}$-CSP lead to improved inapproximability results for graph and hypergraph coloring.

3.3 Label-Cover

In this section, we consider another class of CSPs — Label-Cover. It is a very important class of CSP problem in the study of hardness of approximation, because it is often used as the starting point for proving strong inapproximability results. Many variants of Label-Cover have been constructed for proving approximation hardness for different kinds of combinatorial optimization problems.

We start with the definition of a basic Label-Cover instance.

Definition 3.7. *A* projective Label-Cover *instance $\mathcal{L}$ is defined by a tuple*

$$\mathcal{L} = (U, V, E, L, R, \Gamma, \Pi)\,.$$

Here U and V are two disjoint sets of vertices of a bipartite multigraph, E is the set of edges between them. The sets L and R are label sets for vertices in U and V, respectively. For each $v \in V$, there is a subset of R indicating the labels allowed for v, given by function $\Gamma : V \to \mathcal{P}(R)$. The set Π is a collection of projections, one for each edge e, $\pi_e : R \to L$.

A labeling $\sigma = (\sigma_U, \sigma_V)$ of the Label-Cover *instance $\sigma_U : U \to L$, $\sigma_V : V \to R$ is* valid *iff $\sigma_V(v) \in \Gamma(v)$ for all $v \in V$. An edge $\{u, v\}$ is satisfied by a labeling σ if $\pi_{\{u,v\}}(\sigma(v)) = \sigma(u)$. The value of a valid labeling $\mathrm{Val}_{\mathcal{L}}(\sigma)$ is the fraction of edges that are satisfied by σ. The value of $\mathcal{L}$ is the maximum value of all possible valid labelings.*

Remark. *The term "projective" refers to the projection constraints in Π. We can define* Label-Cover *more generally by allowing arbitrary relations in Π. The result by Dinur, Mossel and Regev on hardness of* GraphColoring *[32] is a good example of hardness of approximation results obtained from non-projective* Label-Cover *instances.*

In this thesis, however, we only use projective Label-Cover*, thus from now on, we refer to those simply as* Label-Cover*.*

Remark. *In many parts of the thesis — particularly when we use* Label-Cover *together with* Long-Code *— we may assume that the label set R only contains valid labels. In those cases, we omit Γ from the notation.*

The construction of the Label-Cover used in many hardness of approximation results starts from the famous PCP theorem.

Theorem 3.8 (PCP Theorem [6, 7])**.** *There exists a constant $\delta < 1$, such that* $\textsc{Gap}_{1,\delta}$-E3-Sat *is* NP*-hard.*

PCP stands for "Probabilistically Checkable Proofs", and the concept comes from research in interactive proofs. Imagine that for a certain problem, there is an all powerful *prover* that provides certificates/proofs for the instances of the problem, and a *verifier*, usually with limited computational resources, that is supposed

to check whether the certificate is valid and decide the answer for the problem instances. Recall that NP can be defined as the class of problems for which there is a deterministic polynomial time verifier that always accepts correct certificates and rejects incorrect ones. To achieve this, it seems necessary that the verifier reads through the whole certificate. In its original formulation, the PCP theorem states that for every problem in NP, there is a *probabilistic* verifier that uses $O(\log n)$ random bits and reads only a constant number of symbols in the certificate, accepts all correct proofs that are written in some special format, and rejects incorrect ones with constant probability.

The formulation in Theorem 3.8 above gives a polynomial time reduction from E3-SAT to $\text{GAP}_{1,\delta}$-E3-SAT and thus gives exactly such a verifier. Given an E3-SAT formula ϕ, the verifier runs the above reduction in polynomial time to get a new E3-SAT formula ψ, and expects a proof that contains a satisfying assignment for ψ. The verifier can now simply pick a random clause C in ψ and read from the proof the value of the three variables in C and check if those assignments satisfy C. If ϕ is unsatisfiable, then any assignment for ψ violates at least a $1-\delta$ fraction of the clauses in ψ, thus the verifier catches an incorrect proof for unsatisfiable formulas with probability $1-\delta$.

The PCP Theorem gives NP-hardness for $\text{GAP}_{1,\delta'}$-LABEL-COVER for some $\delta' < 1$ by the following construction: given an E3-SAT instance, construct a bipartite graph (U, V, E), where U corresponds to the variables and V corresponds to the clauses, and there is an edge between u and v if variable u appears in clause v. The label set $L = \{0, 1\}$ are the Boolean assignments for the variables, and $R = \{0, 1\}^3$ are the combination of assignments for the clauses. For each clause $v \in V$, $\Gamma(v)$ contains the set of local assignments to variables in v that satisfies clause v. The constraint $\pi_{\{u,v\}}$ between variable u and clause v checks that the assignment to variable u agrees with the one assigned in clause v. It is easy to see that if the E3-SAT instance is satisfiable, then the natural labeling according to a satisfying assignment has value 1 for the LABEL-COVER instance. For the soundness part, suppose the E3-SAT instance is at most δ satisfiable. Consider any labeling of the LABEL-COVER instance. The labeling for vertices in U gives an assignment to the variables in the E3-SAT instance. Since the instance is at most δ satisfiable, at least $(1-\delta)$ fraction of the clauses are not satisfied, therefore any labeling for those clause variables in V will be inconsistent with at least 1 out of 3 of its neighbors in U. This means that if the E3-SAT instance is at most δ satisfiable, then the LABEL-COVER instance is at most $1-(1-\delta)/3$ satisfiable.

The LABEL-COVER problem is sometimes also referred to as a 2-Prover-1-Round (2P1R) Game. The 2 players, Alice and Bob agrees on a strategy before the game starts. The referee then draws a pair of questions (u, v), sends u to Alice, and v to Bob. No communication is allowed between Alice and Bob. Alice replies with $A(u)$, and Bob replies with $B(v)$, and the referee checks whether $A(u)$ and $B(v)$ satisfies certain relation that is known to all parties. If the relation is such that for any answer from Bob, there is only one acceptable answer for Alice, then this is called a *Projection Game*. It is easy to see that projection games exactly correspond to

(projective) Label-Cover instances. If in addition for each answer from Alice, there are exactly d possible consistent answers from Bob, this is called a *d-to-1 Game*. Finally, a d-to-1 game with $d = 1$ is also known as a *Unique Game*. We will elaborate a bit more on d-to-1 Games and Unique Games later in this section.

To boost the soundness of the Label-Cover hardness we get from Theorem 3.8, we apply the Parallel Repetition Theorem for 2P1R games. The Parallel Repetition Theorem was first proved by Ran Raz [93], and later strengthened and simplified by Thomas Holenstein [57], and Anup Rao [92]. Let Ψ be an instance of a 2P1R game. The idea of an m-round parallel repetition is that the referee now picks m pairs of questions $(u_1, v_1), \ldots, (u_m, v_m)$, sends questions $\{u_1, \ldots, u_m\}$ to Alice and $\{v_1, \ldots, v_m\}$ to Bob. Alice and Bob reply with their answers to each of the m questions, and the referee accepts if for all $i = 1, \ldots, m$, the pair of answer $A(u_1, \ldots, u_m)_i$ and $B(u_1, \ldots, u_m)_i$ satisfies the relation in the original game Ψ. We denote the repeated game by Ψ^m.

Here, we use the version proved by Anup Rao [92], which applies specifically to projection games with better dependencies on parameters. In particular, the rate at which the soundness decreases is independent of the alphabet size of the original game.

Theorem 3.9 (Parallel Repetition [92])**.** *There is a universal constant $\alpha > 0$, such that for a* Label-Cover *instance Ψ, if $\mathrm{Opt}(\Psi) \leq 1 - \varepsilon$, then $\mathrm{Opt}(\Psi^m) \leq (1 - \varepsilon/2)^{\alpha \varepsilon m}$.*

Combining Theorem 3.8 and 3.9, we have the following NP-hardness theorem for the basic Label-Cover problem.

Theorem 3.10. *Let $\varepsilon = \varepsilon(n) > 0$, and let $t := C_0 \log 1/\varepsilon$, where $C_0 > 0$ is a universal constant. There is a reduction that takes as input a* E3-Sat *instance of size n, and outputs a* Label-Cover *instance $\mathcal{L} = (U, V, E, L, R, \Gamma, \Pi)$ with the following properties:*

- *The bipartite graph has size $n^{O(t)}$. The degree of the vertices in the bipartite graph is at most $\exp(O(t))$.*
- *The label set $L = \{0,1\}^t$, and $R = \{0,1\}^{3t}$.*
- *For each $v \in V$, there are t polynomials $p_{v,1}, \ldots, p_{v,t} : \{0,1\}^3 \to \{0,1\}$, and t triples $s_{v,1}, \ldots, s_{v,t} \in \binom{3t}{3}$, such that*

$$\forall x \in R \ (x \in \Gamma(v) \quad \Leftrightarrow \quad \forall j \in [t], p_{v,j}(s_{v,j}) = 0) .$$

  Moreover, $|\Gamma(v)| = 7^t$.
- *For each edge $e \in E$, there is a subset of indices $S \subseteq [3t]$, $|S| = t$, such that for any $r \in R$, $\pi_e(r) = r|_S$.*
- *If the* E3-Sat *instance is satisfiable, then $\mathrm{Opt}(\mathcal{L}) = 1$.*

- *If the* E3-SAT *instance is not satisfiable, then* $\mathrm{Opt}(\mathcal{L}) \leq \varepsilon$.

The reduction runs in time $n^{O(t)}$.

The SMOOTH-MULTI-LAYERED-LABEL-COVER problem is a variant of LABEL-COVER first studied in Khot [68] for showing hardness of coloring 3-colorable 3-uniform hypergraphs. We start with the definition of smoothness.

Definition 3.11 (Smoothness)**.** *A* LABEL-COVER *instance is* ξ*-smooth if for any vertex* $v \in V$ *and any two labels* $r \neq r' \in R$*, over a uniformly random neighbor* u *of* v*, we have*

$$\Pr_{u \sim v}[\pi_{\{u,v\}}(r) = \pi_{\{u,v\}}(r')] \leq \xi\,. \tag{3.1}$$

Similar to LABEL-COVER, we have the following hardness result for SMOOTH-LABEL-COVER.

Theorem 3.12. *Let* $\varepsilon = \varepsilon(n) > 0$*,* $\xi = \xi(n) > 0$*, and* $t_0 := C_0 \cdot \log 1/\varepsilon$*,* $t_1 := t_0/\xi$*, where* $C_0 > 0$ *is a universal constant. There is a reduction that takes as input a* E3-SAT *instance of size* n*, and outputs a* SMOOTH-LABEL-COVER *instance* $\mathcal{L} = (U, V, E, L, R, \Gamma, \Pi)$ *with the following properties:*

- *The bipartite graph has size* $n^{O(t_1)}$*. The degree of the vertices in the bipartite graph is at most* $\exp(O(t_1))$.
- *The label set* $L = \{0,1\}^{3(t_1 - t_0) + t_0}$*, and* $R = \{0,1\}^{3t_1}$.
- *For each* $v \in V$*, there are* t_1 *polynomials* $p_{v,1}, \ldots, p_{v,t_1} : \{0,1\}^3 \to \{0,1\}$*, and* t_1 *triples* $s_{v,1}, \ldots, s_{v,t_1} \in \binom{3t_1}{3}$*, such that*

  $$\forall x \in R\ (x \in \Gamma(v) \quad \Leftrightarrow \quad \forall j \in [t_1], p_{v,j}(s_{v,j}) = 0)\,.$$

  Moreover, $|\Gamma(v)| = 7^{t_1}$.
- *For each edge* $e \in E$*, there is a subset of indices* $S \subseteq [3t_1]$*,* $|S| = 3(t_1 - t_0) + t_0$*, such that for any* $r \in R$*,* $\pi_e(r) = r|_S$.
- *If the* E3-SAT *instance is satisfiable, then* $\mathrm{Opt}(\mathcal{L}) = 1$.
- *If the* E3-SAT *instance is not satisfiable, then* $\mathrm{Opt}(\mathcal{L}) \leq \varepsilon$.
- *The instance* $\mathcal{L}$ *is* ξ*-smooth.*

The reduction runs in time $n^{O(t_1)}$.

Proof. Let $t := 1/\xi$. Hence $t_1 = t \cdot t_0$.

Let $\mathcal{L}_0 = (U_0, V_0, E_0, L_0, R_0, \Gamma_0, \Pi_0)$ be the LABEL-COVER instance we get from Theorem 3.8. The vertex set V of the SMOOTH-LABEL-COVER instance we are going to construct consists of all t-tuples of V_0 vertices, and the vertex set

U consists of all t-tuples of vertices with exactly 1 vertex from U_0. For each $(v_1, \dots, v_t) \in V$, index $i \in [t]$, and neighbor $u \in N(v_i)$ of v_i in $\mathcal{L}$, add an edge between $(v_1, \dots, v_{i-1}, u, v_{i+1}, \dots, v_t)$ and $(v_1, \dots, v_t)$. The projection constraint naturally requires the labelings to be identical on coordinate $[t] - \{i\}$, and for the i-th coordinate satisfy the projection constraint $\pi_{u,v_i} \in \Pi_0$. As for Γ, a labeling for $(v_1, \dots, v_t)$ is valid if it is valid in $\mathcal{L}_0$ for each of $v_1, \dots, v_t$.

The completeness and soundness properties are all straightforward. As for smoothness, observe that for $r, r' \in R = R_0^t$, if we have $r \neq r'$, then there must be at least one $j \in [t]$ such that $r_j \neq r'_j$. Unless index j is chosen, the projection on the j-th coordinate will be the identity projection and thus r and r' will be projected to different labels in L. This happens with probability at least $1 - 1/t = 1 - \xi$. □

Multi-Layered-Label-Cover was first devised in [30] to prove strong approximation hardness for hypergraph vertex cover, and used in [36] for improving query efficiency of PCPs and hardness of approximation of Max-CSP. Briefly speaking, a normal Label-Cover instance checks consistency of labeling between a pair of vertices, whereas in a k-Layered-Label-Cover instance, we consider tuples of $k-1$ independently sampled edges

$$(\{u_1, v_1\}, \{u_2, v_2\}, \dots, \{u_{k-1}, v_{k-1}\}) \,,$$

the k hybrid tuples of vertices

$$(u_1, \dots, u_i, v_{i+1}, \dots, v_{k-1}), \quad \text{for } i = 0, \dots, k-1 \,,$$

and their corresponding labelings, and we require consistency between all pairs of tuples. Formally, given a Label-Cover instance as defined above, the constraint between pairs of labelings on tuples is defined as follows.

Definition 3.13. *Let $\vec{e} = (e_1, \dots, e_{k-1}) \in E^{k-1}$ be a vector, and let $1 \leq i < j \leq k$. Define the mapping $\pi_{\vec{e}, j \to i} : L^{k-j} \times R^{j-1} \to L^{k-i} \times R^{i-1}$ as*

$$\begin{aligned} & (l_1, \dots, l_{k-j}, r_{k-j+1}, \dots, r_{k-1}) \\ \mapsto\ & (l_1, \dots, l_{k-j}, \pi_{e_{k-j+1}}(r_{k-j+1}), \dots, \pi_{e_{k-i}}(r_{k-i}), r_{k-i+1}, \dots, r_{k-1}) \,. \end{aligned}$$

It is not hard to see that the above definition preserves smoothness in the Multi-Layered-Label-Cover instances.

Lemma 3.14. *For any k-Layered-Label-Cover instance constructed from a ξ-Smooth-Label-Cover instance $(U, V, E, L, R, \Gamma, \Pi)$, any positive integer $1 < i \leq k$, vertex tuple $\vec{u} = (u_1, \dots, u_{k-1}) \in U^{k-i} \times V^{i-1}$, two tuples of labelings $\vec{r} \neq \vec{r'} \in L^{k-i} \times R^{i-1}$, we have*

$$\Pr_{\vec{e} \sim \vec{u}} [\pi_{\vec{e}, i \to 1}(\vec{r}) = \pi_{\vec{e}, i \to 1}(\vec{r'})] < \xi \,,$$

where we sample $\vec{e}$ by picking each $e_i \sim u_i$ independently.

Proof. If there exists $j \in \{1, \dots, k-i\}$ such that $r_j \neq r'_j$, then we always have

$$\pi_{\vec{e}, i\to 1}(\vec{r}) \neq \pi_{\vec{e}, i\to 1}(\vec{r'})$$

and hence the above inequality holds for any $\xi > 0$.

We now assume that for all $j \in \{1, \dots, k-i\}$, we have $r_j = r'_j$, and that there exists $j_0 \in \{k-i+1, \dots, k-1\}$ such that $r_j \neq r'_j$. Observe that

$$\pi_{\vec{e}, i\to 1}(\vec{r}) = \pi_{\vec{e}, i\to 1}(\vec{r'})$$

implies that for all $j \in \{k-i+1, \dots, k-1\}$, we have $\pi_{e_j}(r_j) = \pi_{e_j}(r'_j)$, and in particular

$$\pi_{e_{j_0}}(r_{j_0}) = \pi_{e_{j_0}}(r'_{j_0}).$$

By definition of smoothness, this happens with probability less than ξ. □

In many applications, it is often easier to work with a slightly stronger notion of smoothness. We extend the definition of projection $\pi : R \to L$ to sets of labels $S \subseteq R$ by $\pi(S) := \{l \in L \mid \exists r \in S, \pi(r) = l\}$.

Definition 3.15. *A* Label-Cover *instance is (J, ξ)-smooth if for any vertex $v \in V$ and any set of labels $S \subset R$, $|S| \leq J$, over a uniformly at random neighbor u of v, we have*

$$\Pr_{u\sim v}\left[|\pi_{\{u,v\}}(S)| < |S|\right] \leq \xi. \tag{3.2}$$

*Similarly, a k-*Layered-Label-Cover *instance is (J, ξ)-smooth if for any integer $1 < i \leq k$, vertex tuple $\vec{u} = (u_1, \dots, u_{k-1}) \in U^{k-i} \times V^{i-1}$, and set of labelings $S \subseteq L^{k-i} \times R^{i-1}$ with $|S| \leq J$, we have*

$$\Pr_{\vec{e}\sim\vec{u}}\left[|\pi_{\vec{e}, i\to 1}(S)| < |S|\right] \leq \xi.$$

Observe that $|\pi_e(S)| < |S|$ if and only if there exists $r \neq r' \in S$ such that $\pi_e(r) = \pi_e(r')$. By simple union bound over all possible pairs of labelings in S, we can show that for constant J, the above two notions of smoothness differs only by a constant factor. The same argument applies to multi-layered instances.

Lemma 3.16. *A ξ-smooth k-layered* Label-Cover *instance is $(J, \binom{J}{2}\cdot\xi)$-smooth.*

Combining all we have, we get the following hardness result for Smooth-k-Multi-Layered-Label-Cover. In the statement below, we omit certain size parameters that were stated in Theorem 3.10 and Theorem 3.12 since they are not needed in the rest of the thesis.

Theorem 3.17. *Let $\varepsilon = \varepsilon(n) > 0$, $\xi = \xi(n) > 0$, $J > 0$, and $t_0 := C_0 \cdot \log 1/\varepsilon$, $t_1 := t_0/(J^2\xi)$, where $C_0 > 0$ is a universal constant. There is a reduction that takes as input a* E3-Sat *instance of size n, and outputs a* Smooth-k-Multi-Layered-Label-Cover *instance $\mathcal{L}$ with the following properties:*

- *The size of* $\mathcal{L}$ *is* $n^{O(t_1 k)}$.
- *The base label set* $L = \{0,1\}^{3(t_1-t_0)+t_0}$, *and* $R = \{0,1\}^{3t_1}$. *The label set for layer* i *is* $L^{k-i} \times R^{i-1}$.
- *If the* E3-SAT *instance is satisfiable, then there exist assignments* $\sigma_m : U^{k-m} \times V^{m-1} \to L^{k-m} \times R^{m-1} (1 \le m \le k)$, *such that for all* $\vec{e} = (e_1, \ldots, e_{k-1}) \in E^{k-1}$ *and all* $i, j, 1 \le i < j \le k$, *it holds that*

$$\begin{aligned} &\pi_{\vec{e}, j \to i}(\sigma_j(u_1, \ldots, u_{k-j}, v_{k-j+1}, \ldots, v_{k-1})) \\ &= \sigma_i(u_1, \ldots, u_{k-i}, v_{k-i+1}, \ldots, v_{k-1}). \end{aligned}$$

- *If the* E3-SAT *instance is not satisfiable, then there are no two integers* l *and* h $(1 \le l < h \le k)$ *such that there exist functions* $P_h : U^{k-h} \times V^{h-1} \to L^{k-h} \times R^{h-1}$ *and* $P_l : U^{k-l} \times V^{l-1} \to L^{k-l} \times R^{l-1}$, *such that for more than* ε *fraction of* $(e_1, \ldots, e_{k-1}) \in E^{k-1}$, *we have*

$$\begin{aligned} &\pi_{\vec{e}, l \to 1}(P_l(u_1, \ldots, u_{k-l}, v_{k-l+1}, \ldots, v_{k-1})) \\ = \ &\pi_{\vec{e}, h \to 1}(P_h(u_1, \ldots, u_{k-h}, v_{k-h+1}, \ldots, v_{k-1})). \end{aligned} \tag{3.3}$$

  If an edge tuple $(e_1, \ldots, e_{k-1})$ *satisfies the above condition, we say that it is* weakly satisfied.

- *The instance* $\mathcal{L}$ *is* (J, ξ)*-smooth.*

The reduction runs in time $n^{O(t_1 k)}$.

Proof. The proof is similar to [36]. We start with a SMOOTH-LABEL-COVER instance and turn it into a SMOOTH-k-MULTI-LAYERED-LABEL-COVER instance as described above.

The completeness case is straightforward. For soundness, suppose there exists $1 \le l < h \le k$ and functions P_l, P_h, such that (3.3) holds for more than η fraction of $(e_1, \ldots, e_{k-1}) \sim E^{k-1}$. Pick any coordinate $i \in \{k-h+1, \ldots, k-l\}$. Then there is a way to fix edges $e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_{k-1}$, such that (3.3) holds for at least ε fraction of the edges e_i. We conclude the proof by noting that the restriction of P_l and P_h on the i-th coordinate gives a labeling with value at least ε for the original SMOOTH-LABEL-COVER instance. □

3.3.1 The Unique Games Conjecture

The discovery of the PCP Theorem and subsequent improvements have led to many hardness of approximation results, many of which are optimal unless $\mathsf{P} = \mathsf{NP}$. Despite this, some problems turned out to be quite resilient, examples include MINIMUMVERTEXCOVER, GRAPHCOLORING, MAXCUT, and many optimization problems on graphs. In 2002, Khot introduced a conjecture known as the *Unique Games*

Conjecture (UGC) as a possible solution [69]. The UGC had inspired a remarkable number of works, and many optimal inapproximability results has been achieved assuming UGC, including the optimality of the Goemans-Williamson algorithm for MaxCut [71], a $2-\varepsilon$ hardness for MinimumVertexCover [73], optimal approximability for Max-E2-Sat [8], approximation resistance for all ordering problems [43], characterization of approximation resistant predicates [10, 79, 53], to name just a few. In a beautiful result [91], Raghavendra shows that for every Max-CSP, the integrality gap for a natural SDP relaxation is the same as the inapproximability threshold for the CSP, assuming the UGC. In other words, the natural SDP relaxation gives the best efficient approximation algorithm.

The UGC has become one of the most important open problems in theoretical computer science. We refer to a comprehensive survey by Khot [70] for more historical context, the implications of UGC, and its connections with other related questions.

Much effort has been devoted to proving or disproving UGC. In 2005, Khot and Vishnoi proved that the canonical SDP algorithms fail to solve Unique-Games [80, 81]. Similar results demonstrating the limitations of other LP and SDP algorithms were subsequently proved in [75, 13]. From the other side, Arora, Khot, Kolla, Steurer, Tulsiani and Vishnoi showed that Unique-Games is easy on expanding constraint graphs [5], which implies that Unique-Games is easy for random instances. This is in contrast to NP-hard problems such as E3-Sat, for which people believe that even random instances are hard. In 2010, Arora, Barak and Steurer gave a subexponential time algorithm for Unique-Games [3]. Recently in [12], it was shown that the Sum-of-Squares method of constant degree solves the instances that are hard for algorithms in [81, 75, 13]. To sum up, although there is no polynomial time algorithm that solves Unique-Games in the worst case, there is no class of Unique-Games instances that are hard for current algorithmic techniques either. We refer to [14] for a survey on recent works.

Earlier in Section 3.3, we formulated Unique-Games as a special case of 2-Prover-1-Round games. We now formally define the Unique-Games Problem as follows.

Definition 3.18. *A* Unique-Games *instance is a* Label-Cover *instance*

$$\mathcal{U} = (U, V, E, L, R, \Pi)\,,$$

where the label sets $L = R$, and the edge projections in Π are permutations.

We can specify a Unique-Games *instance by a tuple (U, V, E, L, Π).*

Let $\mathcal{U}$ be a Unique-Games instance. If $\mathrm{Opt}(\mathcal{U}) = 1$, we can find such a labeling in polynomial time by trying out all labels for one arbitrary vertex in each connected component in $\mathcal{U}$, and then propagate the value to the whole connected component with the permutation constraints, and start over with a different label if at some point we reach a conflict. The Unique Games Conjecture states that when $\mathrm{Opt}(\mathcal{U}) = 1-\varepsilon$ for some small constant ε, it is very hard to find a good labeling.

Conjecture 3.19 ([69]). *For every $\varepsilon, \delta > 0$, there exists a constant $l = l(\varepsilon, \delta)$, such that given a* Unique-Games *instance $\mathcal{U} = (U, V, E, L, \Pi)$ with $|L| = l$, it is* NP*-hard to distinguish between the following two cases:*

- *Completeness:* $\mathrm{Opt}(\mathcal{U}) \geq 1 - \varepsilon$.
- *Soundness:* $\mathrm{Opt}(\mathcal{U}) \leq \delta$.

In many cases, having perfect completeness — where we have $\mathrm{Opt}(\mathcal{U}) = 1$ in the completeness case — is desirable. This motivates the following d-to-1 game.

Definition 3.20. *Let L, R be label sets such that $|R| = d|L|$. A projection $\pi : R \to L$ is d-to-1 if for all $l \in L$, the pre-image $|\pi^{-1}(l)| = d$.*

For a fixed d, a Label-Cover *instance (U, V, E, L, R, Π) is a d-to-1* Label-Cover *instance if $|R| = d|L|$ and all mappings $\pi_e \in \Pi$ are d-to-1.*

The following set of conjectures (one for each d) is known as the d-to-1 Conjectures.

Conjecture 3.21 ([69]). *For every $\delta > 0$, there exists a constant $l = l(\delta)$, such that given a d-to-1* Label-Cover *instance $\mathcal{L} = (U, V, E, L, R, \Pi)$ with $|L| \leq |R| = l$, it is* NP*-hard to distinguish between the following two cases:*

- *Completeness:* $\mathrm{Opt}(\mathcal{L}) = 1$.
- *Soundness:* $\mathrm{Opt}(\mathcal{L}) \leq \delta$.

Remark. *The d-to-1* Label-Cover *problem becomes harder as d increases. In particular, Theorem 3.10 implies that we in fact have* NP*-hardness for $d = \mathrm{poly}(1/\delta)$.*

Assuming the 2-to-1 Conjecture, Dinur, Mossel and Regev [32] proved that it is NP-hard to q'-color a q-colorable graph for any $4 \leq q < q'$, and a similar hardness result for $3 = q < q'$ assuming a variant of d-to-1 Conjecture. O'Donnell and Wu proved optimal hardness for Max-3-CSP assuming the d-to-1 Conjectures for any d [90]. As we will discuss in more detail in Chapter 4, some problems that were proved to be hard assuming the d-to-1 Conjectures, such as the one by O'Donnell and Wu, were later proved to be hard only assuming $\mathsf{P} \neq \mathsf{NP}$.

It is worth noting that compared to the UGC, there have been far fewer results assuming the d-to-1 Conjectures. It seems that the chasm between "1-to-1" and "2-to-1" is much wider than that between "2-to-1" and "poly$(1/\delta)$-to-1". This perhaps illustrates some of the limitations of current techniques in handling d-to-1 projections for $d > 1$.

Part II

Gap_s-CSP and PCPs with Perfect Completeness

人皆知有用之用
而莫知无用之用也

—— 庄子 · 人间世

Chapter 4

Predicates Strictly Dominating Ek-LIN

We give an introduction to proving inapproximability results and demonstrate how the machinery described in Part I are typically assembled.

As we have discussed in Section 3.2, deciding $\text{GAP}_{1-\varepsilon,1/2+\varepsilon}$-E$k$-LIN is NP-hard, but the problem becomes solvable in polynomial time by Gaussian Elimination when the completeness parameter becomes 1. One could ask if this is still true for some other predicate P that looks similar to Ek-LIN. In this chapter, we prove that for any $k \geq 4$ and k-arity predicate P that strictly contains Ek-LIN, the gap problem $\text{GAP}_{\rho(P)+\varepsilon}$-$P$ is hard, assuming the d-to-1 Conjectures.

The reduction is very similar to that in [90], where the authors proved hardness for $\text{GAP}_{5/8+\varepsilon}$-NOTTWO assuming the d-to-1 Conjecture. In fact, we can view the NOTTWO predicate as accepting input $(1,1,1)$ in addition to what was accepted by the E3-LIN predicate. Note that throughout this chapter and Chapter 5, we use the $-1/1$ notation rather than $0/1$.

For both GAP-NOTTWO and the result presented in here, NP-hardness results have been proved without assuming the d-to-1 Conjecture [54, 104] with much more sophisticated constructions.

4.1 The LABEL-COVER–LONG-CODE Framework

Most of the recent inapproximability results are based on LABEL-COVER and LONG-CODE. In this section, we describe the main idea of such reductions.

The starting point of such a reduction is usually some NP-hardness results for some variant of LABEL-COVER, such as Theorem 3.10, or Unique Games or d-to-1 Games as described in Section 3.3.1.

Let $\mathcal{L} = (U, V, E, L, R, \Pi, \Gamma)$ be a LABEL-COVER instance. Consider a Boolean predicate P of arity k. For each vertex $u \in U$ and $v \in V$, we expect functions $f_u : \{-1,1\}^L \to \{-1,1\}$ and $g_v : \{-1,1\}^R \to \{-1,1\}$. These are supposed to be

the LONG-CODE encodings of the labelings for u and v. Recall from Section 2.5 that f_u and g_v are LONG-CODE encodings for label $l \in L$, $r \in R$ if for each $x \in \{-1,1\}^L$, $f_u(x) = x_l$, and for each $y \in \{-1,1\}^R$, $g_v(y) = y_r$. The variables in the GAP-P instance we produce are exactly the entries of these functions. Therefore, for each $u \in U$, we have $2^{|L|}$ Boolean variables, and for each $v \in V$, we have $2^{|R|}$ Boolean variables.

We require that the functions f_u and g_v be folded over constant, that is, for any $x \in \{-1,1\}^L$, $y \in \{-1,1\}^R$,

$$f_u(-x) = -f_u(x) \qquad \text{and} \qquad g_v(-y) = -g_v(y)\,.$$

As described in Section 2.5, we can enforce this by choosing some $l_0 \in L$ and $r_0 \in R$, and return $x_{l_0} \cdot f_u(x')$ for $f_u(x)$ and $y_{r_0} \cdot g_v(y')$ for $g_v(y)$, where x' is identical to x except in coordinate l_0, where $x'_{l_0} = 1$, and y' is defined similarly. The signs of x_{l_0} and y_{r_0} correspond to the signs of the Boolean variables $f_u(x')$ and $g_v(y')$ in the output CSP. Hence in the actual reduction we only use $2^{|L|-1}$ Boolean variables for each $u \in U$ and $2^{|R|-1}$ variables for each $v \in V$. We can verify that functions that are actual LONG-CODE encodings are all folded over constant.

In a correct proof for a satisfiable LABEL-COVER instance, the functions are LONG-CODE for the corresponding labelings of u and v

$$f_u(x) = x_{\{\sigma(u)\}} \quad \text{and} \quad g_v(y) = y_{\{\sigma(v)\}}\,.$$

We now describe the constraints. The goal is to verify that the functions f_u and g_v are close to LONG-CODE encoding of labelings with high value.

For an edge $\{u,v\}$ in the LABEL-COVER, we sample *queries*

$$\left(x^{(1)}, \ldots, x^{(m)}, y^{(m+1)}, \ldots, y^{(w)}\right)$$

according to some carefully chosen *test distribution* $\mathcal{T}$. This corresponds to adding to the CSP instance a constraint

$$P\left(x^{(1)}, \ldots, x^{(m)}, y^{(m+1)}, \ldots, y^{(w)}\right)$$

with weight equal to the probability of the query.

Remark. *It is of course not necessary for all x queries to come before the y queries, and there are many other flexibilities with the distribution, depending on the particular use cases. The above is just an example to illustrate the main ideas.*

The distribution $\mathcal{T}$ has the property that for any $l \in L$ and $r \in R$ such that $\pi_{(u,v)}(r) = l$, the predicate P accepts

$$\left(x_l^{(1)}, \ldots, x_l^{(m)}, y_r^{(m+1)}, \ldots, y_r^{(w)}\right)$$

with probability 1 (or $1 - \varepsilon$ for some small constant ε if we are considering non-perfect completeness). This guarantees that if f_u and g_v are LONG-CODE encodings

of a consistent labeling of u and v, then all (or a $(1-\varepsilon)$ fraction) of the P constraints related to edge $\{u, v\}$ is satisfied.

Let the value of an edge be the following expectation

$$\mathop{\mathbf{E}}_{(x^{(1)},\dots,x^{(m)},y^{(m+1)},\dots,y^{(w)})\sim\mathcal{T}} \left[P(f_u(x^{(1)}),\dots,f_u(x^{(m)}),g_v(y^{(m+1)}),\dots,g_v(y^{(w)}))\right] . \tag{4.1}$$

Observe that from the above discussion, in the completeness case where the Label-Cover instance has an assignment that satisfies all the edges, setting f_u and g_v to the long code of the labelings would give value 1 (or close to 1) for the above expectation.

In the soundness case, of course the functions f_u and g_v are not guaranteed to be Long-Code. Typically, when proving approximation resistance, we start the analysis by taking the Fourier expansion of predicate P in Equation (4.1). The constant term in the expansion is exactly the density of P. We then show that if for some non-constant terms we have $|\mathbf{E}[\prod f_u \prod g_v]| \geq \delta$ for some small constant $\delta > 0$, then we can find a *good* labeling for the Label-Cover instance we started with, allowing us to distinguish between the completeness case and the soundness case. In some cases we can show that all possible non-constant terms—including those that do not appear in the expansion of P—are small, and this implies that predicate P is useless in the sense of [9], a stronger notion of inapproximability. If a predicate P is useless, then for any other predicate $Q \supseteq P$, the problem Gap-Q is also hard.

It is not hard to adapt the above reduction to Multi-Layered-Label-Cover. Instead of encoding the labelings of single vertices as long codes, we encode labelings for the hybrid vertex tuples. The rest of the analysis is similar.

4.2 The Predicate and the Test Distribution

The goal of this chapter is to prove that for any $k \geq 4$ all $P \not\supseteq$ Ek−Lin are useless. It suffices to establish this for P with $|P| = |\text{E}k\text{-Lin}| + 1$, and by combining negations of inputs, we further assume without loss of generality that $P := \text{E}k\text{-Lin} \cup \{1^k\}$.

We now describe the test distribution. The starting point of our reduction is a d-to-1 Label-Cover. Pick a random edge $e = \{u, v\}$. We view the function f_u as $f_u : \prod_{l\in L} \mathcal{X}^l \to \{-1, 1\}$, where each $\mathcal{X}^l = \{-1, 1\}^{(l)}$ and the function g_v as $g_v : \prod_{l\in L} \mathcal{Y}^l_j \to \{-1, 1\}$, where $\mathcal{Y}^l_j = \{-1, 1\}^{\pi_e^{-1}(l)}$ for each $j = 2, \dots, k$. The test distribution $\mathcal{T}_e$ is a distribution over the following product space

$$\prod_{l\in L} \left(\mathcal{X}^l \times \prod_{j=2}^{k} \mathcal{Y}^l_j \right) \simeq \left(\prod_{l\in L} \mathcal{X}^l \right) \times \prod_{j=2}^{k} \left(\prod_{l\in L} \mathcal{Y}^l_j \right) .$$

In fact, for each $l \in L$, we define a distribution $\mathcal{T}^l_e$ on $\mathcal{X}^l \times \prod_{j=2}^{k} \mathcal{Y}^l_j$, and the final distribution $\mathcal{T}_e = \bigotimes_{l\in L} \mathcal{T}^l_e$. The individual distribution $\mathcal{T}^l_e$ only depends

on d, and we can simply view it as a distribution on $\{-1,1\} \times \prod_{j=2}^{k} \{-1,1\}^d$. Furthermore, the distributions for different $l \in L$ are actually constructed in a uniform way. We define a family of distributions $\{-1,1\} \times \prod_{j=2}^{k} \{-1,1\}^d$, which we now write as $\mathcal{X} \times \prod_{j=2}^{k} \mathcal{Y}_j$ for simplicity.

The test distribution is a combination of several simple distributions. The first such distribution is the Ek-LIN distribution which we denote by $\mathcal{H}$ in its simple form and $\mathcal{H}(d)$ in its product form.

Definition 4.1. *Define distribution $\mathcal{H}$ on $(x, y_2, \ldots, y_k)$ by sampling as follows: pick $x, y_2, \ldots, y_{k-1}$ independently and uniformly at random from $\{-1,1\}$, then set $y_k = -x \cdot \prod_{j=2}^{k-1} y_j$.*

Define distribution $\mathcal{H}(d)$ on $\mathcal{X} \times \prod_{j=2}^{k} \mathcal{Y}_j$ as follows: pick x and $y_{r,j}$ independently and uniformly at random for $r \in [d]$ and $j = 2, \ldots, k-1$, then for each $r \in [d]$, set $y_{r,k} = -x \cdot \prod_{j=2}^{k-1} y_{r,j}$.

Next, we define the "noise" distribution. Denote $\alpha := 1^k \notin$ Ek-LIN. Observe that if we define α' to be the same as α except for the first bit $\alpha'_1 = -1$, then we have $\alpha' \in$ Ek-LIN. To generate the noise distribution, we first generate the parity distribution with some bias on the first bit. The difference now is that in the noise distribution, whenever we get $\alpha' \in$ Ek-LIN, we switch to α. We assign probabilities so that the marginal on the first bit is uniform.

Definition 4.2. *Define $\mathcal{N}$ on $\{-1,1\}^k$ as follows: generate $y_2, \ldots, y_{k-1}$ independently and uniformly at random, and with probability $2^{k-3}/(2^{k-2}-1)$, set $x = -1$, and $x = 1$ otherwise. Then let $y_k = -x \prod_{j=2}^{k-1} y_j$, and flip x if $x = -1$ and $y_2 = \ldots = y_k = 1$.*

For $r \in [d]$, define $\mathcal{N}_r(d)$ on $\mathcal{X} \times \prod \mathcal{Y}_j$: generate $(x, y_{r,2}, \ldots, y_{r,k}) \sim \mathcal{N}$, then generate $(y_{i,2}, \ldots, y_{i,k})$ for all $i \in ([d] - \{r\})$ according to $\mathcal{H}$ and x.

Define $\mathcal{N}(d) = (\sum_{r \in [d]} \mathcal{N}_r(d))/d$.

We sample $\mathcal{N}(d)$ by first choosing a random $r \in [d]$ for which we generate the noise distribution according to $\mathcal{N}$, and then fill in the rest according to $\mathcal{H}$ conditioned on the given x bit. We can view this as a generalization of the distribution $\mathcal{N}$ used in [90].

Our distribution $\mathcal{N}$ has the nice property that the marginals are uniform if we condition on y_S where $S \subsetneq \{2, \ldots, k\}$.

Lemma 4.3. *Let $S \subsetneq \{2, \ldots, k\}$ be a set of coordinates, and let y_S be an assignment to the bits in S. Then $\Pr_{\mathbf{y} \sim \mathcal{N}}[\mathbf{y}_S = y_S] = 2^{-|S|}$. Moreover, the marginal distribution over $\mathcal{X}$ is also uniform.*

Proof. If $k \notin S$, then since $\{y_i\}_{i \in S}$ are picked uniformly at random, the statement holds.

For $S \ni k$, we can change the role between y_k and y_j for some $j \notin S$. This does not affect the probability because the flip in the last step only affects x, and we are not conditioning on x.

For the marginal of $\mathcal{X}$, note that in the first step, $x = -1$ with probability $2^{k-3}/(2^{k-2}-1)$, and this will remain the case at the end unless $(x, y_2, \dots, y_{k-1}) = (-1, 1, \dots, 1)$. Thus the probability that $x = -1$ is $(1-2^{-(k-2)}) \cdot 2^{k-3}/(2^{k-2}-1) = 1/2$. □

We are now ready to define the test distribution $\mathcal{T}_e$.

Definition 4.4. *For $0 < \gamma < 1$, define distribution $\mathcal{H}_\gamma(d) = (1-\gamma)\mathcal{H}(d) + \gamma\mathcal{N}(d) = (1-\gamma)\mathcal{H}(d) + \gamma(\sum_{s\in[d]} \mathcal{N}_s(d))/d$. For each $l \in L$, define $\mathcal{T}_e^l$ to be $\mathcal{H}_\gamma(d)$. The test distribution $\mathcal{T}_e := \bigotimes_{l\in L} \mathcal{T}_e^l$.*

We have the following correlation bound for $\rho(\mathcal{X}, \prod_{i\in S} \mathcal{Y}_i; \mathcal{H}_\gamma(d))$ for $S \subsetneq \{2, \dots, k\}$. The proof is similar to that of Lemma 5.2 of [90].

Lemma 4.5. *For all $S \subsetneq \{2, \dots, k\}$, we have $\rho(\mathcal{X}, \prod_{i\in S} \mathcal{Y}_i; \mathcal{H}_\gamma(d)) \le \gamma$.*

Proof. Suppose $f : \mathcal{X} \to \mathbb{R}$ and $g : \prod_{i\in S} \mathcal{Y}_i \to \mathbb{R}$ are any functions with $\mathbf{E}[f] = \mathbf{E}[g] = 0$, $\mathbf{E}[f^2] \le 1$ and $\mathbf{E}[g^2] \le 1$ under the uniform distribution. Then

$$\mathop{\mathbf{E}}_{\mathcal{H}_\gamma(d)}[f(x)g(\mathbf{y})] = (1-\gamma) \mathop{\mathbf{E}}_{\mathcal{H}(d)}[f(x)g(\mathbf{y})] + \gamma \mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)g(\mathbf{y})] = \gamma \mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)g(\mathbf{y})],$$

because x and $\mathbf{y}$ are independent under $\mathcal{H}(d)$ as long as $S \subsetneq \{2, \dots, k\}$. By Cauchy-Schwarz, we have

$$\gamma \mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)g(\mathbf{y})] \le \gamma \sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)^2]} \sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[g(\mathbf{y})^2]} \le \gamma\,.$$

By definition of correlation, we conclude that $\rho(\mathcal{X}, \prod_{i\in S} \mathcal{Y}_i; \mathcal{H}_\gamma(d)) \le \gamma$. □

When $S = \{2, \dots, k\}$, we have perfect correlation between $\mathcal{X}$ and $\prod_{i\in S} \mathcal{Y}_i$. This makes it difficult to bound product terms such as $\left|\mathbf{E}\left[f(x)\prod_{i\in S} g(y_i)\right]\right|$. For any $k_0 \in \{2, \dots, k\}$, we do have correlation strictly smaller than 1 between $\mathcal{X} \times \prod_{i\ne k_0} \mathcal{Y}_i$ and $\mathcal{Y}_{k_0}$.

Lemma 4.6. *Let*

$$\beta = \frac{\gamma \cdot (2^{k-3}-1)}{(2^{k-2}-1) \cdot 2^{(k-2)d} \cdot d}$$

be a lower-bound of the least probability atom in $\mathrm{supp}(\mathcal{H}_\gamma(d))$. For any $k_0 \in \{2, \dots, k\}$, we have $\rho(\mathcal{X} \times \prod_{i\ne k_0} \mathcal{Y}_i, \mathcal{Y}_{k_0}, \mathcal{H}_\gamma(d)) \le 1 - \beta^2/2$.

Proof. For notational simplicity, we only prove this for $k_0 = k$, since all coordinates are entirely symmetric. We bound the correlation by Lemma 2.7. To apply the lemma, we only need to show that the distribution is connected. Fix some $s \in [d]$, and $(w_1, \dots, w_d) \in \mathcal{Y}_k$ be an arbitrary vertex on the right side of the bipartite graph, and $(w'_1, \dots, w'_d)$ be the right vertex that has $w'_j = w_j$ for $j \neq s$ and $w'_s = 1$. We claim that for any $s \in [d]$, $(w_1, \dots, w_d)$ and $(w'_1, \dots, w'_d)$ are connected. In fact, if we already have $w_s = 1$, then these two are the same vertex and we are done. Otherwise, we pick a left vertex that is connected to both of them as follows: denote the vertex by $(x, y_{2,1}, y_{2,2}, \dots, y_{2,d}, y_{3,1}, \dots, y_{k-1,d})$. Set $x = 1$. For $j \neq s$, we let

$$(y_{2,j}, y_{3,j}, \dots, y_{k-2,j}, y_{k-1,j}) = (1, 1, 1, \dots, 1, -w_j),$$

and for $j = s$, let

$$(y_{2,s}, y_{3,s}, \dots, y_{k-2,s}, y_{k-1,s}) = (1, 1, \dots, 1).$$

For any $j \neq s$, we have that $(x, y_{2,j}, \dots, y_{k-1,j}, w_j) \in \mathcal{H}$, and for $j = s$, we have that $(x, y_{2,s}, \dots, y_{k-1,s}, w_s) \in \mathcal{H}$ and $(x, y_{2,s}, \dots, y_{k-1,s}, w'_s) \in \mathcal{N}(d)$. Therefore it is connected to $(w_1, \dots, w_d)$ due to $\mathcal{H}(d)$ and to $(w'_1, \dots, w'_d)$ due to $\mathcal{N}_s(d)$.

We conclude that all right vertices are connected to the right vertex $(1, 1, \dots, 1)$. It is also easy to see that all left vertices are at least connected with one right vertex, therefore the whole graph is connected. □

Similar to [90], we need to pass from $\mathcal{H}_\gamma(d)$ to a new distribution $\mathcal{I}_\gamma(d)$ with almost no correlation between $\mathcal{X}$ and $\prod \mathcal{Y}_i$.

Definition 4.7. *Define distribution $\mathcal{I}(d)$ on $\mathcal{X} \times \prod_{i=2}^k \mathcal{Y}_i$ as follows: first sample from $\mathcal{H}(d)$, then uniformly rerandomize x. Define $\mathcal{I}_\gamma(d) := (1-\gamma)\mathcal{I}(d) + \gamma\mathcal{N}(d)$.*

We bound $\rho(\mathcal{X}, \prod \mathcal{Y}_i; \mathcal{I}_\gamma(d))$ in the following lemma.

Lemma 4.8. $\rho(\mathcal{X}, \prod_{i=2}^k \mathcal{Y}_i; \mathcal{I}_\gamma(d)) \leq \sqrt{\gamma}$.

The proof of this lemma is almost identical to Lemma 5.4 of [90].

Proof. Let $f : \mathcal{X} \to \mathbb{R}$ be any function with $\mathbf{E}[f] = 0$ and $\mathbf{E}[f^2] \leq 1$ under the uniform distribution (which is the marginal of $\mathcal{X}$ under $\mathcal{I}(d)$ and $\mathcal{N}(d)$, and hence $\mathcal{I}_\gamma(d)$). Let $G : \prod_{i=2}^k \mathcal{Y}_i \to \mathbb{R}$ be any function with $\mathbf{E}_{\mathcal{I}_\gamma(d)}[G] = 0$ and $\mathbf{E}_{\mathcal{I}_\gamma(d)}[G^2] \leq 1$. Decomposing $\mathcal{I}_\gamma(d)$, we get

$$1 \geq \mathop{\mathbf{E}}_{\mathcal{I}_\gamma(d)}[G^2] = (1-\gamma) \mathop{\mathbf{E}}_{\mathcal{I}(d)}[G^2] + \gamma \mathop{\mathbf{E}}_{\mathcal{N}(d)}[G^2] \geq \gamma \mathop{\mathbf{E}}_{\mathcal{N}(d)}[G^2].$$

This implies

$$\mathop{\mathbf{E}}_{\mathcal{N}(d)}[G^2] \leq 1/\gamma.$$

Observe that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathcal{J}_\gamma(d)}[f(x)G(y_2,\dots,y_k)] \\
&= (1-\gamma)\mathop{\mathbf{E}}_{\mathcal{J}(d)}[f(x)G(y_2,\dots,y_k)] + \gamma\mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)G(y_2,\dots,y_k)] \\
&= (1-\gamma)\mathop{\mathbf{E}}_{\mathcal{J}(d)}[f(x)]\mathop{\mathbf{E}}_{\mathcal{J}(d)}[G(y_2,\dots,y_k)] + \gamma\mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)G(y_2,\dots,y_k)] \\
&\le (1-\gamma)\mathop{\mathbf{E}}_{\mathcal{J}(d)}[f(x)]\mathop{\mathbf{E}}_{\mathcal{J}(d)}[G(y_2,\dots,y_k)] + \gamma\sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)^2]}\sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[G(y_2,\dots,y_k)^2]}\,.
\end{aligned}$$

We know that $\mathbf{E}_{\mathcal{J}(d)}[f]=0$, and that

$$\gamma\sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[f(x)^2]}\sqrt{\mathop{\mathbf{E}}_{\mathcal{N}(d)}[G(y_2,\dots,y_k)^2]} \le \gamma\cdot 1\cdot\sqrt{1/\gamma} = \sqrt{\gamma}\,.$$

Hence

$$\mathop{\mathbf{E}}_{\mathcal{J}_\gamma(d)}[f(x)G(y_2,\dots,y_k)] \le \sqrt{\gamma}\,.$$

This completes the proof. □

Combining the above lemmas with Lemma 2.6, we have the following upper-bounds for correlation on product spaces.

Lemma 4.9. *Let*

$$\beta = \frac{\gamma\cdot(2^{k-3}-1)}{(2^{k-2}-1)\cdot 2^{(k-2)d}\cdot d}\,.$$

For any $S\subsetneq\{2,\dots,k\}$, $k_0\in\{2,\dots,k\}$, we have

$$\begin{aligned}
\rho\left(\prod_{l\in L}\mathcal{X}^l,\prod_{l\in L}\left(\prod_{i\in S}y_i^l\right);\bigotimes_{l\in L}\mathcal{H}_\gamma(d)\right) &\le \gamma;\\
\rho\left(\prod_{l\in L}\left(\mathcal{X}^l\times\prod_{i\neq k_0}y_i^l\right),\prod_{l\in L}y_{k_0}^l;\bigotimes_{l\in L}\mathcal{H}_\gamma(d)\right) &\le 1-\beta^2/2;\\
\rho\left(\prod_{l\in L}\mathcal{X}^l,\prod_{l\in L}\left(\prod_{i=2}^{k}y_i^l\right);\bigotimes_{l\in L}\mathcal{J}_\gamma(d)\right) &\le \sqrt{\gamma}\,.
\end{aligned}$$

4.3 Analysis of the Reduction

We now analyze the completeness and soundness of the reduction.

The completeness analysis is standard.

Theorem 4.10. *The reduction has completeness* 1.

Proof. Let $\mathcal{L}$ be a satisfiable d-to-1 Label-Cover instance, and σ be a perfect labeling. To construct a satisfying assignment for the CSP instance, for each $u \in U$, we let f_u be the Long-Code for $\sigma(u)$, and for each $v \in V$, let g_v be the Long-Code for $\sigma(v)$. For any edge e, we have that $\pi_e(\sigma(v)) = \sigma(u)$. It follows from the definition of the test distribution that the tuple

$$\left(x_{\sigma(u)}, y_{2,\sigma(v)}, \dots, y_{k,\sigma(v)}\right)$$

is in the support of P. Thus the functions serve as an assignment that satisfies all constraints. □

The following statement gives the soundness of the construction.

Theorem 4.11. *Let $\mathcal{L}$ be a d-to-1* Label-Cover *instance, and let Ψ be the* Max-P *instance produced by the above reduction. If there is an assignment to Ψ with value at least $|P|/2^k + \varepsilon$ then there is a randomized labeling strategy for $\mathcal{L}$ achieving expected value at least η, for some positive constant η depending only on d and ε.*

Proof. We first arithmetize the probability predicate P accepts a random query

$$\begin{aligned}&\Pr_{e=\{u,v\}\sim E} \Pr_{\mathcal{T}_e}\left[P\left(f_u(x), g_v(y_2), \dots, g_v(y_k)\right)\right] \\ &= \mathop{\mathbf{E}}_{e,\mathcal{T}_e}\left[\sum_{S\in\mathbb{F}_2^k} \widehat{P}_S \chi_S\left(f_u(x), g_v(y_2), \dots, g_v(y_k)\right)\right].\end{aligned}$$

We have that $\widehat{P}_{\mathbf{0}} = |P|/2^k$.

By Lemma 4.3 and the fact that f_u and g_v are folded over constant, we conclude that if $S \subsetneq \{2, \dots, k\}$ or S only contains the coordinate corresponding to x, then

$$\mathop{\mathbf{E}}_{e,\mathcal{T}_e}\left[\chi_S(f_u(x), g_v(y_2), \dots, g_v(y_k))\right] = 0\,.$$

Also, by Lemma 4.9, the absolute value of terms with S containing coordinate x but not all of $\{2, \dots, k\}$ is at most γ. Therefore the acceptance probability can be simplified as

$$\begin{aligned}&\Pr_{e=\{u,v\}\sim E} \Pr_{\mathcal{T}_e}\left[P\left(f_u(x), g_v(y_2), \dots, g_v(y_k)\right)\right] \\ &\le \frac{|P|}{2^k} + \frac{2^{k-2}-1}{2^{k-1}} \cdot \gamma \\ &\quad + \mathbf{E}\left[\widehat{P}_{\{2,\dots,k\}} \prod_{i=2}^{k} g_v(y_i)\right] + \mathbf{E}\left[\widehat{P}_{\{x\}\cup\{2,\dots,k\}} f_u(x) \prod_{i=2}^{k} g_v(y_i)\right].\end{aligned}$$

We bound the remaining two expectations in the following theorems.

Theorem 4.12. *For any $e = \{u, v\}$ and function $g_v : \{-1,1\}^R \to \{-1,1\}$ that is folded over constant, we have $\mathbf{E}_{\mathcal{T}_e}\left[\prod_{i=2}^{k} g_v(y_i)\right] \le \gamma$.*

The proof of the above theorem is given in Section 4.4.

Theorem 4.13. *There exists constant $\delta', \gamma > 0$ depending only on d, k and γ, such that the following holds: if for every $l \in L$ and every odd-cardinality set $S \subseteq \pi_e^{-1}(l)$, we have that*

$$\min\left\{\mathrm{Inf}_l(T_{1-\delta'/2}f_u), \mathrm{Inf}_S(T_{1-\delta'/2}g_v)\right\} \leq \tau\,,$$

then

$$\left|\mathop{\mathbf{E}}_{e,\mathcal{T}_e}\left[f_u(x)\prod_{i=2}^{k} g_v(y_i)\right]\right| \leq (k+2)\sqrt{\gamma}\,.$$

The proof of the theorem is almost the same as the corresponding one in [90] and we include it in Section 4.5.

Now we combine Theorem 4.12 and Theorem 4.13 to complete the soundness analysis. Under the hypothesis of Theorem 4.13, the acceptance probability can be upper-bounded by

$$\frac{|P|}{2^k} + \frac{2^{k-2}-1}{2^{k-1}}\cdot\gamma + \gamma + (k+2)\sqrt{\gamma} \leq \frac{|P|}{2^k} + 2\gamma + (k+2)\sqrt{\gamma}\,.$$

Equivalently, suppose some sets of functions $\{f_u\}_{u\in U}$ and $\{g_v\}_{v\in V}$ cause the acceptance probability to exceed $|P|/2^k + \varepsilon = |P|/2^k + 2\gamma + 2(k+2)\sqrt{\gamma}$, then

$$\left|\mathop{\mathbf{E}}_{e,\mathcal{T}_e}\left[f_u(x)\prod_{i=2}^{k} g_v(y_i)\right]\right| > 2(k+2)\sqrt{\gamma}\,.$$

By an averaging argument, this implies that for at least a $(k+2)\sqrt{\gamma}$ fraction of the edges, we have

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}_e}\left[f_u(x)\prod_{i=2}^{k} g_v(y_i)\right]\right| > (k+2)\sqrt{\gamma}\,.$$

We call such edges "good".

By Theorem 4.13, we know that for each good edge $e = \{u, v\}$, there exists $l_e \in L$ and an odd cardinality set $S_e \subseteq \pi_e^{-1}(l_e)$, such that

$$\min\left\{\mathrm{Inf}_l(T_{1-\delta'/2}f_u), \mathrm{Inf}_S(T_{1-\delta'/2}g_v)\right\} \geq \tau\,.$$

For each $u \in U$, define $L_u := \left\{l \in L \mid \mathrm{Inf}_i(T_{1-\delta'/2}f_u) > \tau\right\}$, and for each $v \in V$, define $L_v := \left\{r \in R \mid \exists S, |S| \leq d, |S| \text{ is odd, s.t. } \mathrm{Inf}_S(T_{1-\delta'/2}g_v) > \tau, r \in S\right\}$. By Proposition 2.31, we have that $|L_u| \leq 1/\delta'\tau$.

As for $|L_v|$, note that $\sum_{|S|\leq d}\mathrm{Inf}_S(T_{1-\delta'/2}g_v) \leq (d/\delta')^d$, therefore at most $(d/\delta')^d/\tau$ sets S can contribute in the definition of L_v, and each S contribute at most d elements to L_v, thus $|L_v| \leq d\cdot(d/\delta')^d/\tau$.

For a good edge e, we have that $l_e \in L_u$ and S_e contributes to L_v. Since S_e is odd, it is nonempty, and therefore there exists $r_e \in S \subseteq \pi_e^{-1}(l_e)$. If we pick

randomly a label from L_u and a label from L_v, then the probability that we pick a pair of matching labels is at least $1/(|L_u||L_v|) \geq \tau^2(\delta'/d)^{d+1}$. Recall that at least $(k+2)\sqrt{\gamma}$ fraction of the edges are good, thus the expected fraction of edges satisfied by this randomized labeling is at least $(k+2)\sqrt{\gamma}\tau^2(\delta'/d)^{d+1}$, a positive constant depending only on d, k and ε, as desired. This completes the analysis of the soundness of the reduction. □

4.4 Analyzing $\mathbf{E}[\prod_{i=2}^{k} g_v(y_i)]$

In this section, we prove Theorem 4.12. We follow essentially the same approach as in [90]. The main difference is that the approach in [90] is defined for product of two functions, whereas in our case we have $k-1$ of them. To apply their approach, we study the conditional distribution where $y_2, \ldots, y_{k-2}$ is given. Note that x is not being conditioned on. Let $\mathcal{H}(d, \{z_i\}_{i=2}^{k-2})$ be the distribution $\mathcal{H}(d)$ conditioned on $y_i = z_i$ for $i = 2, \ldots, k-2$. Similarly, we define $\mathcal{N}(d, \{z_i\})$, $\mathcal{N}_s(d, \{z_i\})$ and $\mathcal{H}_\gamma(d, \{z_i\})$.

Let $(\{-1,1\}^d \times \{-1,1\}^d, \mu)$ be correlated probability spaces, and let $M(\mu)$ be the $2^d \times 2^d$ matrix associated with μ, defined as

$$M(\mu)_{S,T} = \mathop{\mathbf{E}}_{(x,y)\sim\mu}[\chi_S(x)\chi_T(y)].$$

For a function $g : \{-1,1\}^d \to \mathbb{R}$, we can identify it with a column vector of dimension 2^d, with entries indexed by $S \subseteq [d]$ in the same order as in the matrix notation for $M(\mu)$. The S-th entry of g is $\hat{g}_S$. Then we have

$$\mathop{\mathbf{E}}_{\mu}[g(x)g(y)] = g^T M(\mu) g.$$

Thus, to bound $\mathbf{E}_{\mathcal{H}_\gamma(d)}[\prod_{i=2}^{k} g(y_i)]$, all we need is to upper-bound the absolute value of

$$g^T \left(\bigotimes_{l \in L} M(\mathcal{H}_\gamma(d, \{y_i\}_{i=2}^{k-2}) \right) g.$$

For notational simplicity, we use $\vec{y}$ to denote $\{y_i\}_{i=2}^{k-2}$.

The following statement is straightforward to verify.

Proposition 4.14. *For any $\{y_i\}_{i=2}^{k}$, the entry $M(\mathcal{H}(d,\vec{y}))_{S,T}$ is nonzero iff $S = T$ and that $|S| = |T|$ are even, in which case the entry is ± 1.*

Define the distribution $\mathcal{E}(d)$ on $\mathcal{X} \times \prod_{i=2}^{k} \mathcal{Y}_i$ which generates pairs (y_{k-1}, y_k) by choosing y_{k-1} uniformly at random and setting $y_k = y_{k-1}$ regardless of the values of x and other y_i. It is easy to see that $M(\mathcal{E}(d,\vec{y}))$ is the identity matrix. Further, denote by $\mathcal{E}_\gamma(d) = (1-\gamma)\mathcal{H}(d) + \gamma\mathcal{E}(d)$.

The remaining of the proof is divided into two steps. First, we show that the absolute value of the expectation under $\mathcal{H}_\gamma(d,\vec{y})$ is upper-bounded by the expectation under $\mathcal{E}_\gamma(d,\vec{y})$. We then derive an upper-bound for the latter.

For the first step, we use the following lemma in matrix algebra.

Lemma 4.15 ([90])**.** *Let A_i and B_i be $m \times m$ matrices, $i = 1, \ldots, n$, and suppose that $A_i - B_i$ and $A_i + B_i$ are both positive semidefinite. Then $\bigotimes_{i=1}^n A_i - \bigotimes_{i=1}^n B_i$ and $\bigotimes_{i=1}^n A_i + \bigotimes_{i=1}^n B_i$ are both positive semidefinite.*

We have the following for the matrices associated with $\mathcal{H}_\gamma(d,\vec{y})$ and $\mathcal{E}_\gamma(d,\vec{y})$.

Lemma 4.16. *For any fixed value of $\vec{y}$, the matrices*

$$\bigotimes_{l\in L} M\left(\mathcal{E}_\gamma(d,\vec{y})\right) \pm \bigotimes_{l\in L} M\left(\mathcal{H}_\gamma(d,\vec{y})\right)$$

are positive semidefinite.

Proof. By Lemma 4.15, it suffices to show that the matrices

$$M\left(\mathcal{E}_\gamma(d,\vec{y})\right) \pm M\left(\mathcal{H}_\gamma(d,\vec{y})\right)$$

are positive semidefinite. For notational simplicity, we omit the dependence on d and $\vec{y}$ in the rest of the proof.

For the conditional distributions, we still have $\mathcal{H}_\gamma = (1-\gamma)\mathcal{H} + \gamma\mathcal{N}$ and $\mathcal{E}_\gamma = (1-\gamma)\mathcal{H} + \gamma\mathcal{E}$. Therefore $M(\mathcal{E}_\gamma) - M(\mathcal{H}_\gamma) = \gamma(M(\mathcal{E}) - M(\mathcal{N}))$. To show that $M(\mathcal{E}_\gamma) - M(\mathcal{H}_\gamma)$ is positive semidefinite, it suffices to show it for $M(\mathcal{E}) - M(\mathcal{N})$. For any $h : \{-1,1\}^d \to \mathbb{R}$, using Cauchy-Schwarz, we have

$$\begin{aligned} h^T M(\mathcal{N}) h \;&=\; \mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{N}}[h(y_{k-1})h(y_k)] \\ &\le \sqrt{\mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{N}}[h(y_{k-1})^2]}\sqrt{\mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{N}}[h(y_k)^2]}\,. \end{aligned}$$

The conditional marginals under $\mathcal{N}$ of y_{k-1} and y_k are uniform by Lemma 4.3. By the way we define $\mathcal{E}$, the same conditional marginals are also uniform under $\mathcal{E}$. Therefore we can continue the calculation above and get

$$\begin{aligned} &\sqrt{\mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{N}}[h(y_{k-1})^2]}\sqrt{\mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{N}}[h(y_k)^2]} \\ =\; &\mathbf{E}[h^2] = \mathop{\mathbf{E}}_{y_{k-1},y_k\sim\mathcal{E}}[h(y_{k-1})h(y_k)] = h^T M(\mathcal{E}) h\,, \end{aligned}$$

therefore $h^T M(\mathcal{N}) h \le h^T M(\mathcal{E}) h$ for all h, and hence $M(\mathcal{E}) - M(\mathcal{N})$ is positive semidefinite.

As for $M(\mathcal{E}_\gamma) + M(\mathcal{H}_\gamma)$, it equals $2(1-\gamma)M(\mathcal{H}_\gamma) + \gamma(M(\mathcal{E}) + M(\mathcal{N}))$. The matrix $M(\mathcal{H})$ is diagonal with only nonnegative numbers on the diagonal and

therefore is positive semidefinite. As for $M(\mathcal{E}) + M(\mathcal{N})$, the proof is essentially the same as $M(\mathcal{E}) - M(\mathcal{N})$: we start with $h^T(-M(\mathcal{N}))h$ and the minus sign disappears with the application of Cauchy-Schwarz. □

We use the following lemma to bound the expectation under $\mathcal{E}_\gamma$.

Lemma 4.17. *Let $g : \{-1,1\}^R \to \{-1,1\}$ be a function folded over constant. Then for any $\vec{y}$, we have*

$$\left| g^T \left(\bigotimes_{l \in L} M(\mathcal{E}_\gamma(d, \vec{y})) \right) g \right| \leq \gamma \,.$$

Proof. Recall that $M(\mathcal{H}(d,\vec{y}))$ is a diagonal matrix with (S,S) equals 0 if $|S|$ is odd and ± 1 if $|S|$ is even, and $M(\mathcal{E}(d,\vec{y}))$ is the identity matrix. Therefore $M(\mathcal{E}_\gamma(d,\vec{y}))$ is a diagonal matrix whose (S,S) entry has absolute value at most 1 if $|S|$ is even and γ if $|S|$ is odd. It follows that

$$\left| g^T \left(\bigotimes_{l \in L} M(\mathcal{E}_\gamma(d, \vec{y})) \right) g \right| \leq \sum_{S \subseteq R} \hat{g}_S^2 \cdot \gamma^{\#\{l \in L \mid |S \cap \pi^{-1}(l)| \text{ is odd.}\}} \,.$$

Since g is folded over constant, $\hat{g}_S^2$ is nonzero only if $|S|$ is odd. Note that if $|S|$ is odd, then for at least one l, we have that $|S \cap \pi^{-1}(l)|$ is odd, therefore we can bound the above by

$$\sum_{S \subseteq [R]} \hat{g}_S^2 \cdot \gamma \leq \mathbf{E}[g^2] \cdot \gamma = \gamma \,.$$

□

We can now conclude that for any fixed $\vec{y}$, the expectation of $g(y_{k-1})g(y_k)$ is small.

Theorem 4.18. *For any $e = \{u, v\}$, $g_v : \{-1,1\}^R \to \{-1,1\}$ that is folded over constant, and any $\vec{y} = \{y_i\}_{i=2}^{k-2}$, we have*

$$\left| \mathop{\mathbf{E}}_{(y_{k-1}, y_k) \sim \mathcal{T}_e(\mathbf{y})} [g(y_{k-1})g(y_k)] \right| \leq \gamma \,.$$

Proof. Using the matrix notation, we have

$$\begin{aligned} \mathop{\mathbf{E}}_{\mathcal{T}_e(\vec{y})} [g(y_{k-1})g(y_k)] &= g^T M(\mathcal{T}_e(\vec{y}))g \\ = g^T \left(\bigotimes_{l \in L} M(\mathcal{H}_\gamma(d, \vec{y})) \right) g \,. \end{aligned}$$

We bound the above with Lemma 4.16.

$$g^T \left(\bigotimes_{l\in L} M(\mathcal{E}_\gamma(d,\vec{y})) - \bigotimes_{l\in L} M(\mathcal{H}_\gamma(d,\vec{y})) \right) g \ \geq 0,$$
$$g^T \left(\bigotimes_{l\in L} M(\mathcal{E}_\gamma(d,\vec{y})) + \bigotimes_{l\in L} M(\mathcal{H}_\gamma(d,\vec{y})) \right) g \ \geq 0\,.$$

This implies that

$$\begin{aligned} &- g^T \left(\bigotimes_{l\in L} M(\mathcal{E}_\gamma(d,\vec{y})) \right) g \\ &\leq g^T \left(\bigotimes_{l\in L} M(\mathcal{H}_\gamma(d,\vec{y})) \right) g \ \leq \ g^T \left(\bigotimes_{l\in L} M(\mathcal{E}_\gamma(d,\vec{y})) \right) . \end{aligned}$$

Finally by Lemma 4.17

$$\left| \mathop{\mathbf{E}}_{(y_{k-1},y_k)\sim\mathcal{T}_e(\mathbf{y})} [g(y_{k-1})g(y_k)] \right| \leq \gamma\,.$$

□

4.5 Analyzing $\mathbf{E}[f(x)\prod_{i=2}^{k} g_v(y_i)]$

We now prove Theorem 4.13. The analysis is almost an exact copy of O'Donnell and Wu's approach. We include the full analysis here to demonstrate an Invariance-Principle style soundness analysis.

We first explains the main steps, followed by the proofs of the key theorem of each step.

Let $\mathcal{H}_\gamma := \bigotimes_{l\in L} \mathcal{H}_\gamma(d)$, and $\mathcal{I}_\gamma := \bigotimes_{l\in L} \mathcal{I}_\gamma(d)$. The goal is to first show that

$$\mathop{\mathbf{E}}_{\mathcal{H}_\gamma} \left[f_u \prod_{i=2}^{k} g_v(y_i) \right] \approx \mathop{\mathbf{E}}_{\mathcal{I}_\gamma} \left[f_u \prod_{i=2}^{k} g_v(y_i) \right] ,$$

and then bound the right hand side.

We omit subscripts u and v in the rest of the proof.

We first apply the Bonami-Beckner operator to the functions. We call this the noise introduction step.

Theorem 4.19. *There are positive constants $\delta \geq \delta' > 0$ depending only on γ and d such that*

$$\left| \mathop{\mathbf{E}}_{\mathcal{H}_\gamma} \left[f(x) \prod_{i=2}^{k} g(y_i) \right] - \mathop{\mathbf{E}}_{\mathcal{H}_\gamma} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] \right| \leq k\sqrt{\gamma}\,.$$

Next, we move from distribution $\mathcal{H}_\gamma$ to $\mathcal{I}_\gamma$ by an Invariance-Principle type argument.

Theorem 4.20. *There exists constants $\tau > 0$ depending only on d, γ and δ', such that the following holds: if for every $l \in L$ and every odd cardinality-set $S \subseteq \pi_e^{-1}(l)$, we have*

$$\min\left\{\mathrm{Inf}_l(T_{1-\delta'}f, \mathrm{Inf}_S(T_{1-\delta'}g)\right\} \leq \tau\,,$$

then we have

$$\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[T_{1-\delta'}f(x)\prod_{i=2}^{k}T_{1-\delta}g(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{I}_\gamma}\left[T_{1-\delta'}f(x)\prod_{i=2}^{k}T_{1-\delta}g(y_i)\right]\right| \leq \sqrt{\gamma}\,.$$

Finally, we have the following theorem bounding the expectation under $\mathcal{I}_\gamma$.

Theorem 4.21.

$$\left|\mathop{\mathbf{E}}_{\mathcal{I}_\gamma}\left[T_{1-\delta'}\prod_{i=2}^{k}T_{1-\delta'}g(y_i)\right]\right| \leq \sqrt{\gamma}\,.$$

We give the proofs of Theorem 4.19 and Theorem 4.20 below. Theorem 4.21 follows directly from Lemma 4.9.

Let us now start with the proof of Theorem 4.19. The idea is to apply the Bonami-Beckner operator to the functions one by one. Intuitively, the Bonami-Beckner operator does not change the low-degree parts of the functions by too much. For the high-degree parts, it follows from Proposition 2.36 that their overall contributions are small regardless of whether $T_{1-\delta'}$ is applied or not.

We first introduce noise to the g functions.

Lemma 4.22. *There exists a small $\delta > 0$ as a function of d and γ, such that for any $j \in \{2, \ldots, k\}$*

$$\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j}g(y_i)\cdot\prod_{i=j+1}^{k}T_{1-\delta}g(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j-1}g(y_i)\cdot\prod_{i=j}^{k}T_{1-\delta}g(y_i)\right]\right| \leq \sqrt{\gamma}\,.$$

Proof. We specify the parameter δ at the end of the proof.

Let $\mathcal{U}$ be the conditional expectation operator for the correlated probability space

$$\left(\left(\{-1,1\}^L \times \prod_{i=2}^{k-1}\{-1,1\}^R\right) \times \{-1,1\}^R, \mathcal{H}_\gamma\right)\,,$$

mapping function h on the latter space to the former space by

$$(\mathcal{U}h)(x, \{y_i\}_{i\in\{2,\ldots,k\}-\{j\}}) = \mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[h(y_j) \mid (x, \{y_i\}_{i\in\{2,\ldots,k\}-\{j\}})\right]\,.$$

We have

$$\begin{aligned}
&\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j} g(y_i)\cdot\prod_{i=j+1}^{k} T_{1-\delta}g(y_i)\right]-\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j-1} g(y_i)\cdot\prod_{i=j}^{k} T_{1-\delta}g(y_i)\right]\right| \\
&=\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j-1} g(y_i)\cdot\prod_{i=j+1}^{k} T_{1-\delta}g(y_i)\cdot(\mathrm{id}-T_{1-\delta})g(y_j)\right]\right| \\
&=\left|\mathop{\mathbf{E}}_{(x,\{y_i\}_{i\in\{2,\dots,k\}-\{j\}})\sim\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{j-1} g(y_i)\cdot\prod_{i=j+1}^{k} T_{1-\delta}g(y_i)\right.\right. \\
&\quad\left.\left.(\mathcal{U}(\mathrm{id}-T_{1-\delta})g(y_j))(x,\{y_i\}_{i\in\{2,\dots,k\}-\{j\}})\right]\right|. \qquad (4.2)
\end{aligned}$$

Consider the function inside the expectation as two functions, $G=\mathcal{U}(\mathrm{id}-T_{1-\delta})g$, and everything else $F=f\cdot(\prod_{i=2}^{j-1} g)\cdot(\prod_{i=j+1}^{k} T_{1-\delta}g)$. Take the Efron-Stein decomposition of F and G with respect to $\mathcal{H}_\gamma$. By orthogonality of the Efron-Stein decomposition, we have

$$\begin{aligned}
(4.2) &= \left|\sum_{S\subseteq L}\mathop{\mathbf{E}}_{(x,\{y_i\}_{i\in\{2,\dots,k\}-\{j\}})\sim\mathcal{H}_\gamma}[F_S\cdot G_S]\right| \\
&\le \sqrt{\sum_{S\subseteq L}\|F_S\|_2^2}\sqrt{\sum_{S\subseteq L}\|G_S\|_2^2}\le\sqrt{\sum_{S\subseteq L}\|G_S\|_2^2}\,, \qquad (4.3)
\end{aligned}$$

where on the first line the inputs to F_S and G_S are $(x,\{y_i\}_{i\in\{2,\dots,k\}-\{j\}})$, the $\|\cdot\|_2$ are with respect to the marginals of $\mathcal{H}_\gamma$ on $\mathcal{X}\times\prod_{i\in\{2,\dots,k\}-\{j\}}\mathcal{Y}_i$. The conditional expectation operator U commutes with taking Efron-Stein decomposition, so we have $G_S=\mathcal{U}G'_S$, where $G'=(\mathrm{id}-T_{1-\delta})g$. Note that the Efron-Stein decomposition for G' is with respect to the marginal distribution of $\mathcal{H}_\gamma$ on $\mathcal{Y}_j$, namely the uniform distribution. Applying the Bonami-Beckner operator also commutes with taking Efron-Stein decomposition, hence we have $G_S=\mathcal{U}G'_S=\mathcal{U}(\mathrm{id}-T_{1-\delta})g_S$. Substituting this into (4.3) yields

$$(4.3)=\sqrt{\sum_{S\subseteq L}\|\mathcal{U}(id-T_{1-\delta})g_S\|_2^2}\,.$$

Recall that the Efron-Stein decomposition for g satisfies

$$g_S=\sum_{U\subseteq R:\pi(U)=S}\hat{g}_U\chi_U\,,$$

where π is the projection on the edge. Let ρ_0 be the bound in Lemma 4.9. Applying

Proposition 2.36, we get

$$\begin{aligned}(4.3) &\le \sqrt{\sum_{S\subseteq L} \rho_0^{|S|} \|(\mathrm{id} - T_{1-\delta})g_S\|_2^2}\,,\\ \|(\mathrm{id} - T_{1-\delta})g_S\|_2^2 &= \sum_{U\subseteq R:\pi(U)=S} \left(1 - (1-\delta)^{2|U|}\,|\hat{g}_U|^2\right)\\ &\le \sum_{U\subseteq R:\pi(U)=S} \left(1 - (1-\delta)^{2d|S|}\,|\hat{g}_U|^2\right)\\ &= (1 - (1-\delta)^{2d|S|})\|g_S\|_2^2\,,\end{aligned}$$

therefore

$$(4.3) \le \sqrt{\sum_{S\subseteq L} \rho_0^{|S|}(1 - (1-\delta)^{2d|S|})\|g_S\|_2^2}\,.$$

We bound the coefficients by

$$\rho_0^{|S|}(1 - (1-\delta)^{2d|S|}) \le \exp(-|S|\beta^2)\cdot(2d|S|\delta)\,.$$

We can now choose $\delta > 0$ small enough so that the above is upper-bounded by γ, and thus $(4.2) \le \sqrt{\gamma}$. □

Applying Lemma 4.22 to the g functions one by one, and we have the following

$$\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{k} g(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{k} T_{1-\delta}g(y_i)\right]\right| \le (k-1)\sqrt{\gamma}\,.$$

It remains to apply the Bonami-Beckner operator to f.

Lemma 4.23. *There exists a constant $\delta' > 0$ depending on δ, d, γ, such that*

$$\left|\mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[f(x)\prod_{i=2}^{k} g(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{H}_\gamma}\left[T_{1-\delta'}f(x)\prod_{i=2}^{k} T_{1-\delta}g(y_i)\right]\right| \le \sqrt{\gamma}\,.$$

The proof is almost identical to Lemma 4.22. The only difference is that we need a noisy version of $\mathcal{H}_\gamma$, denoted as $\mathcal{H}^*_\gamma$, where we first generate according to $\mathcal{H}_\gamma$, then rerandomize each bit in y_i with probability δ. Define $\mathcal{H}_\gamma(d)$ similarly. We have the following correlation bound similar to Lemma 4.9.

Lemma 4.24. *$\rho(\{-1,1\}, \prod_{i=2}^{k}\{-1,1\}^d; \mathcal{H}^*_\gamma(d)) \le 1-\beta^2/2$, where $\beta = \gamma\cdot(2^{k-3}-1)\cdot\delta^{(k-1)d}/(2^{(2k-3)d}\cdot d\cdot(2^{k-2}-1))$ is a lower-bound of the least probability of an atom in $\mathcal{H}^*_\gamma(d)$.*

This completes the proof of Theorem 4.19.

Next we prove Theorem 4.20. Recall that $\mathcal{H}_\gamma = \bigotimes \mathcal{H}_\gamma(d)$ and $\mathcal{I}_\gamma = \bigotimes \mathcal{I}_\gamma(d)$. The overall plan is to change the distribution one by one from $\mathcal{H}_\gamma(d)$ to $\mathcal{I}_\gamma(d)$. We prove the following theorem.

Theorem 4.25. *For each $l \in L$*

$$\left| \mathop{\mathbf{E}}_{\bigotimes_{i=1}^{l-1} \mathcal{J}_\gamma(d) \otimes \bigotimes_{i=l}^{L} \mathcal{H}_\gamma(d)} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] \right.$$
$$\left. - \mathop{\mathbf{E}}_{\bigotimes_{i=1}^{l} \mathcal{J}_\gamma(d) \otimes \bigotimes_{i=l+1}^{L} \mathcal{H}_\gamma(d)} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] \right| \le \Delta_l \,, \tag{4.4}$$

where

$$\Delta_l := \tau^{\delta'/(2k)} \left(2^d \operatorname{Inf}_l(T_{1-\delta'/2} f) + \sum_{S \subseteq \pi^{-1}(l), |S| \text{ is odd}} \operatorname{Inf}_S(T_{1-\delta'/2} g) \right) .$$

We first prove Theorem 4.20 using Theorem 4.25.

Proof of Theorem 4.20. Summing over all $l \in L$, by triangle inequality and Proposition 2.31, we have

$$\begin{aligned}
&\left| \mathop{\mathbf{E}}_{\mathcal{H}_\gamma} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] - \mathop{\mathbf{E}}_{\mathcal{J}_\gamma} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] \right| \\
&\le \tau^{\delta'/(2k)} \left(2^d \sum_{l \in L} \operatorname{Inf}_l(T_{1-\delta'/2} f) + \sum_{S \subseteq \pi^{-1}(l') \text{ for some } l' \in L} \operatorname{Inf}_S(T_{1-\delta'/2} g) \right) \\
&\le \tau^{\delta'/(2k)} \left(2^d (1/\delta') + (d/\delta')^d \right) \\
&\le 2\tau^{\delta'/(2k)} (d/\delta')^d \,.
\end{aligned}$$

Choose τ small enough so that the last line is bounded by $\sqrt{\gamma}$, and this completes the proof. □

Now we prove Theorem 4.25.

Proof of Theorem 4.25. We show the theorem for the case $l = 1$. The other cases are similar.

Given x and y_i, we write $x' = (x_2, \dots, x_L)$, and $y_i' = (y_{i,d+1}, \dots, y_{i,R})$. We break up the Fourier expansion of f according to its dependence on x_1:

$$f(x) = F_\emptyset(x') + x_1 F_1(x') \,.$$

Similarly, we break up the Fourier expansion of g according to its dependence on $y_1, \dots, y_d$:

$$g(y) = \sum_{S \subseteq [d]} \chi_S(y_1, \dots, y_d) G_S(y') \,,$$

where for any $S \subseteq [d]$, we denote

$$G_S(y') = \sum_{Q \subseteq R, Q \cap [d] = S} \hat{g}_Q \chi_{Q-S}(y') \,.$$

Since $\hat{g}_Q = \mathbf{E}_y\left[g(y)\chi_Q(y)\right]$, we have

$$G_S(y') = \mathop{\mathbf{E}}_{y_1,y_2,\dots,y_d}\left[g(y_1,\dots,y_d,y')\chi_S(y_1,\dots,y_d)\right],$$

therefore G_S is bounded in $[-1,1]$. Similarly, so are $F_\emptyset$ and F_1. Also, for the Fourier expansion of the noisy functions, we have

$$\begin{aligned} T_{1-\delta'}f(x) &= T_{1-\delta'}F_\emptyset(x') + (1-\delta')x_1 T_{1-\delta'}F_1(x'), \\ T_{1-\delta}g(y) &= \sum_{S\subseteq[d]}(1-\delta)^{|S|}\chi_S(y_1,\dots,y_d)T_{1-\delta}G_S(y'). \end{aligned}$$

Lemma 4.26. *For any functions $F:\{-1,1\}\to\mathbb{R}$ and $G_i:\{-1,1\}^d\to\mathbb{R}$, we have*

$$\begin{aligned} &\mathop{\mathbf{E}}_{\mathcal{H}_\gamma(d)}\left[F(x)\prod_{i=2}^k G_i(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{I}_\gamma(d)}\left[F(x)\prod_{i=2}^k G_i(y_i)\right] \\ &= \sum_{S\subseteq[d],|S| \text{ is odd}}(1-\gamma)\widehat{F}_{\{1\}}\prod_{i=2}^k \widehat{G}_{i,S}. \end{aligned}$$

Proof. Taking Fourier expansion for the left hand side, we have

$$\begin{aligned} &LHS \\ &= \sum_{\substack{U\subseteq[1]\\ V_i\subseteq[d]}}\widehat{F}_U\prod_{i=2}^k\widehat{G}_{V_i}\left(\mathop{\mathbf{E}}_{\mathcal{H}_\gamma(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{I}_\gamma(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right]\right). \end{aligned}$$

Since $\mathcal{H}_\gamma(d) = (1-\gamma)\mathcal{H}(d)+\gamma\mathcal{N}(d)$, and $\mathcal{I}_\gamma(d) = (1-\gamma)\mathcal{I}(d)+\gamma\mathcal{N}(d)$, by linearity of expectation, we have

$$\begin{aligned} &\mathop{\mathbf{E}}_{\mathcal{H}_\gamma(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{I}_\gamma(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right] \\ &= (1-\gamma)\left(\mathop{\mathbf{E}}_{\mathcal{H}(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right] - \mathop{\mathbf{E}}_{\mathcal{I}(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right]\right). \end{aligned} \tag{4.5}$$

Note that $\mathcal{H}(d)$ and $\mathcal{I}(d)$ have the same marginal distribution on $\prod_{i=2}^k\{-1,1\}^d$, therefore for (4.5) to be nonzero, it must be that $U=\{1\}$.

For the expectation under $\mathcal{I}(d)$, we have

$$\mathop{\mathbf{E}}_{\mathcal{I}(d)}\left[\chi_U(x)\prod_{i=2}^k\chi_{V_i}(y_i)\right] = \mathop{\mathbf{E}}_{\mathcal{I}(d)}\left[\chi_U(x)\right]\mathop{\mathbf{E}}_{\mathcal{I}(d)}\left[\prod_{i=2}^k\chi_{V_i}(y_i)\right] = 0.$$

For $\mathcal{H}(d)$, it is easy to see that the expectation is zero unless for all $i\in\{2,\dots,k\}$, $V_i = V$ for some common V. Moreover, $|V|$ must be odd, and in this case, the expectation is 1. Thus we get the RHS of the statement. □

We denote $\mathcal{H}'_\gamma = \bigotimes_{l\neq 1} \mathcal{H}_\gamma(d)$. We now rewrite the LHS of (4.4) as

$$\left| \mathop{\mathbf{E}}_{\mathcal{H}'_\gamma} \left[\mathop{\mathbf{E}}_{\mathcal{H}_\gamma(d)} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] - \mathop{\mathbf{E}}_{\mathcal{J}_\gamma(d)} \left[T_{1-\delta'} f(x) \prod_{i=2}^{k} T_{1-\delta} g(y_i) \right] \right] \right| .$$

By Lemma 4.26, the above is equal to

$$\begin{aligned}
&\left| \sum_{S\subseteq[d],|S| \text{ is odd}} (1-\gamma)(1-\delta')(1-\delta)^{(k-1)|S|} \mathop{\mathbf{E}}_{\mathcal{H}'_\gamma} \left[T_{1-\delta'} F_1(x') \prod_{i=2}^{k} T_{1-\delta} G_S(y'_i) \right] \right| \\
\leq &\sum_{S\subseteq[d],|S| \text{ is odd}} (1-\gamma)(1-\delta')(1-\delta)^{(k-1)|S|} \mathop{\mathbf{E}}_{\mathcal{H}'_\gamma} \left[\left| T_{1-\delta'} F_1(x') \prod_{i=2}^{k} T_{1-\delta} G_S(y'_i) \right| \right] \\
\leq &\sum_{S\subseteq[d],|S| \text{ is odd}} (1-\delta')(1-\delta)^{(k-1)|S|} \|T_{1-\delta'} F_1\|_k \|T_{1-\delta} G_S\|_k^{k-1} ,
\end{aligned}$$

where the last step uses the generalized Hölder's Inequality (Theorem 2.4), and the norms $\|\cdot\|_k$ are with respect to the corresponding marginals of $\mathcal{H}'_\gamma$, which are uniform.

The following follows easily from the Hypercontractivity Theorem (Theorem 2.30).

Lemma 4.27. *For any function $f : \{-1,1\}^n \to [-1,1]$ and $0 < \eta < 1$*

$$\|T_{1-\eta} f\|_k \leq \|T_{1-\eta/2} f\|_2^{(2+\eta)/k} .$$

Proof. Let $\eta' = \eta/2$ and $f' = T_{1-\eta/2} f$. Observe that

$$\begin{aligned}
\|T_{1-\eta'} f'\|_k &= \mathbf{E}\left[|T_{1-\eta'} f'|^k \right]^{1/k} \\
&\leq \mathbf{E}\left[|T_{1-\eta'} f'|^{2+2\eta'} \right]^{1/k} = \|T_{1-\eta'} f'\|_{2+2\eta'}^{(2+2\eta')/k} .
\end{aligned}$$

Since $2 + 2\eta' \leq (1-\eta')^2 + 1$, by Theorem 2.30, the above is upper-bounded by $\|f'\|_2^{(2+2\eta')/k}$. Finally, note that

$$\|T_{1-\eta} f\|_k \leq \|T_{1-\eta'} T_{1-\eta'} f\|_k \leq \|T_{1-\eta/2} f\|_2^{(2+\eta)/k} .$$

□

Since F_1 and G_S are bounded in $[-1,1]$, we can apply the above and get

$$\|T_{1-\delta'} F_1\|_k \|T_{1-\delta} G_S\|_k^{k-1} \leq \|T_{1-\delta'/2} F_1\|_2^{(2+\delta')/k} \|T_{1-\delta/2} G_S\|_2^{(k-1)(2+\delta)/k} .$$

Expressing G_S as Fourier coefficients of g, we have

$$\begin{aligned}\|T_{1-\delta/2}G_S\|_2^2 &= \sum_{Q\subseteq R, Q\cap[d]=S}(1-\delta/2)^{2|Q|-2|S|}\hat{g}_Q^2\\ &\le \sum_{S\subseteq Q\subseteq R}(1-\delta/2)^{2|Q|-2|S|}\hat{g}_Q^2\\ &\le (1-\delta/2)^{-2|S|}\cdot \mathrm{Inf}_S(T_{1-\delta'/2}g)\,,\end{aligned}$$

where in the last step we used $\delta \ge \delta'$. For F_1, we relate it with f as follows

$$\|T_{1-\delta'/2}F_1\|_2^2 \le (1-\delta'/2)^2\cdot \mathrm{Inf}_1(T_{1-\delta'/2}f)\,.$$

Plugging this two upper-bounds back, we have that the LHS is at most

$$\sum_{S\subseteq[d],\,S\text{ is odd}} \mathrm{Inf}_1(T_{1-\delta'/2}f)^{(2+\delta)/2k}\cdot \mathrm{Inf}_S(T_{1-\delta'/2}g)^{(k-1)(2+\delta)/2k}\,,$$

where we also used $\delta \ge \delta'$. By the hypothesis that

$$\min\left\{\mathrm{Inf}_1(T_{1-\delta'/2}f), \mathrm{Inf}_S(T_{1-\delta'/2}g)\right\} \le \tau\,,$$

either $\mathrm{Inf}_1(T_{1-\delta'/2}f)^{\delta/2k} \le \tau^{\delta/2k}$, or $\mathrm{Inf}_S(T_{1-\delta'/2}g)^{(k-1)\delta/2k} \le \tau^{(k-1)\delta/2k}$ for each S in the sum. In either case, we can bound the above by

$$\begin{aligned}&\tau^{\delta/2k}\cdot \sum_{S\subseteq[d],\,S\text{ is odd}} \mathrm{Inf}_1(T_{1-\delta'/2}f)^{1/k}\cdot \mathrm{Inf}_S(T_{1-\delta'/2}g)^{(k-1)/k}\\ \le\ &\tau^{\delta'/2k}\cdot \sum_{S\subseteq[d],\,S\text{ is odd}} \left(\mathrm{Inf}_1(T_{1-\delta'/2}f) + \mathrm{Inf}_S(T_{1-\delta'/2}g)\right)\\ \le\ &\tau^{\delta'/2k}\cdot \left(2^d\,\mathrm{Inf}_1(T_{1-\delta'/2}f) + \sum_{S\subseteq[d],\,S\text{ is odd}} \mathrm{Inf}_S(T_{1-\delta'/2}g)\right)\,.\end{aligned}$$

This completes the proof. □

Chapter 5

Hardness of Gap$_s$-k-CSP

In this chapter, we study the complexity of Boolean Gap$_s$-k-CSP problems, that is, to understand how hard it is to distinguish satisfiable k-CSP instances from k-CSP instances that are far from satisfiable. In particular, we are interested in how the soundness parameter s decreases as the arity k grows. Hardness results of this type are usually proved by giving a predicate P of arity k that has few accepting inputs, and show that P is *approximation resistant on satisfiable instances*, that is, Gap$_{\rho(P)+\varepsilon}$-P is NP-hard.

Let us consider the predicate of arity k that is a conjunction of $k/3$ E3-Sat constraints. We can derive the approximation resistance of this predicate from the approximation resistance of E3-Sat. The density of this predicate is $(7/8)^{k/3} \approx 2^{0.94k}/2^k$. Although it tends to 0 as $k \to \infty$, it is quite far from the best hardness one would expect.

In this chapter, we give a predicate P of arity k that has $2^{\widetilde{O}(k^{1/3})}$ accepting assignments, and prove that Gap$_{\rho(P)+\varepsilon}$-P is NP-hard. This is an improvement over the best previous known ratio of $2^{O(k^{1/2})}/2^k$ by Håstad and Khot [55], though still a long way from the performance of the best algorithm, which only achieves around $\Theta(k)/2^k$.

In fact, even for arity as small as $k = 4, 5$, there is no complete characterization of hardness of Gap$_s$-k-CSP. Håstad [51] proved that Ek-Sat is approximation resistant on satisfiable instances. In [55], the authors started by showing that for a certain Boolean predicate on 5 variables with 2^4 accepting inputs, distinguishing between satisfiable instances and $(1/2 + \varepsilon)$-satisfiable instances is hard. Then, they apply an iterated construction not too different from that in [96] to get a sparse predicate that is hard to approximate. Assuming the d-to-1 Conjecture, O'Donnell and Wu proved a strong result in [90] that the NotTwo predicate is approximation resistant on satisfiable instances. The density of NotTwo is $\rho(\text{NotTwo}) = 5/8$, and there is a 5/8-approximation algorithm for Max-3-CSP by Zwick [106]. Their approach was generalized by Tang [100] to Max-3-CSP$_q$ where q is a prime greater than 3, and in Chapter 4, we extended their method to

show approximation resistance for Boolean predicates of arity $k \geq 3$ that accepts a strict superset of inputs of odd parity.

Recently, Håstad [54] and Wenner [104] proved approximation resistance for the above predicates without assuming the d-to-1 Conjecture. Their proofs are based on new analytic tools as well as Khot's Smooth-Label-Cover [68]. We note that several previous results that bypassed the UGC [68, 72, 38, 46] started from hardness of Smooth-Label-Cover.

We could relax the perfect completeness condition, and try to understand the complexity of Gap$_{1-\varepsilon,s}$-k-CSP. Designing algorithms for such problems may become much more challenging with this seemingly small change. As we have seen in Section 3.2, for the Ek-Lin predicate, we can solve Gap$_{1,1}$-Ek-Lin in polynomial time, whereas for Gap$_{1-\varepsilon,s}$-Ek-Lin, even doing something non-trivial for $s = 1/2 + \varepsilon$ is NP-hard.

The other side of this is that proving hardness for Gap$_{1-\varepsilon,s}$-k-CSP might seem somewhat more tractable. This is still, by no means, an easy task. In [97], Samorodnitsky and Trevisan showed approximation resistance of the Hadamard$_K$ predicate assuming the UGC. This also implies a UG-hardness of $O(K)/2^K$ for general k-CSP, matching, up to multiplicative constant factor, the performance of the algorithm by Charikar, Makarychev and Makarychev [25]. For some time, not much progress has been made in settling the NP-hardness of Gap$_{1-\varepsilon,s}$-k-CSP. Indeed, until recently, the best soundness for Gap$_{1-\varepsilon,s}$-k-CSP assuming NP-hardness has been $2^{O(k^{1/2})}/2^k$ by Samorodnitsky and Trevisan [96]. Engebretsen and Holmerin [36] later improved the constant in the exponents and pointed out some technical difficulties in getting better hardness than $2^{O(k^{1/2})}/2^k$ using certain kind of PCP reduction. A major advancement came recently, when Siu On Chan proved the NP-hardness of Gap-Hadamard$_K$. Chan introduced the idea of using direct sums of PCPs to improve soundness, which worked very well for predicates that are subgroups of a domain. In particular, the accepting assignments of the Hadamard$_K$ predicate is a subgroup under elementwise product, and Chan's result implies that it is approximation resistant assuming only $\mathsf{P} \neq \mathsf{NP}$.

On the face of it, Chan's new idea does not seem to be applicable for Gap$_s$-k-CSP. This is because in the setting of [24], for his direct sum technique to work, the predicates need to be a subgroup of the domain. For Boolean predicates, this means that the accepting inputs of a predicate form a linear subspace, and as we should all be familiar by now, Gap$_s$-k-CSP with these predicates can be decided in polynomial time.

It is therefore an interesting question whether we could combine these recent developments to get approximation resistance result for Max-P on satisfiable instances for predicate P sparser than the one in Håstad and Khot [55].

An immediate proposal to achieve tight lower-bound for Max-k-CSP on satisfiable instances would be to construct predicates as in [59, 104], that is, adding a single additional accepting assignment to the Hadamard$_K$ predicate of arity $2^k - 1$. However, this simple approach does not work—the accepting inputs of

HADAMARD$_K$ form a k-dimensional subspace, so if we add d new accepting inputs to it and get some other predicate P', we only need a $(k+d)$-dimensional subspace to contain all the accepting inputs of P'. Let Q be the predicate that accepts exactly all inputs from this $(k+d)$-dimensional subspace. Given any satisfiable MAX-P' instance with satisfying assignment α, we replace the predicate P' in each constraint with the predicate Q. The solution space of the instance with predicate Q is just a linear subspace satisfying the following:

- It contains the solution α to the original instance with predicate P'.
- If we project the solution space to the set of variables in each constraint, the resulting subspace has dimension at most $(k+d)$.

Therefore, if we sample a random point from this linear subspace, then for each constraint, the probability that we hit α restricted to the variables in that constraint (and hence satisfy the constraint with predicate P') is at least $1/2^{k+d}$. Thus whenever $d = o(2^k)$, the *expected* performance of the above sampling method beats simple random assignment, which only gives $(2^k+d)/2^{2^k}$.

The problem with adding more accepting assignments to HADAMARD$_K$ is that the resulting predicate does not have the group structure as in [24]. If we still take many rounds of direct sums as in [24], then to ensure perfect completeness, we need to accept many assignments that are products of the additional assignments we added and end up with a predicate that has more accepting assignments than we would want. On the other hand, as is demonstrated in [24], having more rounds of direct sum helps us to improve soundness dramatically and so if we are looking for sparse predicates that are approximation resistant, it would be natural to have more rounds of proofs in the direct sum.

In this chapter, we attempt to strike a balance. The following is a formal statement of the theorem we prove in this chapter.

Theorem 5.1. *There is a predicate P of arity K with density $2^{\widetilde{O}(K^{1/3})}/2^K$, for which* GAP$_{1,\rho(P)+\varepsilon}$-$P$ *is* NP*-hard for any constant $\varepsilon > 0$.*

The construction is based on many ideas developed in a number of previous works, including [36, 104, 24]. On the highest level, we use direct sum of several PCPs to get improved soundness result. However, as argued above, we also want to limit the number of PCPs involved. Therefore, we use LONG-CODE-based PCP constructions that are already rather efficient, for example those used by Engebretsen and Holmerin [36]. In [104], Wenner showed how different types of noise operators behave similarly when the reduction is based on SMOOTH-LABEL-COVER. This is helpful when analyzing soundness of PCPs in that it allows us to move from correlated noise with perfect completeness to independent noise that are not perfect but easier to analyze. We also use a multivariate invariance theorem in [104], which extends methods of Mossel et al. [87, 86] to projection games. Similar techniques were developed also in other works such as [89] as well as in [24].

5.1 Chan's Direct Sum of PCPs

Before we describe our construction, let us first have a look at the main ideas in Chan's work [24].

Recall that for $r \in \mathbb{N}^+$ and $K := 2^r - 1$, the HADAMARD$_K$ predicate is defined on variables $\{x_S\}_{\emptyset \neq S \subseteq [r]}$, and the value of the predicate is defined as

$$\text{HADAMARD}_K(x) = \begin{cases} 1 & \forall S \subseteq [r], |S| > 1, x_S = \prod_{i \in S} x_{\{i\}} \\ 0 & \text{otherwise.} \end{cases}$$

In [24], Chan proved that HADAMARD$_K$ is approximation resistant assuming $\mathsf{P} \neq \mathsf{NP}$.

In Section 4.1, we described a typical reduction for inapproximability results, where one first sample one edge from some LABEL-COVER instance and use a set of constraints to test whether the assignment to the variables correspond to a LONG-CODE encoding of a good labeling. Briefly speaking, in Chan's direct sum PCP, we now sample K edges and run K independent copies of the above test. In the i-th PCP, the i-th query is a uniform random string from $\{-1,1\}^L$ and all other queries are sampled from $\{-1,1\}^R$ as described below in Definition 5.3. In a correct proof, the strategies are expected to be products of LONG-CODE encoding the labeling of the vertices.

We now formally define the PCP and how queries are sampled. In the following description, we identify integers from $[K]$ with non-empty subsets of $[r]$ in some canonical way. First we describe the test distribution for a single PCP, indexed by non-empty sets $\emptyset \neq S \subseteq [r]$.

Definition 5.2. *Let e_S be an edge and π be the constraint on e. Denote the set of possible queries to the T-th position by Q_T, where*

$$Q_T = \begin{cases} \{-1,1\}^L & T = S \\ \{-1,1\}^R & T \neq S. \end{cases}$$

The test distribution $\mathcal{T}_{S,e_S}$ is a distribution on $\prod_{\emptyset \neq T \subseteq [r]} Q_T$. To sample query $(q_T)_{T \subseteq [r]}$ from $\mathcal{T}_{S,e_S}$, first sample q_S from $\{-1,1\}^{|L|}$ uniformly at random. Then, for each $i \in [R]$, let $\{q_{T,i}\}_{T \neq S}$ be a uniformly random accepting assignment of HADAMARD$_K$*, conditioned on the S-th bit being equal to $q_{S,\pi(i)}$. Finally, independently for each bit, we add noise by resampling from the uniform distribution on $\{-1,1\}$ with probability η.*

The final test distribution in the PCP is a product of the above distribution.

Definition 5.3. *Let (U,V,E,L,R,Π) be a* LABEL-COVER *instance. For $i \in [K]$, define $\mathcal{V}_i = V^{i-1} \times U \times V^{K-i}$. For each $\mathbf{v} \in \mathcal{V}_i$, the proof contains function $\mathbf{f}_\mathbf{v} : \left(\{-1,1\}^R\right)^{i-1} \times \{-1,1\}^L \times \left(\{-1,1\}^R\right)^{K-i} \to \{-1,1\}$. The verifier checks the proof as follows:*

1. *Sample independently* $K = 2^r - 1$ *uniformly random edges* $\{e_S\}_{\emptyset \neq S \subseteq [r]}$. *Denote* $e_S = \{u_S, v_S\}$.

2. *Sample queries* $\{\mathbf{q}_i\}_{i=1}^K$ *from distribution* $\prod_{\emptyset \neq T \subseteq [r]} \mathcal{T}_{T, e_T}$.

3. *Let* $\mathbf{v}_i = (v_1, \ldots, v_{i-1}, u_i, v_{i+1}, \ldots, v_K)$. *Accept if*

$$\textsc{Hadamard}_K(\mathbf{f}_{\mathbf{v}_1}(\mathbf{q}_1), \ldots, \mathbf{f}_{\mathbf{v}_K}(\mathbf{q}_K)) = 1\,.$$

In a correct proof, the function $\mathbf{f}_{\mathbf{v}}$ is the product of Long-Code encodings of the labeling of each vertex in $\mathbf{v}$. That is, suppose we have a labeling σ for the Label-Cover instance, then for $\mathbf{v} = (v_1, \ldots, v_{i-1}, u_i, v_{i+1}, \ldots, v_K)$, we expect

$$\mathbf{f}_{\mathbf{v}} = \left(\prod_{j=1}^{i-1} \chi_{\{\sigma(v_j)\}} \right) \cdot \chi_{\{\sigma(u_i)\}} \cdot \left(\prod_{j=i+1}^{K} \chi_{\{\sigma(v_j)\}} \right) .$$

It should be clear that for the completeness case to hold, it is important that the element-wise product of K accepting inputs has to be an accepting input.

Remark. *As in the ordinary case, we require that the functions* $\mathbf{f}_{\mathbf{v}}$ *are folded in the following sense — for any* $j \in [K]$, *query* $\{\mathbf{q}_{j,i}\}_{i \in [K]}$ *and* $i_0 \in [K]$ *we have*

$$\begin{aligned} & \mathbf{f}_{\mathbf{v}}(\mathbf{q}_{j,1}, \ldots, -\mathbf{q}_{j,i_0}, \ldots, \mathbf{q}_{j,K}) \\ = \quad & -\mathbf{f}_{\mathbf{v}}(\mathbf{q}_{j,1}, \ldots, \mathbf{q}_{j,i_0}, \ldots, \mathbf{q}_{j,K})\,. \end{aligned}$$

Theorem E.1 along with Theorem A.1, 6.9 and C.2 of Chan [23] shows completeness and soundness of the above reduction and we summarize in the following theorem.

Theorem 5.4. *Fix some small* $\eta, \delta > 0$. *Let* σ *be the soundness of* Label-Cover, *satisfying* $\delta = \mathit{poly}(K/\eta) \cdot \sigma^{\Omega(1)}$. *Given a* Label-Cover *instance* $LC_{L,dL}$, *we have the following:*

1. *If* $LC_{L,dL}$ *has value 1, the above verifier accepts a correct proof with probability at least* $1 - K^2\eta$.

2. *If* $LC_{L,dL}$ *has value at most* σ, *then given any proof the verifier accepts with probability at most* $(K+1)/2^K + 2\delta$.

5.2 Proof Overview

Fix some k. Given ε, the starting point of our reduction is a $(J, \xi)-$Smooth-k-Multi-Layered-Label-Cover, where J and ξ are constants solely dependent on ε and k that we will specify later.

The predicate. Let

$$\mathcal{S}_3 := \{S \subseteq [k] \mid |S| = 3\} \qquad \text{and} \qquad \mathcal{S}_1 := \{S \subseteq [k] \mid |S| = 1\}.$$

The predicate is on variables $\{x^{(S)}\}_{S \in \mathcal{S}_1 \cup \mathcal{S}_3}$ taking values from $\{-1, 1\}$. We call the variables $x^{(\{i\})}$ singleton variables and the remaining ones parity check variables. The predicate accepts if there exists $\vec{w} \in \{-1, 1\}^{\mathcal{S}_1 \cup \mathcal{S}_3}$ such that the number of -1 entries in $\vec{w}$ is no more than k, and

$$w_S x^{(S)} \cdot \prod_{i \in S} w_{\{i\}} x^{(\{i\})} = 1$$

for all $S \in \mathcal{S}_3$.

We can view $\vec{w}$ as an error vector, and the predicate accepts inputs that are no more than Hamming distance k away from an assignment that satisfies all parity checks.

The predicate is on $k + \binom{k}{3}$ variables, and it has

$$O\left(2^k \cdot \binom{\binom{k}{3} + k}{k}\right) = 2^{O(k \log k)}$$

accepting inputs, thus if we denote the arity of the predicate by K, then its density is $2^{\widetilde{O}(K^{1/3})}/2^K$, where the $\widetilde{O}$ hides logarithmic factors.

Our reduction is based on direct sums of PCPs as described in Section 5.1. We prove that all non-constant terms in the Fourier expansion of Equation (4.1) are small.

We first point out several challenges in applying the direct sum technique in our case.

One crucial difference between Chan's proof and ours is that we require perfect completeness. This means that sometimes there would be perfect correlation between certain queries which makes it possible for provers to find good cheating strategies. In Chan's proof as well as in many related results where perfect completeness is not required, one can usually break this correlation by applying some independent noise to each query bit. However, in the case of perfect completeness, we cannot afford perturbing each bit independently, and thus we need to take extra care when designing test distributions. That is the main reason our predicate accepts inputs that *almost* satisfy all $\binom{k}{3}$ linear constraints. In some sense, these extra accepting inputs serve as noise that breaks up perfect correlations.

Another important property that Chan uses is the "group" structure of the predicate. This makes it relatively easy to take direct sums of a large number of PCPs, each handling a small number of non-constant terms from Equation (4.1), without worrying too much about the completeness of the resulting PCP. Our predicate, however, does not satisfy this property due to the extra "noise" we added. It is certainly possible that if we take the sum of two assignments that are of distance k away from assignments that satisfies all linear equations, we end

up with something that is distance $2k$ away from an assignment that satisfies all linear constraints, and that breaks perfect completeness. To avoid this situation, we limit both the number of PCPs in the direct sum and in each PCP the distance from an assignment that satisfies all linear constraints. More specifically, in our construction, the queries to each PCP are generated such that if the provers (of each individual PCP) answer according to some consistent Long-Code, then the answers is at most distance 1 away from an assignment that satisfies all linear equations. When taking direct sum of the k PCPs, an answer that is the direct sum of k Long-Code would give us an answer at most distance k away from satisfying all linear equations, which is exactly what would be accepted by our predicate.

It remains to find a number of suitable PCPs. If we try to generalize previous approaches, for example those in [96, 36], to larger predicates such as Hadamard_K, it is instructive to look at the main obstacles there. One of the main adversarial strategies that we need to consider is that of inconsistent Long-Code encodings, that is, the assignments represent valid Long-Code encoding of labelings, but the labelings do not satisfy the projection constraints on the edges.. For example, consider a predicate P on variable $(x_1, \ldots, x_k)$ and a simple PCP reduction where we sample an edge $\{u, v\}$ and query functions f^u and g^v according to some test distribution $\mathcal{T}$ as described in Section 4.1. For simplicity, assume that the query to f^u corresponds to input variable x_1, and the remaining queries are on g^v. Suppose further that for a $1/2+\delta$ fraction of the accepting inputs of P, we have $x_2x_3x_4 = 1$ (both Hadamard_K and the predicate we are studying here have properties similar to this.) Let g^v be long code for some arbitrary label $r \in R$. Observe that the non-constant term $g^v(x_2)g^v(x_3)g^v(x_4)$ will always have expectation roughly order of δ simply due to the requirements on $\mathcal{T}$. In this case, we get a large non-constant term but it does not help us find a consistent labeling for Label-Cover. A similar argument can be made for Multi-Layered-Label-Cover. Chan's construction in [24] solves this problem by making sure that for each term, in at least one of the many PCPs in the direct sum the queries are on different functions. As discussed before, since we are aiming for fewer PCPs in the direct sum, it would be good if each PCP can carry out as many consistency checks as possible, and Multi-Layered-Label-Cover becomes a very natural choice. We also need to decide which query should be in which layer for each PCP so that we do not miss any sets of variables that has linear relations. This is mostly done in Section 5.3.1.

Now we describe the PCPs in more details.

The PCPs. Let $\mathcal{C} = \{\sigma_0, \ldots, \sigma_{k-1}\}$ be the set of cyclic permutations on $[k]$. The permutation σ_i maps i to k, $i+1$ to 1, and so on. We identify 0 with k, and thus σ_0 is the identity permutation. Each permutation corresponds to a PCP for a k-Layered-Label-Cover instance, and the permutation decides which query should be in which layer in the Multi-Layered-Label-Cover. We design a test distribution for each permutation. As stated above, the final proof is the direct sum of these k PCPs.

We now describe the i-th PCP. It is based on a k-Layered-Label-Cover

instance, and there are $k+\binom{k}{3}$ queries, each corresponding to an input variable. We denote the queries as $x^{(S)}$. For $S \in \mathcal{S}_1 \cup \mathcal{S}_3$, define $m_i(S) \coloneqq \max \sigma_i(S)$ to be the maximum element of S under permutation σ_i. The query $x^{(S)}$ is in layer $m_i(S)$. Let

$$\mathcal{V}_i(S) \coloneqq U^{k-m_i(S)} \times V^{m_i(S)-1}$$

be the set of vertex tuples in layer $m_i(S)$. The proof has a function for each vertex tuple in $\mathcal{V}_i(S)$, and the input to the functions are $\{-1,1\}$ strings indexed by the labelings $L^{k-m_i(S)} \times R^{m_i(S)-1}$ in layer $m_i(S)$. We denote the domain of the functions as $X_i^{(S)}$. In a correct proof of a correct labeling, the function would be a Long-Code encoding a proper labeling for all vertices in the tuple. We require that all functions are folded over constant.

The test distributions. We first define the test distributions for each individual PCP.

Fix $i \in [k]$ and consider the i-th PCP. For notational simplicity we omit i in the subscript for now. We first independently sample $k-1$ edges $\vec{e} = \{e_1, \dots, e_{k-1}\}$. For $S \in \mathcal{S}_1$, sample $x^{(S)} \in X^{(S)}$ uniformly at random. For $S = \{s_1, s_2, s_3\} \in \mathcal{S}_3$, let $m = m(S)$ be the layer in which query $x^{(S)}$ is located, $m_j = m(s_j)$ for $j = 1, 2, 3$ be the layer query $x^{(\{s_j\})}$ is in, and set

$$x_r^{(S)} = \prod_{j=1}^{3} x^{(\{s_j\})}_{\pi_{\vec{e},m\to m_j}(r)}$$

for all labeling $r \in L^{k-m} \times R^{m-1}$.

We then make use of the extra inputs allowed by the predicate to add some "noise" to the distributions. As discussed above, the resulting distribution must have the property that the output obtained by applying some consistent Long-Code is at most distance 1 away from an assignment that satisfies all $\binom{k}{3}$ equations. The idea is to perturb one of the variables $x^{(S)}$. For each $r \in L^{k-1}$, pick a uniformly random set $N_r \in \mathcal{S}_1 \cup \mathcal{S}_3$, and for each

$$t \in \pi^{-1}_{\vec{e},m(N_r)\to 1}(r)\,,$$

set $x_t^{(N_r)}$ to a uniform random bit independently with probability $1/2$.

We denote the test distribution by $\mathcal{T}$. For each $r \in L^{k-1}$, let $\mathcal{T}_r$ be the marginal distribution of the bits that map to r under $\pi_{\vec{e},l\to 1}$ for all $l \in [k]$. Observe that we have $\mathcal{T} = \bigotimes_{r\in L^{k-1}} \mathcal{T}_r$.

Let us start by analyzing the standard completeness case.

Lemma 5.5. *For any sampling of edges, let $f^{(S)}$ be the functions we are querying, and let $x^{(S)}$ be the corresponding queries. If the k-layered* Label-Cover *instance has a labeling that satisfies all the edges, then we can find functions $f^{(S)}$ such that the answers*

$$\left\{f^{(S)}(x^{(S)})\right\}_{S\in\mathcal{S}_1\cup\mathcal{S}_3}$$

is at most Hamming distance 1 away from an assignment that satisfies all linear constraints on 3 *singleton variables and* 1 *parity check variable.*

Proof. The argument is similar to a standard completeness argument.

Fix a labeling that satisfies all the edges. The proof in the PCP consists of Long-Code encoding the labeling of all hybrid vertex tuples.

Let $r \in L^{k-1}$ be the labeling for the vertex tuple in layer 1. The answers we get from the long codes is the same as returning one bit from each query generated according to $\mathcal{T}_r$. The claim follows by observing that for each tuple of bits produced as above, either it already satisfies all linear constraints, or it would satisfy all linear constraints after we flip the N_r-th bit. □

Denote the test distribution of the i-th PCP defined above as $\mathcal{T}_i$. The distribution of the final composed PCP is simply the product of the individual test distributions $\bigotimes_{i=1}^{k} \mathcal{T}_i$. The verifier samples the edges and the inputs to the functions, queries the functions (those that correspond to the chosen vertex tuples) and accepts if the answers returned by the functions are accepted by the predicate.

It is not hard to see from above discussions that the above PCP has perfect completeness.

Lemma 5.6. *If the* k-Layered-Label-Cover *instance has a labeling that satisfies all edges, then there exists a set of functions* $\{f^{(S)}\}$ *such that after querying* $\{f^{(S)}\}$ *the verifier accepts with probability 1.*

Proof. We let our final proof be the product of proofs of the k individual PCPs given by Lemma 5.5. Since the answer for each proof is at most distance 1 away from an assignment that satisfies all linear constraints, their product is at most distance k away, which is exactly what the verifier (and our predicate) accepts. □

Remark. *We briefly discuss some of the difficulties in getting hardness result better than* $2^{\widetilde{O}(K^{1/3})}/2^K$. *The key issue here is not unlike the one discussed in [36]. As we will see in Section 5.3, our construction requires us to identify a permutation cover that covers all the queries, otherwise products of a random set of dictatorship functions would be a good cheating strategy.*

To be more specific, one possibility is to consider a predicate on $k+\binom{k}{4}$ *variables which does "parity checks" on tuples of 4 variables and accepts everything that has less than* t *errors for some* t*, and devise a protocol which is a direct sum of* t *PCPs. As we can see from Section 5.3.1, in this case we actually need* $\Omega(k^2)$ *permutations to cover all queries which means that* $t = \Omega(k^2)$*. Such a predicate already has much more accepting inputs than the one we study here.*

5.3 Soundness Analysis

In this section, we analyze the soundness of our PCP. We set

$$\begin{aligned}\varepsilon_1 &= \varepsilon/(7k^3+1)\,,\\ \xi &= \varepsilon_1^2\,,\\ \rho_0 &= 1-1/4\binom{k}{3}\,,\\ J &= 2\lceil \log_{\rho_0}\varepsilon_1\rceil\,,\end{aligned}$$

and γ such that $1-(1-\gamma)^{J/2} < \varepsilon_1$. Note that this gives $\rho_0^{J/2} \le \varepsilon_1$, and that all parameters depend only on k and ε. Also $\gamma < \varepsilon$.

As discussed in Section 5.2, we would like to prove that for all $\mathcal{S} \neq \emptyset$, the expectation

$$\mathbf{E}\left[\prod_{S\in\mathcal{S}} f^{(S)}(x^{(S)})\right] \tag{5.1}$$

is small unless there is good labeling.

Remark. *Since we are able to bound (5.1) for any $\mathcal{S} \neq \emptyset$, we actually proved that our predicate is useless in the sense of [9].*

Remark. *The functions $f^{(S)}$ actually depend on the underlying edges we sampled. For notational convenience we suppress this dependency and save another layer of subscripts (of subscripts of subscripts).*

As discussed in previous sections, we need to show that for each non-constant term, there is at least one PCP among those in the direct sum, such that if the expectation of the term under the PCP is large, we can find a good labeling for the underlying Label-Cover instance by looking at the functions f restricted to that PCP. Formally, we have the following lemma which is a reformulation of Lemma 5.3 in Chan [24].

Lemma 5.7. *Let $\mathcal{T} = \bigotimes_{i=1}^{k} \mathcal{T}_i$, where $\mathcal{T}_i$ is the test distribution for the i-th PCP. Suppose for some $\mathcal{S} \neq \emptyset$, we have*

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}}\left[\prod_{S\in\mathcal{S}} f^{(S)}(x^{(S)})\right]\right| = \delta\,,$$

then for any $i \in [k]$, there exists functions $g^{(S)}$ whose inputs are query bits to the i-th PCP, such that

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}_i}\left[\prod_{S\in\mathcal{S}} g^{(S)}(x^{(S)})\right]\right| \ge \delta\,.$$

Given $f^{(S)}$, we find $g^{(S)}$ by fixing query bits that are not in the i-th PCP in a way that does not lower the expectation.

Thus to bound each term, we need to carefully find an i, such that the test restricted to the i-th PCP has small expectation. We show how to choose such i in Section 5.3.1. We would be back to the traditional setting with Label-Covers and dictatorship testing from then on. In Section 5.3.2, we show that we can instead look at the distribution where each individual bit is further perturbed independently by some random noise. The way random noise is introduced is a bit similar to the way we did it in Chapter 4. Then we show in Section 5.3.3 how to apply an invariance-type theorem from [104] in this new setting to get our soundness result.

5.3.1 Permutation covering

Our k PCPs use cyclic permutations $C \in \mathcal{C}$ to decide the layer of each query and the inputs to the corresponding function. We first give a general definition of the crucial property we need from such sets of permutations.

Definition 5.8. *Let $\mathcal{P}$ be a set of permutations on $[k]$. We say that $\mathcal{P}$ covers $\mathcal{S}_1 \cup \mathcal{S}_3$ if for all $\emptyset \neq \mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, there exists a permutation $\sigma \in \mathcal{P}$, some $j, l_0 \in [k]$, such that*

$$\Big| \{S \in \mathcal{S} \mid j \in S, \max \sigma(S) = l_0\} \Big| \text{ is odd}.$$

We now reformulate the above definition and prove a necessary and sufficient condition for general sets of permutations $\mathcal{P}$ to cover $\mathcal{S}_1 \cup \mathcal{S}_3$.

For each set $S \in \mathcal{S}_1 \cup \mathcal{S}_3$, we construct a Boolean vector $v_S^{\mathcal{P}}$ as the following: the elements in the vector are indexed by a tuple $(i, l, j) \in [|\mathcal{P}|] \times [k] \times [k]$, and $v^{\mathcal{P}}_{S,(i,l,j)} = 1$ if $\max \sigma_i(S) = l$ and $j \in S$, and $v_{S,(i,l,j)} = 0$ otherwise.

Proposition 5.9. *The set of permutations $\mathcal{P}$ covers $\mathcal{S}_1 \cup \mathcal{S}_3$ if and only if the vectors $\{v_S^{\mathcal{P}}\}_{S \in \mathcal{S}_1 \cup \mathcal{S}_3}$ are linearly independent over $\mathbb{F}_2$.*

Proof. If the set $\mathcal{P}$ does not cover $\mathcal{S}_1 \cup \mathcal{S}_3$, then there exists a set $\emptyset \neq \mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, such that for any permutation $\sigma_i \in \mathcal{P}$ and $j, l_0 \in [k]$, we have that

$$\Big| \{S \in \mathcal{S} \mid S \ni j, \max \sigma_i(S) = l_0\} \Big| \text{ is even}.$$

Observe that for any $S \in \mathcal{S}_1 \cup \mathcal{S}_3$, the segment of $v_S^{\mathcal{P}}$ indexed by (i, l) for some fixed i and l is all zero if $\max \sigma_i(S) \neq l$, and otherwise it is exactly the character vector of the set S. Therefore the above is equivalent to saying that for any $i \in [|\mathcal{P}|]$ and l_0, we have

$$\sum_{S \in \mathcal{S}} v^{\mathcal{P}}_{S,(i,l_0)} = 0,$$

where the summation is modulo 2. Since the above holds for all i and l_0, we have

$$\sum_{S \in \mathcal{S}} v_S^{\mathcal{P}} = 0,$$

or the vectors $\{v_S^{\mathcal{P}}\}_{S\in\mathcal{S}}$ are linearly dependent.

Note that all the above steps are equivalent statements. Thus the other direction also holds. □

Remark. *We can see from the above argument that it is necessary to have $\Omega(k)$ permutations in order to cover $\mathcal{S}_1 \cup \mathcal{S}_3$, because otherwise we would have $\Theta(k^3)$ vectors of dimension $o(k^3)$ and thus they could not be linearly independent.*

We now prove that the set of all cyclic permutations $\mathcal{C} = \{\sigma_0, \dots, \sigma_{k-1}\}$ covers $\mathcal{S}_1 \cup \mathcal{S}_3$.

Lemma 5.10. *The set of all cyclic permutations $\mathcal{C} = \{\sigma_0, \dots, \sigma_{k-1}\}$ covers $\mathcal{S}_1 \cup \mathcal{S}_3$.*

Proof. For any given collection of sets $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, we show how to find the cyclic permutation σ and indices $j, l_0 \in [k]$ required in Definition 5.8.

For a set $S \in \mathcal{S}_1 \cup \mathcal{S}_3$, let

$$\operatorname{span}(S) = \min_{\sigma_i \in \mathcal{C}} \{\max \sigma_i(S) - \min \sigma_i(S)\} ,$$

that is, the minimum distance between the largest and the smallest element under cyclic permutations. Note that for singleton sets $S \in \mathcal{S}_1$, we have $\operatorname{span}(S) = 0$.

For a given collection of sets $\mathcal{S}$, let $S \in \mathcal{S}$ be a set with minimum span in $\mathcal{S}$ where we break ties arbitrarily. Pick i_0 such that $\sigma_{i_0}(S)$ contains 1 and $\operatorname{span}(S)+1$ as its minimum and maximum element. Let $\sigma := \sigma_{i_0}$ be the permutation we want, and let $l_0 = \operatorname{span}(S) + 1$.

Now we select j. If $\operatorname{span}(S) = 0$, then let $j = \sigma^{-1}(1)$ and we are done. This is because for any non-singleton set S', $\max \sigma(S) > 1$, and for any singleton set $S'' \neq S$, clearly $\sigma(S'') \neq \sigma(S)$. Thus S would be the only set containing j with $\max \sigma(S) = 1 = l_0$.

If $\operatorname{span}(S) \neq 0$, then S has three elements, and there is no singleton set in $\mathcal{S}$. If there is any other non-singleton set $S'' \in \mathcal{S}$ with $\max \sigma(S'') = \operatorname{span}(S) + 1$, then $\sigma(S'')$ and $\sigma(S)$ have the same maximum and minimum element, namely $\operatorname{span}(S) + 1$ and 1. That leaves us with the middle element. But since $S \neq S''$, the middle element must be different, so each of them appear only in one set, and setting j to the inverse of any of the middle elements under σ would work. Otherwise we take $j = \max S$. □

For $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, we consider the PCP corresponding to the cyclic permutation $\sigma_i \in \mathcal{C}$ covering $\mathcal{S}$ given by Lemma 5.10. We denote the PCP as PCP$_i$. As discussed before, we only need to show that if (5.1) is large even when restricted to PCP$_i$, we can find a good labeling for the Label-Cover instance we started with.

For notational simplicity, we only prove the case where $i = 0$, that is, for the identity permutation σ_0. Arguments for general cyclic permutations are entirely symmetric.

5.3.2 Introducing Independent Noise

In this section, we show that perturbing the queries does not change the expectation of the terms by too much.

Formally, let $\mathcal{T}'_r$ be the distribution where we first sample according to $\mathcal{T}_r$, and then resample each bit independently with probability γ according to its marginal distribution in $\mathcal{T}_r$—which in our case is uniform. Also define $\mathcal{T}' = \bigotimes_{r \in L^{k-1}} \mathcal{T}'_r$. We prove the following lemma which bounds the difference of expectation of (5.1) under $\mathcal{T}$ and $\mathcal{T}'$.

Lemma 5.11. *For any $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, we have*

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] - \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] \right| < 7k^3 \varepsilon_1 \,, \tag{5.2}$$

where $\varepsilon_1 = \varepsilon/(7k^3 + 1)$ is as defined at the beginning of Section 5.3.

Fix some $S_0 \in \mathcal{S}_1 \cup \mathcal{S}_3$. Let $\mathcal{T}^{(S_0)}$ be the distribution where under $\mathcal{T}$, we independently resample the bits in $x^{(S_0)}$ from the uniform distribution with probability γ. We first show in Lemma 5.12 below that the expectation under $\mathcal{T}$ is close to that under $\mathcal{T}^{(S_0)}$. Lemma 5.11 follows by applying similar arguments to each $x^{(S)}$ in succession.

For $S \in \mathcal{S}_1 \cup \mathcal{S}_3$, let $m(S) = \max S$ be the maximum element in S. Recall that query $x^{(S)}$ is located in layer $m(S)$, and for $r \in L^{k-1}$, $\mathcal{T}_r$ is the distribution containing all bits in

$$\left\{ x_t^{(S)} \;\middle|\; S \in \mathcal{S}_1 \cup \mathcal{S}_3, \pi_{m(S) \to 1}(t) = r \right\},$$

that is, the query bits that map to the same r. We use $\mathcal{T}_r^{(S_0)}$ to denote the marginal distribution of $\mathcal{T}^{(S_0)}$ on bits in

$$\left\{ x_t^{(S_0)} \;\middle|\; \pi_{m(S_0) \to 1}(t) = r \right\}.$$

Let $m = m(S_0)$.

Consider the difference of expectation between $\mathcal{T}$ and $\mathcal{T}^{(S_0)}$. If $f^{(S_0)}(x^{(S_0)})$ does not appear in the product, then there would be no difference. We now assume otherwise. The following lemma shows that introducing independent noise on one query does not change the expectation by too much.

Lemma 5.12. *For any $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, we have*

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] \right| < 7\varepsilon_1 \,. \tag{5.3}$$

In the rest of the section, we prove Lemma 5.12. The proof follows Wenner's approach [104], especially Lemmas 3.15 through 3.17.

For notational simplicity let F' be the product of all terms but $f^{(S_0)}(x^{(S_0)})$ and we abbreviate $f^{(S_0)}$ as f. We use $\overline{X}^{(S_0)}$ to abbreviate

$$\prod_{S\in\mathcal{S}_1\cup\mathcal{S}_3,\, S\neq S_0} X^{(S)}\,.$$

Similarly we define $\overline{X}_r^{(S_0)}$ for $r\in L^{k-1}$. The first step is to use Lemma 2.8 to bound the correlation

$$\rho(X^{(S_0)},\overline{X}^{(S_0)};\mathcal{T}) \qquad \text{and} \qquad \rho(X^{(S_0)},\overline{X}^{(S_0)};\mathcal{T}^{(S_0)})\,.$$

Since $\mathcal{T}$ is simply a product of $\mathcal{T}_r$ with different values r, by Lemma 2.6, we only need to bound

$$\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\mathcal{T}_r) \qquad \text{and} \qquad \rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\mathcal{T}_r^{(S_0)})\,.$$

Claim 5.13. *For any $S_0\in\mathcal{S}_3$, the correlation $\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\mathcal{T}_r)$ is upper-bounded by*

$$\rho_0 = 1-\frac{1}{4\binom{k}{3}}\,.$$

The same bound holds for $\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\mathcal{T}_r^{(S_0)})$.

Proof. We divide $\mathcal{T}_r$ into two parts: (i) the set S_0 is chosen as N_r; or (ii) some set other than S_0 is chosen. It is not hard to verify that the marginal of $X_r^{(S_0)}$ after conditioning on either of them remains uniform and thus we can apply Lemma 2.8. Let μ be the conditional distribution assuming (i) happens, and ν be the one assuming (ii) happens. We have that

$$\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\nu) = 1\,.$$

For the correlation of the other part, we have

$$\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\mu) = \frac{1}{2}\,,$$

achieved by dictatorship functions. Therefore, the overall correlation is upper-bounded by

$$\sqrt{\left(1-1/\binom{k}{3}\right)+1/\binom{k}{3}\cdot(1/2)^2} \le \sqrt{1-1/2\binom{k}{3}} < 1-1/4\binom{k}{3}\,.$$

Intuitively, the correlation under $\mathcal{T}_r^{(S_0)}$ could not exceed that under $\mathcal{T}_r$ since the noise we added are all independent. In particular, the part corresponding to

$$\rho(X_r^{(S_0)},\overline{X}_r^{(S_0)};\nu)$$

becomes less than 1 due to lack of perfect correlation, and the part corresponding to

$$\rho(X_r^{(S_0)}, \overline{X}_r^{(S_0)}; \mu)$$

remains the same. Thus the result follows by similar calculations as in $\mathcal{T}_r$. □

Take the Efron-Stein decomposition $f = \sum_{T \subseteq L^{k-1}} f_T$. More specifically, for $T \subseteq L^{k-1}$, we have that

$$f_T = \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ \pi_{m\to 1}(U)=T}} \hat{f}_U \chi_U \,.$$

Again for notational simplicity, we temporarily drop the subscript and write $\pi_{m\to 1}$ as π. We decompose the terms in the expectation in (5.3) as following

$$fF' \;=\; F' \sum_{T \subseteq L^{k-1}} f_T \;=\; F' \sum_{\substack{T \subseteq L^{k-1} \\ |T| \le J/2}} f_T + F' \sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} f_T \,. \tag{5.4}$$

We first bound the expectation of the high degree parts under both $\mathcal{T}$ and $\mathcal{T}^{(S_0)}$.

This is a standard correlation argument. We first consider the expectation under $\mathcal{T}$. Let $\mathcal{U}_{\mathcal{T}}$ be the conditional expectation operator mapping a function with domain $X^{(S_0)}$ to a function with domain $\overline{X}^{(S_0)}$ with respect to distribution $\mathcal{T}$. We have

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} f_T \right] \right| = \left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} \mathcal{U}_{\mathcal{T}} f_T \right] \right| . \tag{5.5}$$

Note that the expectation on the right hand side is in fact taken under the marginals of $\mathcal{T}$ on $\overline{X}^{(S_0)}$. Applying Cauchy-Schwarz and using the orthogonality of Efron-Stein decomposition, we have

$$\begin{aligned}
\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} f_T^{(S_0)} \right] \right| \;&\le\; \sqrt{\mathop{\mathbf{E}}_{\mathcal{T}} \left[\sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} \left(\mathcal{U}_{\mathcal{T}} f_T^{(S_0)} \right)^2 \right]} \sqrt{\mathop{\mathbf{E}}_{\mathcal{T}}[F'^2]} && (5.6) \\
&\le \sqrt{\sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} \left\| \mathcal{U}_{\mathcal{T}} f_T^{(S_0)} \right\|_2^2} && (5.7) \\
&\le \sqrt{\sum_{\substack{T \subseteq L^{k-1} \\ |T| > J/2}} \rho_0^{2|T|} \left\| f_T^{(S_0)} \right\|_2^2} \le \rho_0^{J/2} \le \varepsilon_1 \,, && (5.8)
\end{aligned}$$

where the inequality in (5.8) follows from Proposition 2.36 and that the norm in (5.8) is with respect to the marginal of $\mathcal{T}$ on $X^{(S_0)}$, which is uniform. The analysis for expectation under $\mathcal{T}^{(S_0)}$ is identical as it only involves correlation. Therefore

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}}[fF'] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}[fF']\right| \le \left|\mathop{\mathbf{E}}_{\mathcal{T}}\left[F' \sum_{\substack{T\subseteq L^{k-1}\\ |T|\le J/2}} f_T\right] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}\left[F' \sum_{\substack{T\subseteq L^{k-1}\\ |T|\le J/2}} f_T\right]\right| + 2\varepsilon_1 . \quad (5.9)$$

Now we turn to the low degree parts. Further unraveling the Efron-Stein decomposition, we have

$$F' \sum_{\substack{T\subseteq L^{k-1}\\ |T|\le J/2}} f_T \quad (5.10)$$

$$= F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U \quad (5.11)$$

$$= F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|=|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U + F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|>|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U . \quad (5.12)$$

Following the terminology in [54, 104], we refer to the first term as *shattered term*, and the second as *non-shattered term*. We study these two terms separately. From (5.9), we have

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}}[fF'] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}[fF']\right| \quad (5.13)$$

$$\le 2\varepsilon_1 + \left|\mathop{\mathbf{E}}_{\mathcal{T}}\left[F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|=|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U\right] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}\left[F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|=|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U\right]\right| \quad (5.14)$$

$$+ \left|\mathop{\mathbf{E}}_{\mathcal{T}}\left[F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|>|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U\right] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}\left[F' \sum_{\substack{U\subseteq L^{k-m}\times R^{m-1}\\ |U|>|\pi(U)|\\ |\pi(U)|\le J/2}} \hat{f}_U \chi_U\right]\right| . \quad (5.15)$$

We first use smoothness to bound the non-shattered terms. The process is very

similar to that in [104], and we get

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| > |\pi(U)| \\ |\pi(U)| \leq J/2}} \hat{f}_U \chi_U \right] \right| \leq 2\varepsilon_1 \,. \tag{5.16}$$

The same argument holds under distribution $\mathcal{T}^{(S_0)}$. For the difference involving the shattered terms, we have

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| = |\pi(U)| \\ |\pi(U)| \leq J/2}} \hat{f}_U \chi_U \right] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| = |\pi(U)| \\ |\pi(U)| \leq J/2}} \hat{f}_U \chi_U \right] \right| \tag{5.17}$$

$$= \left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| = |\pi(U)| \\ |\pi(U)| \leq J/2}} \hat{f}_U \chi_U \right] - \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| = |\pi(U)| \\ |\pi(U)| \leq J/2}} (1-\gamma)^{|U|} \hat{f}_U \chi_U \right] \right| \tag{5.18}$$

$$= \left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[F' \sum_{\substack{U \subseteq L^{k-m} \times R^{m-1} \\ |U| = |\pi(U)| \leq J/2}} (1-(1-\gamma)^{|U|}) \hat{f}_U \chi_U \right] \right| \tag{5.19}$$

$$\leq 1 - (1-\gamma)^{J/2} \leq \varepsilon_1 \,. \tag{5.20}$$

The key step is (5.18) where we switch the distribution of the second term from $\mathcal{T}^{(S_0)}$ to $\mathcal{T}$. We rely crucially on the fact that $|U| = |\pi(U)|$. To see why this holds, denote the query to f as x (just for the current argument). Observe that the variables x_t are independent for $t \in U$ with different $\pi(t)$, so we first focus on those values t that map to the same $r \in U$. Looking at each $r \in \pi(U)$, $|\pi(U)| = |U|$ implies that there is a unique $t \in U$ such that $\pi(t) = r$, and thus perturbing those x_t satisfying $\pi(t) = r$ with probability γ would give exactly a multiplicative factor of $(1-\gamma)$ to the expectation. Since each $r \in \pi(U)$ contributes a factor of $(1-\gamma)$, the final factor thus becomes $(1-\gamma)^{|\pi(U)|} = (1-\gamma)^{|U|}$.

Summing up the above, we have

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}}[fF'] - \mathop{\mathbf{E}}_{\mathcal{T}^{(S_0)}}[fF'] \right| \leq 7\varepsilon_1 \,. \tag{5.21}$$

This completes the proof.

5.3.3 Influence based decoding

Suppose we have that for some $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_3$, the following term is large

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] \right| > \varepsilon_1 \,, \tag{5.22}$$

then for at least an $\varepsilon_1/2$ fraction of all possible edge samplings, we have

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] \right| > \varepsilon_1/2 \,. \tag{5.23}$$

In the rest of the proof, we focus on samplings of edges where (5.23) holds. We show how to extract a labeling for these edges.

Observe that after we fixed the edges, which function we query only depends on the layer of the query, so for the rest of this section, let f_l be the function on layer l. Also recall that $m(S) = \max S$ is the layer query $x^{(S)}$ is in, and thus in the PCP query $x^{(S)}$ goes to function $f_{m(S)}$. Let $l_m = \max_{S \in \mathcal{S}} m(S)$ be the maximum layer among queries that appears in $\mathcal{S}$.

For $l \in [k]$, denote the queries that appear on layer l as $\mathcal{L}_l := \{S \in \mathcal{S}_1 \cup \mathcal{S}_3 \mid \max S = l\}$, and let $\mathcal{L}_{\leq l} := \cup_{l' \leq l} \mathcal{L}_{l'}$, and similarly define $\mathcal{L}_{<l}$. We need the following observation on independence between queries.

Claim 5.14. *For any $l \in [k]$ and $S_0 \in \mathcal{L}_l$, $x^{(S_0)}$ and $\{x^{(S)}\}_{S \in \mathcal{L}_{<l}}$ are independent under both $\mathcal{T}$ and $\mathcal{T}'$.*

Proof. We first consider $\mathcal{T}$. We can write $x^{(S_0)} = x_e \cdot x^{(\{l\})}$, where $x^{(\{l\})}$ is a uniform random string, "$\cdot$" denotes the elementwise product, and x_e depends on: (1) S_0, (2) $\{x^{(S)}\}_{S \in \mathcal{L}_{<l}}$, (3) the choice of the locations N_r for $r \in L^{k-1}$, and (4) the decision whether the bits in query $x^{(N_r)}$ are resampled. Observe that $x^{(\{l\})}$ is independent of $\{x^{(S)}\}_{S \in \mathcal{L}_{<l}}$, the N_r, and whether the bits are resampled, thus its marginal is still uniform no matter how we fix everything else, and so is the marginal of $x^{(S_0)}$. This implies that $x^{(S_0)}$ is independent of everything else and in particular $\{x^{(S)}\}_{S \in \mathcal{L}_{<l}}$.

For $\mathcal{T}'$, note that the additional noise is applied independently to each bit, and we can use a similar argument as above to show that the marginal of $x^{(S_0)}$ is always uniform, no matter how we fix the other parameters. □

We rewrite the left hand side of (5.23) as

$$\mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f_{m(S)}(x^{(S)}) \right] = \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{l \in [k]} \prod_{S \in \mathcal{L}_l \cap \mathcal{S}} f_l(x^{(S)}) \right] .$$

By our choice of permutation and Lemma 5.10, there exists l_0 and j_0 such that

$$\left| \left\{ S \in \mathcal{L}_{l_0} \cap \mathcal{S} \mid S \ni j_0 \right\} \right| \text{ is odd} \,.$$

Then flipping $x^{(\{j_0\})}$ while leaving all other $x^{(\{j'\})}$ unchanged changes the sign of the following

$$\prod_{S\in\mathcal{L}_{l_0}\cap\mathcal{S}} f_{l_0}(x^{(S)})\,,$$

and since the marginal of $x^{(\{j_0\})}$ is uniform and all functions are folded, we have

$$\mathop{\mathbf{E}}_{\mathcal{T}'}\left[\prod_{S\in\mathcal{L}_{l_0}\cap\mathcal{S}} f_{l_0}(x^{(S)})\right] = 0\,.$$

To complete the proof of soundness, we show that if

$$\begin{aligned}\left|\mathop{\mathbf{E}}_{\mathcal{T}'}\left[\prod_{S\in\mathcal{S}} f_{m(S)}(x^{(S)})\right]\right| &= \left|\mathop{\mathbf{E}}_{\mathcal{T}'}\left[\prod_{l\in[k]}\prod_{S\in\mathcal{L}_l\cap\mathcal{S}} f_l(x^{(S)})\right]\right| \\ = \left|\mathop{\mathbf{E}}_{\mathcal{T}'}\left[\prod_{l\in[k]}\prod_{S\in\mathcal{L}_l\cap\mathcal{S}} f_l(x^{(S)})\right] - \prod_{l\in[k]}\mathop{\mathbf{E}}_{\mathcal{T}'}\left[\prod_{S\in\mathcal{L}_l\cap\mathcal{S}} f_l(x^{(S)})\right]\right| &> \varepsilon_1/2\,, \qquad (5.24)\end{aligned}$$

then there exists two layers $1 \le l < l_m \le k$ such that

$$\sum_{\substack{r_l\in L^{k-l}\times R^{l-1}\\ r_m\in L^{k-l_m}\times R^{l_m-1}\\ \pi_{l_m\to 1}(r_m)=\pi_{l\to 1}(r_l)}} \mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l)\,\mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m}) > \frac{\varepsilon_1^2}{4Z}\,, \qquad (5.25)$$

where $Z = Z(k,\gamma) := 2^{4k^3}k^9\gamma^{-1}$. This enables us to define a good labeling as the following: choose r_l with probability $\mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l)/\,\mathrm{Inf}^{(1-\gamma)}(f_l)$, and similarly choose r_m with probability $\mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m})/\,\mathrm{Inf}^{(1-\gamma)}(f_{l_m})$, then the probability that the labeling weakly satisfies the edge is

$$\begin{aligned}&\sum_{\substack{r_l\in L^{k-l}\times R^{l-1}\\ r_m\in L^{k-l_m}\times R^{l_m-1}\\ \pi_{l_m\to 1}(r_m)=\pi_{l\to 1}(r_l)}} \frac{\mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l)}{\mathrm{Inf}^{(1-\gamma)}(f_l)}\frac{\mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m})}{\mathrm{Inf}^{(1-\gamma)}(f_{l_m})} \\ > \gamma^2 &\sum_{\substack{r_l\in L^{k-l}\times R^{l-1}\\ r_m\in L^{k-l_m}\times R^{l_m-1}\\ \pi_{l_m\to 1}(r_m)=\pi_{l\to 1}(r_l)}} \mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l)\,\mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m}) \;\ge\; \frac{\gamma^2\varepsilon_1^2}{4Z}\,.\end{aligned}$$

This holds for at least $\varepsilon_1/2$ fraction of choices of edges, thus the expected value achieved by the above random labeling procedure is at least $\gamma^2\varepsilon_1^3/8Z$, a value depending only on k and ε.

The key step to proving (5.24) is to bound the following difference

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f_{m(S)}(x^{(S)}) \right] - \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{l<l_m} \prod_{S \in \mathcal{L}_l \cap \mathcal{S}} f_l(x^{(S)}) \right] \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{L}_{l_m} \cap \mathcal{S}} f_{l_m}(x^{(S)}) \right] \right|, \tag{5.26}$$

where we recall that l_m is the highest layer of queries involved in $\mathcal{S}$. We can iteratively apply the bound on (5.26) to get (5.24). In order to establish (5.26), we use an invariance-type result from [104].

Theorem 5.15 ([104])**.** *Consider functions*

$$\left\{ f^{(t)} \in L^\infty(\Omega_t^n) \right\}_{t \in [d]} \qquad \text{on a probability space} \qquad \mathcal{P} = \left(\prod_{t=1}^{d} \Omega_t, P \right)^{\otimes n}$$

and a set $M \subsetneq [d]$. Furthermore, let $\mathcal{C}$ be the collection of minimal sets $C \subseteq [d]$, $C \nsubseteq M$, such that the spaces $\{\Omega_t\}_{t \in C}$ are dependent. Then

$$\begin{aligned} &\left| \mathbf{E}\left[\prod f^{(t)} \right] - \prod_{t \notin M} \mathbf{E}[f^{(t)}] \, \mathbf{E}\left[\prod_{t \in M} f^{(t)} \right] \right| \\ &\leq \; 2^{2d} \max_{C \in \mathcal{C}} \left(\sqrt{\min_{r \in C} \mathrm{Inf}(f^{(r)}) \sum_i \prod_{t \in C - \{r\}} \mathrm{Inf}_i(f^{(t)})} \prod_{t \notin C} \left\| f^{(t)} \right\|_\infty \right). \end{aligned}$$

To apply the above theorem, we first combine all functions that are not in the highest layer. Let

$$Q = \prod_{S \in \mathcal{S} \cap \mathcal{L}_{<l_m}} X^{(S)},$$

and $q \in Q$ simply be concatenation of $\{x^{(S)}\}_{S \in \mathcal{S} \cap \mathcal{L}_{<l_m}}$. Define the combined function

$$F = \prod_{S \in \mathcal{S} \cap \mathcal{L}_{<l_m}} f_{m(S)},$$

and the noisy version

$$F' = \prod_{S \in \mathcal{S} \cap \mathcal{L}_{<l_m}} T_{1-\gamma} f_{m(S)}.$$

We still have by Claim 5.14 that Q and $X^{(S_0)}$ are independent for all $S_0 \in \mathcal{L}_{l_m}$. Then the first term in (5.26) becomes

$$\mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f_{m(S)}(x^{(S)}) \right] = \mathop{\mathbf{E}}_{\mathcal{T}} \left[F'(q) \prod_{S \in \mathcal{S} \cap \mathcal{L}_{l_m}} T_{1-\gamma} f_{l_m}(x^{(S)}) \right].$$

Let us set $M = \mathcal{S} \cap \mathcal{L}_{l_m}$. Consider the sets C in Theorem 5.15. Since Theorem 5.15 requires that $C \nsubseteq M$, we have that C must include variable q. Due to the

independence in Claim 5.14, C must also include at least two variables from $\mathcal{S}\cap\mathcal{L}_{l_m}$. Applying Theorem 5.15, we have

$$\left|\mathop{\mathbf{E}}_{\mathcal{T}}\left[F'(q)\prod_{S\in\mathcal{S}\cap\mathcal{L}_{l_m}}T_{1-\gamma}f_{l_m}(x^{(S)})\right]-\mathop{\mathbf{E}}_{\mathcal{T}}\left[F'(q)\right]\mathop{\mathbf{E}}_{\mathcal{T}}\left[\prod_{S\in\mathcal{S}\cap\mathcal{L}_{l_m}}T_{1-\gamma}f_{l_m}(x^{(S)})\right]\right|$$
$$\leq 2^{2k^3}\sqrt{\mathrm{Inf}(\overline{T_{1-\gamma}f_{l_m}})\sum_{r\in L^{k-1}}\mathrm{Inf}_r(\overline{F'})\,\mathrm{Inf}_r(\overline{T_{1-\gamma}f_{l_m}})}\,,$$

where $\overline{F'}$ and $\overline{f}_{l_m}$ are lifted versions of F' and f_{l_m} as defined in Definition 2.32.

Using Proposition 2.33, we have

$$\mathrm{Inf}(\overline{T_{1-\gamma}f_{l_m}})\leq\mathrm{Inf}(T_{1-\gamma}f_{l_m})=\mathrm{Inf}^{(1-\gamma)}(f_{l_m})\leq\gamma^{-1}\,,$$

and similarly

$$\begin{aligned}\mathrm{Inf}_r(\overline{T_{1-\gamma}f_{l_m}}) &\leq \sum_{\substack{r_m\in L^{k-l_m}\times R^{l_m-1}\\ \pi_{l_m\to1}(r_m)=r}}\mathrm{Inf}_{r_m}(T_{1-\gamma}f_{l_m})\\ &= \sum_{\substack{r_m\in L^{k-l_m}\times R^{l_m-1}\\ \pi_{l_m\to1}(r_m)=r}}\mathrm{Inf}^{(1-\gamma)}_{r_m}(f_{l_m})\,.\end{aligned}$$

Now we need to relate $\mathrm{Inf}_r(\overline{F'})$ with $\mathrm{Inf}^{(1-\gamma)}_r(f_{m(S)})$. We use the following generalization of Lemma 6.5 from [86].

Lemma 5.16 ([86])**.** *Let $(\prod_{i=1}^m\Omega_i^n,\mu)$ be correlated probability space, and $f_i:\Omega_i^n\to[-1,1]$ for $i=1,\ldots,m$. Then for all r:*

$$\mathrm{Inf}_r\left(\prod_{i=1}^m f_i\right)\leq m\sum_{i=1}^m\mathrm{Inf}_r(f_i)\,.$$

The argument goes exactly the same so we omit the proof here.

Applying Lemma 5.16, we can upper-bound $\mathrm{Inf}_r(\overline{F'})$ by the following

$$\begin{aligned}\mathrm{Inf}_r(\overline{F'}) &\leq k^3\sum_{S\in\mathcal{S}\cap\mathcal{L}_{<l_m}}\mathrm{Inf}_r(\overline{T_{1-\gamma}f_{m(S)}})\leq k^6\sum_{l<l_m}\mathrm{Inf}_r(\overline{T_{1-\gamma}f_l})\\ &\leq k^6\sum_{l<l_m}\sum_{\substack{r_l\in L^{k-l}\times R^{l-1}\\ \pi_{l\to1}(r_l)=r}}\mathrm{Inf}^{(1-\gamma)}_{r_l}(f_l)\,,\end{aligned}$$

where we used Proposition 2.33 to obtain the last inequality.

Summing up, we have

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}} \left[T_{1-\gamma} F(q) \prod_{S \in \mathcal{S} \cap \mathcal{L}_{l_m}} T_{1-\gamma} f_{l_m}(x^{(S)}) \right] - \right.$$

$$\left. \mathop{\mathbf{E}}_{\mathcal{T}} \left[T_{1-\gamma} F(q) \right] \mathop{\mathbf{E}}_{\mathcal{T}} \left[\prod_{S \in \mathcal{S} \cap \mathcal{L}_{l_m}} T_{1-\gamma} f_{l_m}(x^{(S)}) \right] \right| \tag{5.27}$$

$$\leq 2^{2k^3} \sqrt{k^6 \gamma^{-1} \sum_{\substack{1 \leq l < l_m \\ r_l \in L^{k-l} \times R^{l-1} \\ r_m \in L^{k-l_m} \times R^{l_m - 1} \\ \pi_{l_m \to 1}(r_m) = \pi_{l \to 1}(r_l)}} \mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l) \, \mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m})} \,. \tag{5.28}$$

Let $Z' = 2^{2k^3} \sqrt{k^6 \gamma^{-1}}$, applying (5.28) to all layers, we get

$$\left| \mathop{\mathbf{E}}_{\mathcal{T}'} \left[\prod_{S \in \mathcal{S}} f^{(S)}(x^{(S)}) \right] \right| \; < \; Z' \sum_{2 \leq l_m < k} \sqrt{\sum_{\substack{1 \leq l < l_m \\ r_l \in L^{k-l} \times R^{l-1} \\ r_m \in L^{k-l_m} \times R^{l_m - 1} \\ \pi_{l_m \to l}(r_m) = r_l}} \mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l) \, \mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m})} \,.$$

Thus if the left hand side of the above is larger than $\varepsilon_1/2$, then there exists $1 \leq l < l_m \leq k$ such that

$$\sum_{\substack{r_l \in L^{k-l} \times R^{l-1} \\ r_m \in L^{k-l_m} \times R^{l_m - 1} \\ \pi_{l_m \to l}(r_m) = r_l}} \mathrm{Inf}_{r_l}^{(1-\gamma)}(f_l) \, \mathrm{Inf}_{r_m}^{(1-\gamma)}(f_{l_m}) > \left(\frac{\varepsilon_1/2}{kZ'} \right)^2 \cdot \frac{1}{k} = \frac{\varepsilon_1^2}{4Z} \,.$$

This completes the proof. □

Part III

Graph and Hypergraph Coloring

五色令人目盲
五音令人耳聋
五味令人口爽
驰骋畋猎　令人心发狂
难得之货　令人行妨

—— 老子 · 道德经

Chapter 6

An Introduction to Coloring

We now turn to approximating the chromatic number of graphs and hypergraphs. In this chapter, we review what was known about the complexity of coloring for graphs and hypergraphs and their relation to Gap-CSP. In Chapter 7, we use the recent breakthrough result by [24] as discussed in Section 5.1 to get an exponential improvement in hardness of approximating chromatic number of graphs. Chapter 8 is about inapproximability of hypergraph coloring, where we use some variant of Gap-CSP as a starting point of our reduction.

An undirected graph $G = (V, E)$ consists of vertex set V and edge set $E \subseteq \binom{V}{2}$. For some integer $k \geq 2$, a k-uniform hypergraph $H = (V, F)$ consists of vertex set V and edge set $F \subseteq \binom{V}{k}$. Note that 2-uniform hypergraphs are simply ordinary graphs.

Recall the following definition of independent sets.

Definition 6.1. *For a graph $G = (V, E)$, a set of vertices $S \subseteq V$ is an* independent set *if it does not contain any edge, that is, for all $e \in E$, $e \nsubseteq S$.*

More generally, for a k-uniform hypergraph $H = (V, F)$, a set $S \subseteq V$ is an independent set *if for all $f \in E$, $f \nsubseteq S$.*

We use $\alpha(\cdot)$ to denote the fractional size of the maximum cardinality independent set of a graph or hypergraph, also known as the fractional independence number*.*

Let $q \in \mathbb{N}^+$, and we define q-coloring as follows.

Definition 6.2. *For a given graph $G = (V, E)$ (or k-uniform hypergraph $H = (V, F)$), a* q-coloring *is a function $\sigma : V \to [q]$, such that for any $c \in [q]$, the set of vertices that are colored with color c, denoted as $\sigma^{-1}(c) \subseteq V$, is an independent set.*

The chromatic number *$\chi(\cdot)$ of a graph G or hypergraph H is the minimum q such that a q-coloring exists for G or H.*

Chromatic number and fractional independence number are closely related.

Fact 6.3. *For any k-uniform hypergraph H, we have $\chi(H)\alpha(H) \geq 1$.*

Coloring a graph or a hypergraph using few colors is a classical combinatorial optimization problem, and is one of the most well-studied problems in theoretical computer science. It is also closely related to other problems such as finding maximum independent sets, PCPs with certain special properties, and also Gap-CSP. In the following two sections, we present the algorithms and hardness results known each of them.

6.1 Complexity of Graph Coloring

For a given graph G, deciding whether it has a 2-coloring is the same as deciding whether it is a bipartite graph, and can be easily solved in polynomial time. In general, however, determining the chromatic number of a graph exactly is NP-hard [39]. However, in many applications, it suffices to find a good enough approximation. In other words, given a q-colorable graph, we would like to color it with as few colors as possible. As in Section 3.1.1, this optimization problem has a natural variant of promise problem: given graph G, decide whether $\chi(G) \leq q$ or $\chi(G) > q'$. Another closely related variant is the promise problem where we need to decide whether $\chi(G) \leq q$ or $\alpha(G) < 1/q'$. Note that by Fact 6.3, if $\alpha(G) < 1/q'$, then $\chi(G) > q'$. Therefore if the problem with independent set promise is hard, then so is the promise chromatic number problem.

For $q = 3$, it is known that coloring 3-colorable graphs with 4 colors is NP-hard, and for general q-colorable graphs it is NP-hard to color with $q + 2\lfloor q/3 \rfloor - 1$ colors [66, 45]. For sufficiently large q, a result by Khot [67] showed that it is NP-hard to color a q-colorable graph with $q^{\frac{1}{25}\log q}$ colors. This was later improved to $2^{\Omega(q^{1/3})}$ by Huang [61], and we present this result in Chapter 7.

Assuming a variant of Khot's 2-to-1 Conjecture, Dinur, Mossel and Regev [32] proved that it is NP-hard to q'-color a q-colorable graph for any $3 \leq q < q'$. The dependency between the hardness of graph coloring and the parameters of 2-to-1 Label-Cover was made explicit and improved by Dinur and Shinkar [34], who showed that it is NP-hard to $\log^c n$-color a 4-colorable graph for some constant $c > 0$ assuming the 2-to-1 Conjecture. Guruswami and Sinop [47] proved that assuming the 2-to-1 Conjecture, it is hard to find an independent set with more than $O\left(\frac{n}{\Delta^{1-c/(k-1)}}\right)$ vertices in a k-colorable graph of maximum degree Δ for some absolute constant $c \leq 4$.

There have been many works on approximation algorithms as well. Wigderson [105] gave an algorithm using $O(n^{1-1/(q-1)})$ colors. This was improved by Berger and Rompel [18] to $O\left((n/\log n)^{1-1/(q-1)}\right)$ colors. Karger, Motwani and Sudan [63] used semi-definite programming to achieve $\tilde{O}(n^{1-3/(q+1)})$, which was adapted in Blum and Karger [20] to an algorithm that colors a 3-colorable graph with $\tilde{O}(n^{3/14})$ colors. For 3-colorable graphs, the best algorithm is by Kawarabayashi and Thorup [65] which uses $O(n^{0.19996})$ colors, based on results by Arora and Chlamtáč [4], Chlamtáč [27], and the earlier work of Kawarabayashi and Thorup [64].

As we can see, there is still a huge gap between the best approximation guarantee and the best hardness result.

6.2 Complexity of Hypergraph Coloring

Our understanding of hypergraph coloring is much better. For $k \geq 3$, even determining whether a k-uniform hypergraph has a 2-coloring is NP-hard.

In terms of approximation algorithms, the best algorithm for 2-colorable 3-uniform hypergraphs still requires $n^{\Omega(1)}$ colors [82, 1, 26].

From the hardness side, the first super-constant hardness result was proved in [44]. They proved that for 4-uniform 2-colorable hypergraphs, finding a coloring with any constant number of colors is NP-hard, and finding a coloring with $O(\log\log n/\log\log\log n)$ colors is quasi-NP-hard. For 3-uniform 2-colorable hypergraphs, a similar constant gap NP-hardness was proved in [33]. Khot [68] proved that coloring 3-colorable 3-uniform hypergraphs with any constant number of colors is hard, and for q-colorable 4-uniform hypergraphs, coloring with $\log^{\Omega(q)} n$ colors is quasi-NP-hard for $q \geq 7$. The analysis in [44] was improved by Holmerin, who proved that even finding an independent set of fractional size $\Omega(\log\log\log n/\log\log n)$ is quasi-NP-hard [58]. The construction was further improved recently by Saket [95], where it was shown that it is quasi-NP-hard to find independent set of size $n/\log^{\Omega(1)} n$ in 2-colorable 4-uniform hypergraphs [95]. There has also been work on the hardness of finding independent sets in almost 2-colorable hypergraphs — hypergraphs that becomes 2-colorable after removing a small fraction of vertices. Much stronger result is known, albeit at the cost of imperfect completeness. We refer to [78] for more details.

Recently, the first super-polylogarithmic hardness result was proved in [42], showing hardness for coloring 2-colorable 8-uniform hypergraphs with $2^{2^{\Omega(\sqrt{\log\log n})}}$ colors, using Low-Degree-Long-Code proposed in [13]. The analyses of Low-Degree-Long-Code in [42] employ techniques for testing Reed-Muller codes developed in [29], which in turn uses applies tools for Reed-Muller code testing in the work of Bhattacharyya, Kopparty, Schoenebeck, Sudan and Zuckerman[19].

Using a very different approach, Khot and Saket gave another exponential improvement in [77], showing a quasi-NP-hardness for coloring 2-colorable 12-uniform hypergraphs with $\exp(\log^{\Omega(1)} n)$ colors. The analysis was simplified by Varma in [103] using ideas from [42]. The work in Chapter 8 is based on these recent developments, and we elaborate more on these results there.

Chapter 7

Hardness of Approximating Chromatic Number

The result we present in this chapter is based on, and improves the hardness by Khot [67], where he proved that for sufficiently large q, coloring a q-colorable graph with $q^{\frac{1}{25}\log q}$ colors is NP-hard. Khot's hardness result can be derived using the GAP-CSP results either from Håstad and Khot [55] or Samorodnitsky and Trevisan [96]. We can view both [55] and [96] as showing approximation resistance for a family of Boolean predicates that have very few accepting inputs. For a more extensive discussion about results regarding hardness of GAP-CSP, difference between $\text{GAP}_{1,s}$-CSP and $\text{GAP}_{1-\varepsilon,s}$-CSP, we refer to Chapter 5. It is noted in Khot [67] that having perfect completeness is not necessary but makes the reduction for coloring easier.

There is a canonical reduction that converts a $\text{GAP}_{1,s}$-k-CSP problem (or $\text{GAP}_{1-\varepsilon,s}$-$k$-CSP) into a promise independent set problem where one is asked to decide whether a graph has an independent set of fractional size at least roughly $1/k$ or that it has no independent set of fractional size larger than k/s. This does not immediately give a hardness for approximating chromatic number, but it serves as the basis for the reduction in [67], using a GAP-k-CSP hardness result from [55].

In Chapter 5, we discussed some recent improvements of GAP-k-CSP hardness, and it is a natural question to ask whether these improvements lead to better inapproximability result for approximating chromatic number. In particular, the HADAMARD_K predicate, proved to be approximation resistance in [24], has density $\Theta(K)/2^K$, much lower than the predicate used in [55], which has density $2^{O(K^{1/2})}/2^K$.

In [24], Chan applied his GAP-k-CSP hardness result and showed that for any $K \geq 3$, there is $\nu = o(1)$ such that given a graph with an induced K-colorable subgraph of fractional size $1-\nu$, it is NP-hard to find an independent set of fractional size $1/2^{K/2} + \nu$. Although this gives a larger gap than Khot [67], the result lacks "perfect completeness" and thus is not comparable with Khot [67]. We refer to

[31, 76, 24] for additional discussions on AlmostColoring.

We show improved hardness of approximating chromatic number using the Chan's construction.

Theorem 7.1. *For all sufficiently large K, it is* NP*-hard to color a K-colorable graph with $2^{\Omega(K^{1/3})}$ colors. Moreover, this hardness result holds for graphs that have degree bounded by $K2^{O(K^{1/3})}$.*

Stated in terms of degree, Theorem 7.1 says that there exists some constant c, such that for all large enough Δ, it is NP-hard to color a $(\log \Delta)^3$-colorable graph of maximum degree bounded by Δ with $O(\Delta^c)$ colors.

Our approach follows that of Khot [67]. The main issue is that Khot's technique is for Long-Code-based reductions, and we need to work on adapting his technique so that it works with the new construction by Chan [24], which gives much better dependency between soundness and the arity of Gap-CSP. This is also the main source of the improvement in Theorem 7.1. This reduction alone will give us graphs with degree at least doubly exponential in K. To get a tighter dependency on degree, we apply a technique in Trevisan [102] to sparsify the output of the reduction.

7.1 Main Theorem

In this section, we prove Theorem 7.1 — for sufficiently large K, it is NP-hard to color a K-colorable graph with less than $2^{\Omega(K^{1/3})}$ colors. For convenience of notation, we in fact prove a gap of $O(K^3)$ versus $2^{\Omega(K)}$.

The overall approach follows that in Khot [67]. We start by describing the FGLSS graph [37] of Chan's PCP as summarized in Theorem 5.4, with the following parameters: let $\varepsilon > 0$ be some small constant, $\delta = \varepsilon \cdot 2^{-K}$, and $\eta = \varepsilon/K^2$. By Theorem 5.4, we require the soundness of Label Cover to be $\sigma = (\delta/\text{poly}(K/\eta))^{O(1)} = 2^{-\Omega(K)}$. This means that the size of the label $L = \text{poly}(1/\sigma) = \exp(\Theta(K))$.

The vertices in the FGLSS graph are function queries and their corresponding accepting configurations, denoted as $(\mathbf{f_v}, \mathbf{q}, \mathbf{z})$. The weight of the vertex is the probability that query $(\mathbf{f_v}, \mathbf{q})$ is picked. The total weight of the graph is therefore $K + 1$, the number of accepting assignments of Hadamard$_K$. Two vertices are connected if they are clearly inconsistent (returning different answers for the same query to the same function). An independent set in the graph corresponds to a strategy / set of functions, and its weight is the acceptance probability of such strategy. Note that if the maximum weight independent set of the FGLSS graph has weight w, then we need at least $(K+1)/w$ colors to color the whole graph since vertices having the same color must form an independent set.

To use the FGLSS graph for coloring results, we also need to show that if a PCP has acceptance probability $1-\varepsilon$, we can color the FGLSS graph with a small number of colors. Note that in this case, we know that there is an independent set of weight $1-\varepsilon$ in the FGLSS graph, corresponding to a correct proof. Khot's

idea in [67] is to modify the definition of the PCP so that the correct proofs are parametrized by some global parameter $\alpha \in \{0,1\}^t$. This gives us 2^t different correct proofs and thus 2^t independent sets of weight $1-\varepsilon$, and by choosing the right t, we expect those independent sets cover most of the FGLSS graph of the modified PCP and thus gives a coloring of at most 2^t colors.

Formally, we modify the construction in Section 5.1 so that the functions in the proof become $\mathbf{f}_{\mathbf{v}_i} : \left(\{-1,1\}^{R\cdot 2^t}\right)^{i-1} \times \{-1,1\}^{L\cdot 2^t} \times \left(\{-1,1\}^{R\cdot 2^t}\right)^{K-i} \to \{-1,1\}$. Alternatively, we can think of this as modifying Label-Cover by appending a t-bit binary string to all the labels and defining the new projection in the Label-Cover instance as $\pi'_e(r \circ \alpha) = \pi_e(r) \circ \alpha$ for $r \in R$ and $\alpha \in \{0,1\}^t$, where "$\circ$" denotes string concatenation. The value of this new Label-Cover instance is exactly the same as the original setting. Consider the FGLSS graph in this new setting. Soundness is straightforward. If the new proof makes the verifier accept with probability at least $(K+1)/2^K + 2\delta$, then the value of the new Label-Cover is at least σ and hence the original instance also has value at least σ.

Now let us consider the case of completeness. If the original Label-Cover instance has value 1, then extending a valid labeling with any $\alpha \in \{0,1\}^t$ gives us a valid labeling for the modified instance, which corresponds to an independent set of weight at least $1-\varepsilon$ in the modified FGLSS graph. We need to show that the 2^t independent sets corresponding to different $\alpha \in \{0,1\}^t$ cover almost all of the FGLSS graph of the modified PCP. In fact, we can efficiently identify a small subset of the vertices that contains all vertices that are not covered by any independent sets of the above form and remove them from the FGLSS graph.

To this end, we follow Khot's notation and introduce the following definition characterizing whether we can cover certain vertex with independent sets.

Definition 7.2 (Good Queries)**.** *Let $\mathbf{l} = \{(l_i, r_i)\}_{i=1}^K$ be any K pairs of labelings, where $l_i \in [L]$, $r_i \in [R]$ for all $i \in [K]$. Define the i-th mixed labeling*

$$\mathbf{m}_i(\mathbf{l}) = (r_1, \ldots, r_{i-1}, l_i, r_{i+1}, \ldots, r_K).$$

Let $\mathbf{f}_{i,\mathbf{l}}$ be the product of Long-Code *encodings of the labelings in $\mathbf{m}_i$. Denote by $\mathbf{l}^\alpha := \{(l_i \circ \alpha, r_i \circ \alpha)\}_{i=1}^K$ the labelings extended by α. Define $\mathbf{f}^\alpha_{i,\mathbf{l}}$ similarly.*

A set of queries $\mathbf{q} = (q_1, \ldots, q_K)$ is good *if for any K tuples of labelings $\mathbf{l}$ and any accepting assignment $\mathbf{z} = (z_1, \ldots, z_K)$ of* Hadamard$_K$*, there exists a global extension α, such that $\mathbf{f}^\alpha_{i,\mathbf{l}}(q_i) = z_i$ for all $i \in [K]$.*

Remark. *Given that the vertices in the FGLSS graph correspond to combinations of queries and answers to the queries, it might seem odd that we are defining the notion of* good *for queries rather than queries together with answer to the queries. This is mostly for the purpose of making Lemma 7.3 easier to prove. Moreover, we lose at most a factor $\Theta(K)$ in the soundness from this, which is negligible since the soundness we are aiming for is $\exp(-\Theta(K))$.*

To verify if a set of queries is good, we only need to check all K tuples of labelings and all accepting assignments of the Hadamard predicate Hadamard$_K$.

Those are all constants depending only on K (and ε). The following lemma shows that the fraction of bad queries is small.

Lemma 7.3. *Let t be such that $2^t = C \cdot K^3$ for some large constant C. For large enough K, at most a weighted fraction of $\exp(-\Theta(K))$ of the queries is not good.*

Before proving the lemma, let us see how it leads to our main theorem.

Remove the vertices in the FGLSS graph that correspond to queries that are not good. By Lemma 7.3, the fraction of vertices removed is bounded by $\exp(-O(K))$. In the soundness case coloring the FGLSS graph still needs at least $(K+1)(1-\exp(-\Theta(K)))/2^{-K} = 2^{\Omega(K)}$ colors. In the completeness case, the Label-Cover instance has value 1. Fix a labeling that satisfies all the edges. For a vertex $(\mathbf{f_v}, \mathbf{q}, \mathbf{x})$ in the modified FGLSS graph, let $\mathbf{l_v}$ be the set of K tuples of labelings of the sampled vertices. Each $\alpha \in \{0,1\}^t$ is associated with an independent set consisting of vertices of the form $(\mathbf{f_v}, \mathbf{q}, \mathbf{z})$, where $z_i = \mathbf{f}^{\alpha}_{i,\mathbf{l_v}}(q_i)$ for all $i \in [K]$.

Consider any vertex $(\mathbf{f_v}, \mathbf{q}, \mathbf{x})$ in the modified FGLSS graph. We know that $\mathbf{q}$ is good so by definition there exists $\alpha_0 \in \{0,1\}^t$ such that $\mathbf{f}^{\alpha_0}_{i,\mathbf{l_v}}(q_i) = x_i$ for all $i \in [K]$. Hence, it is covered by the independent set associated with α_0. Therefore the modified FGLSS graph can be colored with $2^t = O(K^3)$ colors.

Proof of Lemma 7.3. For query $\mathbf{q}$, let $Q(\mathbf{q})$ be the event that $\mathbf{q}$ is not good in the sense of Definition 7.2: there exists some labeling $\mathbf{l}$ and some accepting assignment $\mathbf{z}$, such that for any α, there exists $i \in [K]$, $\mathbf{f}^{\alpha}_{i,\mathbf{l}}(q_i) \neq z_i$. It suffices to bound $\Pr_{\mathbf{q}}[Q(\mathbf{q})]$.

Fix some K tuples of labeling $\mathbf{l}$ of the label cover instance and some accepting assignment $\mathbf{z}$. Consider $\alpha \in \{0,1\}^t$. Over the queries sampled, the probability that $\mathbf{f}^{\alpha}_{i,\mathbf{l}}(q_i) = z_i$ for all $i \in [K]$ is $1/(K+1)$ before adding noise. To estimate the effect of noise, note that there are K functions, each being a product of K long codes, therefore the answers $\{\mathbf{f}^{\alpha}_{i,\mathbf{l}}(q_i)\}_{i\in[K]}$ depends on K^2 bits. If none of these K^2 bits are corrupted, then the answer is exactly $\mathbf{z}$. This gives an overall probability of $\Theta(1/K \cdot (1-\eta)^{K^2}) = \Theta(e^{-\eta K^2}/K) = \Theta(1/K)$. The contribution of probability from other sources is negligible.

Note that for different extension α, the bits that $\mathbf{f}^{\alpha}_{i,\mathbf{l}}$ reads from $\mathbf{q}$ are completely independent, so we have

$$\Pr_{\mathbf{q}}\left[\forall \alpha, \exists i, \mathbf{f}^{\alpha}_{i,\mathbf{l}}(q_i) \neq z_i\right] = (1-\Theta(1/K))^{2^t} = \exp(-\Theta(2^t/K)).$$

Picking large enough constant C and taking union bound over all possible labelings and accepting configurations, we get that the weighted fraction of $\mathbf{q}$ that are bad is

$$\Pr_{\mathbf{q}}[Q(\mathbf{q})] \leq R^{K-1} \cdot L \cdot (K+1)\exp(-\Theta(2^t/K)) = \exp(-\Theta(K)).$$

□

Degree Reduction. Now let us consider the degree of the graph produced by the above reduction. Consider a vertex $(\mathbf{f}_{\mathbf{v}}, \mathbf{q}, \mathbf{z})$. Fix some $i \in [K]$. Let $\mathbf{z}'$ be some accepting assignment of HADAMARD_K with $z'_i \neq z_i$. We first estimate the number of queries $\mathbf{q}'$ with $q'_i = q_i$. Let us consider the i-th test distribution $\mathcal{T}_{i,e_i}$, where e_i is the edge sampled for the i-th test, and denote the constraint on e_i by π. Recall that for each $l \in [L]$ and $r \in \pi^{-1}(l) \subseteq [R]$, the bits $\{q'_{j,r}\}_{j \neq i}$ are sampled by uniformly picking an accepting assignment $\mathbf{x}$ of HADAMARD_K conditioned on $x_i = q'_{i,l}$. Thus there are at least $((K+1)/2)^{|R|} = 2^{\exp(\Omega(K))}$ possible choices of $\mathbf{q}'$. Note that for any such $\mathbf{q}'$, there is an edge between $(\mathbf{f}_{\mathbf{v}}, \mathbf{q}', \mathbf{z}')$ and $(\mathbf{f}_{\mathbf{v}}, \mathbf{q}, \mathbf{z})$. Therefore the degree of the graph produced by the above reduction is at least double exponential in K. We now use the approach in Clementi and Trevisan [28] and Trevisan [102] to reduce the degree to $O(K^3 2^K)$.

For ease of presentation, we look at the argument on the original FGLSS graph without removing bad queries. The same argument applies to the graph with bad queries removed because removing vertices from the graph does not increase the maximum degree, and, as seen above, does not significantly affect the soundness and completeness of the reduction.

Denote the FGLSS graph corresponding to the PCP described in Section 5.1 as G. We first turn G into an unweighted graph. Let w_{min} be the minimum weight of vertices in G, and λ be the ratio between the minimum and maximum weight of vertices in G. Since in the test distribution in Section 5.1 edges of the LABEL-COVER instance are sampled uniformly, we have that λ depends only on K. Let ξ be some granularity parameter. We obtain an unweighted version G' of G by duplicating vertices — we make $\lfloor w/w_{min} \cdot 1/\xi \rfloor \leq 1/\lambda\xi$ vertices for a vertex of weight w, and connect the duplicated vertices with all the neighbors. This step blows up the size of the graph by $O(1/\lambda^2\xi^2)$, and the fractional size of the maximum independent set in G' is within a multiplicative factor of $O(\xi)$ from that of G due to error introduced by $\lfloor \cdot \rfloor$ when duplicating vertices.

As observed in [102], the graph G' is a union of bipartite complete subgraphs. More precisely, for every index i and i-th query $(\mathbf{f}_{\mathbf{v}_i}, \mathbf{q}_i)$, there is a complete bipartite graph between configurations that answer "1" for query $(\mathbf{f}_{\mathbf{v}_i}, \mathbf{q}_i)$ — denoted as $Z_{\mathbf{f}_{\mathbf{v}_i}, \mathbf{q}_i}$ — and configurations that answer "-1" for the same query — denoted as $O_{\mathbf{f}_{\mathbf{v}_i}, \mathbf{q}_i}$. By the way we construct the FGLSS graph, it follows that these complete bipartite subgraphs cover the whole G'. Let l be the maximum size of such sets. We claim that l depends only on K, λ and ξ. To estimate l, consider how many tuples $(\mathbf{f}_{\mathbf{v}}, \mathbf{q}, \mathbf{z})$ can include $(\mathbf{f}_{\mathbf{v}_i}, \mathbf{q}_i)$ on the i-th position. By Theorem 3.10, the degree of the LABEL-COVER graph is $\text{poly}(1/\sigma) = \exp(\Theta(K))$. Since there are K vertices in $\mathbf{v}_i$, the $\mathbf{f}_{\mathbf{v}_i}$ coordinate has at most $\exp(\Theta(K^2))$ neighbors. For $\mathbf{q}_i$, consider an edge e the bits in $\mathbf{q}_i$ that are mapped to the same label $l \in [L]$ according to mapping π_e (or a single bit if e is the i-th edge). There are exactly $(K+1)/2$ possible queries. Enumerating over all labels and sampled edges, this gives an upper-bound of $2^{\exp(\Theta(K))}$. Since each of them can be duplicated by at most $1/\lambda\xi$ times, we have $l = 2^{\exp(\Theta(K))}/\lambda\xi$. Also since for each input bit to the

predicate $\textsc{Hadamard}_K$, exactly half of the accepting assignments of $\textsc{Hadamard}_K$ set that bit to 1 and exactly half to -1 — a property also known as $\textsc{Hadamard}_K$ being balanced — we have $|Z_{\mathbf{f}_{\mathbf{v}_i},\mathbf{q}_i}| = |O_{\mathbf{f}_{\mathbf{v}_i},\mathbf{q}_i}|$.

We now replace the above bipartite complete graphs in G' with the following construction on the same set of vertices $Z_{\mathbf{f}_{\mathbf{f}_i},\mathbf{q}_i}$ and $O_{\mathbf{f}_{\mathbf{f}_i},\mathbf{q}_i}$.

Proposition 7.4 ([102])**.** *For every $\zeta > 0$ and $b \geq 1$, there is a bipartite graph $([b],[b],E)$ of degree at most $d = 3\zeta^{-1}\log(\zeta^{-1})$ such that for any $A, B \subseteq [b]$, $|A| \geq \lfloor \zeta b \rfloor$, $|B| \geq \lfloor \zeta b \rfloor$, we have $(A \times B) \cap E \neq \emptyset$.*

Trevisan [102] called such graphs (b,ζ)-dispersers, and he used a probabilistic argument to prove the above proposition. As argued above, l is a constant depending only on K, thus we can find the desired disperser by exhaustive search. An important property of bipartite dispersers is that given an independent set I of a (b,ζ)-disperser, we have that either $|I \cap A| \leq \zeta b$ or $|I \cap B| \leq \zeta b$.

Denote the replaced graph by G''. To understand how much the above replacement step increases the size of the maximum independent set, note that for any independent set in a disperser, we can get an independent set in the complete bipartite graph by discarding all vertices on one side, which, if we choose to discard the smaller side, accounts for at most a ζ fraction of the vertices on one side of the bipartite graph. Also, each vertex in the FGLSS graph is involved in at most K complete bipartite graphs of this kind, thus the size of the independent set in the new graph is at most an additive $K\zeta$ larger than G'. By choosing $\zeta = O(2^{-K}/K)$, $\xi = O(2^{-K})$, we have that in the soundness case the maximum independent set G' has size $O(2^{-K})$. The maximum degree of G'' is bounded by $K \cdot 3\zeta^{-1}\log(\zeta^{-1}) = O(K^3 2^K)$.

Chapter 8

Superposition Complexity and Hypergraph Coloring

We now study the hardness of coloring 2-colorable 8-uniform hypergraphs. The hardness result we get here also starts from our study of CSP hardness, although it follows a different route compared to the one in Chapter 7. In particular, we study the notion of *superposition complexity* for CSPs.

8.1 Overview of the Reduction

We start by describing the PCP reduction of proving hypergraph coloring hardness used in many previous works such as those mentioned in Section 6.2. Most of these results show hardness of finding an independent set of large fractional size in hypergraphs with small chromatic number. We can view the output of these reductions as NotAllEqual$_k$ CSP instances. The variables correspond to the vertices of a hypergraph, and the NotAllEqual$_k$ constraints correspond to the hyperedges. Note that for hypergraph coloring results, all variables appear positively in such instances and no negations are allowed. An assignment that satisfies all the NotAllEqual$_k$ constraints thus gives a perfect 2-coloring for the hypergraph. In the other direction, a set of vertices in the hypergraph naturally corresponds to a $\{0,1\}$ assignment to the variables in the NotAllEqual$_k$ instance, and the vertices form an independent set if for all constraints in the NotAllEqual$_k$ instance, there is at least 1 variable that is assigned 0.

The overall structure of these reductions are similar to the ones we have seen in Part II. The starting point of the reduction is usually some Label-Cover hardness. We then encode the supposed labeling for the Label-Cover instance with some coding scheme, and design a PCP to test the consistency of the labeling.

The classical choice of encoding is the Long-Code, which encodes m bits of information with 2^{2^m} bits. This is the encoding we use in all previous chapters. The resulting instance has size $n^{O(r)}2^{2^r}$, where poly(n) is the size of the particular

Label-Cover instance we use, and r is the the number of bits we need to encode the labels of the Label-Cover instance. This is not a problem in the previous chapters, because for any pre-specified soundness parameter ε — which we think of as constant — the corresponding $r = O(\log(1/\varepsilon))$ is also a constant, and therefore the whole reduction has polynomial size. However, in the case of hypergraph coloring, the soundness we are aiming for grows with the size of the graph. For instance, if in the soundness case we want to show that there is no independent set of fractional size δ, then we need $r = O(\log(1/\delta))$, and reductions based on Long-Code will produce a hypergraph whose vertex set has size at least $2^{\mathrm{poly}(1/\delta)}$. This huge increase in size makes it impossible to prove hardness results better than polylog n via the Label-Cover plus Long-Code approach.

A much more efficient encoding is the Hadamard-Code, which only uses 2^m bits to encode m bits of information. However, the disadvantage of the Hadamard-Code is that one can only enforce linear constraints on the codewords, which means that we can only start from hard problems involving only linear constraints, and as a result, we lose perfect completeness and can only prove results about almost coloring.

The Low-Degree-Long-Code proposed in [13] lies somewhere between Long-Code and Hadamard-Code. See Section 2.5 for a review of these codes. Dinur and Guruswami [29] obtained hardness result for a variant of hypergraph coloring based on Low-Degree-Long-Code, and the techniques were soon adapted in [42] to get a hardness result of $2^{2^{\Omega(\sqrt{\log \log n})}}$.

Khot and Saket [77] proved a hardness result of $2^{\log^{\Omega(1)} n}$ by using Quadratic-Code, which is the same as Low-Degree-Long-Code with $d = 2$. Their construction, however, is completely different from that in [42].

One can view the Quadratic-Code used in [77] as the Hadamard-Code encoding of matrix M that is symmetric and has rank 1, that is, there exists some $u \in \mathbb{F}_2^m$ such that $M = u \otimes u$. Khot and Saket described a 6-query test such that if some encoding function $f : \mathbb{F}_2^{m \times m} \to \mathbb{F}_2$ passes the test with non-trivial probability, then we can decode it into a low rank matrix.

In order to use this encoding, it seems natural that one would like to construct some variant of Label-Cover where the labels are now matrices, with some linear constraints on the entries of the matrices (since as discussed above we are using Hadamard-Code to encode the matrices). In the completeness case, we would like to have some matrix labelings of rank 1 that satisfies all linear constraints on the vertices as well as projection constraints on the edges, and in the soundness case, not even labelings with low rank matrices can satisfy more than a small fraction of them.

Such Label-Cover hardness result is not readily available. Khot and Saket proposed the notion of *superposition complexity* for quadratic equations. Briefly speaking, let $q(x) = c + \sum_{i=1}^{m} c_i x_i + \sum_{1 \le i < j \le m} c_{ij} x_i x_j = 0$ be a quadratic equation on m $\mathbb{F}_2$-variables. We say that t assignments $a^{(1)}, \ldots, a^{(t)} \in \mathbb{F}_2^m$ satisfy the equation

$q(x) = 0$ *in superposition* if

$$c + \sum_{i=1}^{m} c_i \left(\sum_{l=1}^{t} a_i^{(l)} \right) + \sum_{1 \le i < j \le m} c_{ij} \left(\sum_{l=1}^{t} a_i^{(l)} a_j^{(l)} \right) = 0 \,.$$

If we have a system of quadratic equations, then we say that t assignments satisfy the system of quadratic equations in superposition if each quadratic equation is satisfied in superposition. Having a small number of assignments satisfying quadratic constraints in superposition is exactly the same as having a symmetric low-rank matrix satisfying the linearized version of the constraints, as we discuss in more detail in Section 8.2.

Through a remarkable chain of reductions, Khot and Saket established the inapproximability of quadratic equations with superposition complexity, as well as the actual construction of the Label-Cover with matrix labels. They started with superposition hardness for E3-Sat with gap of $1/n$, and used low-degree testing and sum-check protocol like in the original proof of the PCP theorem [6, 7] to achieve a superposition hardness result for systems of quadratic equations with good soundness and moderate increase in size. This is then followed by a Point versus Surface test which produces the actual Label-Cover instance.

The focus of this chapter is also the construction of such Label-Cover instances. Let t be some odd natural number. A set of t assignments odd-covers an equation (or more generally, a constraint) if the number of assignments that satisfy the equation is odd. We show in Section 8.2 that the notion of odd-covering is equivalent to satisfaction in superposition when the number of assignments is odd. This viewpoint enables us to construct the kind of Label-Cover instance used in [77] very easily. In fact, the reduction in Section 8.3 looks very much like a classical CSP inapproximability proof.

Although simpler, the above observation alone is not sufficient to give us a hardness result better than [77]. The issue here is that for the reduction in Section 8.3 to work for our choice of parameters, the soundness of the Label-Cover that we start with needs to be sub-constant, and a typical Long-Code reduction will again blow up the size of the instance by too much. Hence, for this step, we employ Low-Degree-Long-Code, and the analysis relies on Theorem 2.20, a generalization of the Reed-Muller code testing result of [29].

8.2 Superposition and Odd-Covering

Before we discuss the relation between superposition, odd-covering and low rank matrices, we define an operation on vectors and matrices that we will use frequently.

Definition 8.1. *Define* $D_1 : \mathbb{F}_2^{m+1} \to \mathbb{F}_2^m$ *as the operator that removes the first coordinate of a vector. Define* D_1 *similarly for matrices as the operator that removes the first row and column of a given matrix.*

Khot and Saket [77] defined the notion of satisfying in superposition as follows.

Definition 8.2 (Superposition)**.** *Let $a^{(1)}, \dots, a^{(t)} \in \mathbb{F}_2^m$ be t assignments and $q(x) = 0$ be a quadratic equation in m $\mathbb{F}_2$-variables with*

$$q(x) = c + \sum_{i=1}^{m} c_i x_i + \sum_{1 \le i < j \le m} c_{ij} x_i x_j \,.$$

We say that the t assignments satisfy the equation $q(x) = 0$ in superposition *if*

$$c + \sum_{i=1}^{m} c_i \left(\sum_{l=1}^{t} a_i^{(l)} \right) + \sum_{1 \le i < j \le m} c_{ij} \left(\sum_{l=1}^{t} a_i^{(l)} a_j^{(l)} \right) = 0 \,.$$

Definition 8.3. *Given a system of quadratic equations $\{q_i(x) = 0\}_{i=1}^{L}$ on variables $x_1, \dots, x_m$, its* superposition complexity *is the minimum number t, if it exists, such that there are t assignments $a^{(1)}, \dots, a^{(t)} \in \mathbb{F}_2^m$ that satisfy each equation $q_i(x) = 0$ in superposition.*

We define the odd superposition complexity *(or* even superposition complexity*) to be the minimum odd integer t (or even integer t, respectively) such that there are t assignments that satisfy all equations in superposition.*

Note that by simply adding all-0 assignments, we can argue that the above three notions of superposition complexity differ by at most 1.

We now explain the relation between superposition complexity of quadratic equations and low rank matrices. Assume for simplicity of exposition that the quadratic equation $q(x) = 0$ as defined above is homogeneous, that is, the constant term c and the linear coefficients c_i are all 0.

We can express a homogeneous quadratic equation $q(x) = 0$ with a matrix by defining $C \in \mathbb{F}_2^{m \times m}$, where $C_{ij} = c_{ij}$ for $1 \le i < j \le m$, and $C_{ij} = 0$ otherwise. Let $x = (x_1 \ x_2 \ \dots \ x_m)$. Then $q(x) = 0$ is the same as $\langle C, x \otimes x \rangle = x^T C x = 0$, where $\langle \cdot, \cdot \rangle$ denotes the entry-wise dot product of two matrices. Note that $x \otimes x$ is a symmetric rank-1 matrix.

Suppose now that we have a symmetric matrix A such that $\langle C, A \rangle = 0$. For a fixed C, this is a linear constraint on the entries of A. If in addition A has rank 1, then there exists x_a, such that $A = x_a \otimes x_a$, and by the above, we have that x_a satisfies $q(x_a) = 0$. Therefore, if A is a symmetric rank 1 matrix and $\langle C, A \rangle = 0$, then A encodes an assignment that satisfies the quadratic equation $q(x) = 0$.

The following decomposition lemma from [77] illustrates the situation when A has low rank.

Lemma 8.4. *Let $A \in \mathbb{F}_2^{m \times m}$ be a symmetric matrix of rank k over $\mathbb{F}_2$. Then there exists $l \le 3k/2$ and vectors $v_1, \dots, v_l$ in the column space of A, such that $A = \sum_{i=1}^{l} v_i \otimes v_i$.*

Let A be a low rank matrix and $v_1, \dots, v_l$ be $l \leq 3k/2$ assignments given by Lemma 8.4. Then

$$\begin{aligned} 0 = \langle C, A \rangle &= \sum_{t=1}^{l} \langle C, v_t \otimes v_t \rangle \\ &= \sum_{t=1}^{l} \sum_{1 \leq i < j \leq m} c_{ij} v_{ti} v_{tj} \\ &= \sum_{1 \leq i < j \leq m} c_{ij} \sum_{t=1}^{l} v_{ti} v_{tj} \,. \end{aligned}$$

Therefore we have that $v_1, \dots, v_l$ satisfy $q(x) = 0$ in superposition.

The notion we will now consider is the following, which we call *odd-covering*.

Definition 8.5 (Odd-covering)**.** *Let $a^{(1)}, \dots, a^{(t)} \in \mathbb{F}_2^m$ be t assignments and $q(x) = 0$ be a quadratic equation in m $\mathbb{F}_2$-variables as defined above. We say that the t assignments* odd-cover *the equation $q(x) = 0$ if the number of assignments $a^{(l)}$ that satisfies $q(a^{(l)}) = 0$ is odd.*

The key observation is that odd-covering and satisfying in superposition are equivalent when the number of assignments involved is odd.

Lemma 8.6. *Let t be an odd integer and $a^{(1)}, \dots, a^{(t)} \in \mathbb{F}_2^m$ be t assignments, and $q(x) = 0$ be a quadratic equation in m $\mathbb{F}_2$-variables as defined above. Then the t assignments satisfy $q(x) = 0$ in superposition if and only if the t assignments odd-cover $q(x) = 0$.*

Proof. Using the fact that t is odd, we have the following

$$\begin{aligned} \sum_{l=1}^{t} q(a^{(l)}) &= \sum_{l=1}^{t} \left(c + \sum_{i=1}^{m} c_i a_i^{(l)} + \sum_{1 \leq i < j \leq m} c_{ij} a_i^{(l)} a_j^{(l)} \right) \\ &= t \cdot c + \sum_{l=1}^{t} \sum_{i=1}^{m} c_i a_i^{(l)} + \sum_{l=1}^{t} \sum_{1 \leq i < j \leq m} c_{ij} a_i^{(l)} a_j^{(l)} \\ &= c + \sum_{i=1}^{m} c_i \left(\sum_{l=1}^{t} a_i^{(l)} \right) + \sum_{1 \leq i < j \leq m} c_{ij} \left(\sum_{l=1}^{t} a_i^{(l)} a_j^{(l)} \right) . \end{aligned}$$

Now observe that the t assignments odd-cover $q(x) = 0$ if and only if the number of assignments that does not satisfy $q(x) = 0$ is even, which is equivalent to saying that the left hand side of the above equation is 0, and that by definition means that the t assignments satisfy $q(x) = 0$ in superposition. □

In the description above, we assumed that the quadratic equation $q(x) = 0$ is homogeneous, which allows us to encode it with a matrix $C \in \mathbb{F}_2^{m \times m}$ and express

the whole equation as $\langle C, A\rangle = 0$, where $A = x \otimes x$. For quadratic equations that are not homogeneous, we encode them with a $(m+1) \times (m+1)$ matrix. In particular, for $q(x) = c + \sum c_i x_i + \sum c_{ij} x_i x_j = 0$, we have matrix C, where $C_{11} = c$, $C_{1i} = c_{i-1}$ for $i = 2, \dots, m+1$, and $C_{ij} = c_{i-1,j-1}$ for $2 \le i < j \le m+1$. As for the variable vector, we insert an entry with value 1 at the beginning of x.

Definition 8.7. *Given a matrix $A \in \mathbb{F}_2^{(m+1)\times(m+1)}$. We say that A is* pseudo-quadratic *if the following holds:*

- *A is symmetric.*
- *$A_{1,1} = 1$.*
- *For all $i = 2, \dots, m+1$, $A_{1,i} = A_{i,1} = A_{i,i}$.*

Note that for vector $v \in \mathbb{F}_2^{m+1}$ such that $v_1 = 1$, $v \otimes v$ is a pseudo-quadratic rank-1 matrix.

We prove a stronger form of Lemma 8.4 for pseudo-quadratic matrices where we decode a low rank pseudo-quadratic matrix into an odd number of assignments.

Lemma 8.8. *Let $A \in \mathbb{F}_2^{(m+1)\times(m+1)}$ be a pseudo-quadratic matrix of rank k over $\mathbb{F}_2$. Then there exists an odd integer $k_0 < 3k/2+1$, and vectors $v_1, \dots, v_{k_0} \in \mathbb{F}_2^{m+1}$, such that for all $i \in [k_0]$, $v_{i,1} = 1$, and $A = \sum_{i=1}^{k_0} v_i \otimes v_i$. Moreover, for all $i \in [k_0]$, $D_1(v_i)$ is in the column space of $D_1(A)$.*

Proof. Let $A' = D_1(A)$. Note that A' is symmetric and has rank at most k. Therefore by Lemma 8.4, there exists $l < 3k/2$ vectors $u_1, \dots, u_l \in \mathbb{F}_2^m$, such that $A' = \sum_{i=1}^{l} u_i \otimes u_i$. Now consider vectors $v_1, \dots, v_l \in \mathbb{F}_2^{m+1}$, where for each i, $v_{i,1} = 1$ and $v_{i,j} = u_{i,j-1}$ for $j = 2, \dots, m+1$. Let $A'' = \sum_{i=1}^{l} v_i \otimes v_i$, and $B = A - A''$. For $j, j' \in \{2, \dots, m+1\}$, we have

$$A''_{j,j'} = \sum_{i=1}^{l} v_{i,j} v_{i,j'} = \sum_{i=1}^{l} u_{i,j-1} u_{i,j'-1} = A'_{j-1,j'-1} = A_{j,j'}\,.$$

Moreover, we have

$$A''_{1,j} = \sum_{i=1}^{l} v_{i,1} v_{i,j} = \sum_{i=1}^{l} v_{i,j} v_{i,j} = A''_{j,j} = A_{j,j} = A_{1,j}\,.$$

We conclude that for all $(i,j) \neq (1,1)$, $A_{i,j} = A''_{i,j}$. Note that $A''_{1,1} = (l \bmod 2)$. Therefore if $A''_{1,1} = 1 = A_{1,1}$, then we have l is odd and $A = \sum_{i=1}^{l} v_i \otimes v_i$ as promised. Otherwise l is even. Let $e = (1\ 0\ \dots\ 0) \in \mathbb{F}_2^{m+1}$. Then $A = \sum_{i=1}^{l} v_i \otimes v_i + e \otimes e$ gives the desired decomposition. □

The following lemma summarizes the discussion at the beginning of this section and relates odd superposition complexity with low-rank pseudo-quadratic matrices.

Lemma 8.9. *Let $q_1(x) = 0, \ldots, q_s(x) = 0$ be a set of s quadratic equations on variable $x_1, \ldots, x_m$, and let $Q_1, \ldots, Q_s \in \mathbb{F}_2^{(m+1)\times(m+1)}$ be their corresponding matrix forms. Suppose there is a pseudo-quadratic matrix $A \in \mathbb{F}_2^{(m+1)\times(m+1)}$ such that $\mathrm{rank}(A) \leq k$ and for all $i \in [s]$, $\langle Q_i, A\rangle = 0$, then there exists $l < 3k/2 + 1$ vectors $a^{(1)}, \ldots, a^{(l)} \in \mathbb{F}_2^{m+1}$ in the column space of A, for some odd integer l, such that $A = \sum_{i=1}^{l} a^{(i)} \otimes a^{(i)}$. This implies that the assignments $D_1(a^{(1)}), \ldots, D_1(a^{(l)})$ satisfy all equations $q_1(x) = 0, \ldots, q_s(x) = 0$ in superposition.*

Proof. Apply Lemma 8.8 to A, and let $v_1, \ldots, v_l$ be the vectors we get, with $v_{i1} = 1$ for $i \in [l]$, and $A = \sum_{i\in[l]} v_i \otimes v_i$. We now verify that $D_1(v_1), \ldots, D_1(v_l)$ satisfy all equations in superposition.

Consider equation i for $i \in [s]$. We have

$$\begin{aligned} 0 = \langle Q_i, A\rangle &= \sum_{i=1}^{l} \langle Q_i, v_i \otimes v_i\rangle \\ &= \sum_{i=1}^{l} q_i(v_i)\,. \end{aligned}$$

By definition, we have that $v_1, \ldots, v_l$ satisfy q_i in superposition. □

8.3 Superposition Hardness for Gap-TSA

Let b be some large integer parameter. Recall that the TSA predicate is a predicate on 5 $\mathbb{F}_2$-variables defined as follows

$$\mathrm{TSA}(x_1, \ldots, x_5) = 1 + x_1 + x_2 + x_3 + x_4 x_5\,.$$

From the definition, we can see that TSA instances are systems of quadratic equations, each involving exactly 5 $\mathbb{F}_2$-variables.

The predicate was studied in [55] as a starting point of an efficient PCP construction. For the predicate itself, Håstad and Khot proved that it is approximation resistant on satisfiable instances.

We now prove an superposition hardness result for Gap-TSA.

Theorem 8.10. *There is a reduction that takes as input a* E3-Sat *instance of size n, and outputs a* TSA *instance of size $n^{O(b \log\log n)}$ with the following properties:*

- *If the* E3-Sat *instance is satisfiable, then there is an assignment that satisfies all TSA constraints.*

- *If the* E3-Sat *instance is unsatisfiable, then for any odd integer* $t < (\log n)^b$*, and any* t *assignments, at most a* $15/16$ *fraction of the* TSA *constraints are satisfied in superposition.*

The reduction runs in time $n^{O(b \log\log n)}$.

Proof. The reduction follows a similar approach as a typical inapproximability hardness reduction.

Given a E3-Sat instance of size n, we apply Theorem 3.10 with soundness $1/(1000(\log n)^{2b})$ to get a Label-Cover instance. This gives the parameter $r = (2b \log\log n + O(1))/\varepsilon_0$, where ε_0 is some universal constant. The vertex set of the bipartite graph has size $n^{O(b \log\log n)}$, and the label sets are $L = \{0,1\}^r$ and $R = \{0,1\}^{3r}$. Let $d = \Theta(b \log\log n)$ be such that $2^{d/2-4} \approx (\log n)^b + 3$. This implies also that $2^d \approx 256(\log n)^{2b}$.

For each $u \in U$ and $v \in V$, we expect functions $f_u : \mathsf{P}_{r,d} \to \{-1,1\}$ and $g_v : \mathsf{P}_{3r,d} \to \{-1,1\}$. We assume that all functions are folded over constant. The entries of the functions correspond to variables of some TSA instance. Therefore the number of variables in the output instance is $n^{O(b \log\log n)} \cdot (3r)^{(1+o(1))d} = n^{O(b \log\log n)}$, and the number of constraints is polynomial in the number of variables.

Consider the following test:

1. Sample random edge $e = \{u_1, u_2\} \sim E$. Let π be the projection on the edge, and let f and g be the functions associated with u_1 and u_2.
2. Sample uniformly random query $x \sim \mathsf{P}_{r,d}$, $y \sim \mathsf{P}_{3r,d}$, and $v, w \sim \mathsf{P}_{3r,d/2}$.
3. Construct query $z := x \circ \pi + y + vw \in \mathsf{P}_{3r,d}$.
4. Accept iff $f(x)g(y)g(z)(g(v) \wedge g(w)) = 1$, where $\wedge$ here denotes the binary operator that evaluates to -1 when both operands are -1, and 1 otherwise.

The completeness is straightforward. In this case, the Label-Cover instance has a perfect labeling. Setting the functions to be the Low-Degree-Long-Code encoding of the labels gives an assignment that satisfies all TSA constraints.

In the soundness case, there exists some $t < (\log n)^b$ assignments that satisfy in superposition a $15/16$ fraction of the constraints. That is, for each $u_1 \in U$ and $u_2 \in V$, there are t functions that are folded over constant, $f^{(1)}, \ldots, f^{(t)} : \mathsf{P}_{r,d} \to \{-1,1\}$ and $g^{(1)}, \ldots, g^{(t)} : \mathsf{P}_{3r,d} \to \{-1,1\}$ such that over random sample of edges $\{u_1, u_2\}$ and queries x, y, z, v, w, with probability at least $15/16$, the number of $i \in [t]$ such that $f^{(i)}(x)g^{(i)}(y)g^{(i)}(z)(g^{(i)}(v) \wedge g^{(i)}(w)) = 1$ is odd. By an averaging argument, we have that for at least $3/4$ of the edges, over random sample of queries, the above holds with probability at least $3/4$. Call such an edge *good*.

We assume that the functions are folded in the same way. Recall that when applying folding, we partition the domain of the functions into equivalence classes, define the function value in one of the equivalence classes, and then extend to the

full domain by adding appropriate constants. For our reduction, we identify one equivalence class for each vertex, and the t functions associated with it supply value only for that equivalence class. This is to make sure $f^{(1)}, \ldots, f^{(t)}$ and $g^{(1)}, \ldots, g^{(t)}$ corresponds exactly to t assignments in superposition.

Fix a good edge for now, and we drop the subscripts u_1 and u_2. Then we have the following

$$\frac{1}{2} + \frac{1}{2} \mathop{\mathbf{E}}_{x,y,z,v,w} \left[\prod_{i=1}^{t} \left(f^{(i)}(x) g^{(i)}(y) g^{(i)}(z) (g^{(i)}(v) \wedge g^{(i)}(w)) \right) \right] \geq \frac{3}{4},$$

or

$$\mathop{\mathbf{E}}_{x,y,z,v,w} \left[\prod_{i=1}^{t} \left(f^{(i)}(x) g^{(i)}(y) g^{(i)}(z) (g^{(i)}(v) \wedge g^{(i)}(w)) \right) \right] \geq \frac{1}{2}.$$

Let $f' = \prod_{i=1}^{t} f^{(i)}$, and $g' = \prod_{i=1}^{t} g^{(i)}$. Since t is odd, we have that f' and g' are both folded over constant. Taking the Fourier expansion of f' and g', we have the following

$$\begin{aligned}
\frac{1}{2} &\leq \mathop{\mathbf{E}}_{x,y,z,v,w} \left[\prod_{i=1}^{t} \left(f^{(i)}(x) g^{(i)}(y) g^{(i)}(z) (g^{(i)}(v) \wedge g^{(i)}(w)) \right) \right] \\
&= \mathbf{E} \left[f'(x) g'(y) g'(z) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right] \\
&= \sum_{\substack{\alpha \in \Lambda_{r,d} \\ \beta_1, \beta_2 \in \Lambda_{3r,d}}} \widehat{f'}_{\alpha} \widehat{g'}_{\beta_1} \widehat{g'}_{\beta_2} \\
&\quad \mathop{\mathbf{E}}_{x,y,z,v,w} \left[\chi_{\alpha}(x) \chi_{\beta_1}(y) \chi_{\beta_2}(x \circ \pi + y + vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right] \\
&= \sum_{\beta \in \Lambda_{3r,d}} \widehat{f'}_{\pi_2(\beta)} \widehat{g'}^{2}_{\beta} \mathop{\mathbf{E}}_{vw} \left[\chi_{\beta}(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right].
\end{aligned}$$

Applying Cauchy-Schwarz and using Parseval, we have

$$\begin{aligned}
\frac{1}{4} &\leq \left(\sum_{\beta \in \Lambda_{3r,d}} \widehat{g'}^{2}_{\beta} \right) \left(\sum_{\beta \in \Lambda_{3r,d}} \widehat{f'}^{2}_{\pi_2(\beta)} \widehat{g'}^{2}_{\beta} \mathop{\mathbf{E}}_{vw} \left[\chi_{\beta}(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right]^2 \right) \\
&= \sum_{\beta \in \Lambda_{3r,d} : \mathrm{wt}(\beta) \leq 2^{d-4}} \widehat{f'}^{2}_{\pi_2(\beta)} \widehat{g'}^{2}_{\beta} \mathop{\mathbf{E}}_{vw} \left[\chi_{\beta}(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right]^2 + \\
&\quad \sum_{\beta \in \Lambda_{3r,d} : \mathrm{wt}(\beta) > 2^{d-4}} \widehat{f'}^{2}_{\pi_2(\beta)} \widehat{g'}^{2}_{\beta} \mathop{\mathbf{E}}_{vw} \left[\chi_{\beta}(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right]^2.
\end{aligned}$$

For the terms where $\mathrm{wt}(\beta) > 2^{d-4}$, we apply Theorem 2.20 to get

$$\left| \mathop{\mathbf{E}}_{vw} \left[\chi_\beta(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right] \right| \leq 2^{-(2^{d/2-4}-t)/2},$$

and therefore

$$\sum_{\beta \in \Lambda_{3r,d} : \mathrm{wt}(\beta) > 2^{d-4}} \widehat{f'}^2_{\pi_2(\beta)} \widehat{g'}^2_{\beta}$$

$$\mathop{\mathbf{E}}_{vw} \left[\chi_\beta(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right]^2 \leq 2^{-(2^{d/2-4}-t)} < \frac{1}{8}.$$

This gives us

$$\sum_{\beta \in \Lambda_{3r,d} : \mathrm{wt}(\beta) \leq 2^{d-4}} \widehat{f'}^2_{\pi_2(\beta)} \widehat{g'}^2_{\beta}$$

$$\geq \sum_{\beta \in \Lambda_{3r,d} : \mathrm{wt}(\beta) \leq 2^{d-4}} \widehat{f'}^2_{\pi_2(\beta)} \widehat{g'}^2_{\beta} \mathop{\mathbf{E}}_{vw} \left[\chi_\beta(vw) \prod_{i=1}^{t} (g^{(i)}(v) \wedge g^{(i)}(w)) \right]^2 \geq \frac{1}{8}.$$

Let $\{u_1, u_2\}$ be a good edge. Consider the following labeling strategy: for u_1, pick α with probability $\widehat{f'}^2_{\alpha}$ and pick a random label from $\mathrm{supp}(\alpha)$, and for u_2, pick β with probability $\widehat{g'}^2_{\beta}$ and pick a random label from $\mathrm{supp}(\beta)$. The procedure is well defined because f' and g' are all folded, and thus by Lemma 2.53, $\mathrm{supp}(\alpha)$ and $\mathrm{supp}(\beta)$ are nonempty. Also, for β such that $\mathrm{wt}(\beta) \leq 2^{d-4} < 2^{d-3}$, by Lemma 2.55, the assignments in $\mathrm{supp}(\beta)$ all satisfy the clauses in u_2. Then the probability that the labeling of u_1 and u_2 satisfies the projection constraint on a good edge $\{u_1, u_2\}$ is at least $\frac{1}{2^{d-4}} \sum_{\beta : \mathrm{wt}(\beta) \leq 2^{d-4}} \widehat{f'}^2_{\pi_2(\beta)} \widehat{g'}^2_{\beta} \geq 1/(8 \cdot 2^{d-4}) > 1/(100(\log n)^{2b})$. Since there are at least a $3/4$ fraction of good edges, overall the labeling satisfies more than $(3/4) \cdot (1/(100(\log n)^{2b})) > 1/(1000(\log n)^{2b})$, contradicting the fact that in the soundness case the Label-Cover instance does not have labeling with value larger than $1/(1000(\log n)^{2b})$. This completes the proof. □

8.4 Label-Cover with Matrix Labels

We now use Theorem 8.10 to construct a Label-Cover instance with properties similar to that in [77].

Let b be some large integer parameter, and $t \approx (\log n)^b$ be an odd integer. Given a TSA instance with t-superposition hardness gap of $15/16$ from Theorem 8.10, consider the following 2-Prover-1-Round projection game:

1. The referee picks a TSA constraint, which we denote as $\mathcal{C}(x_1, x_2, x_3, x_4, x_5)$, and then picks randomly $i \in [5]$.

2. The referee sends x_i to Alice and $\mathcal{C}$ to Bob.
3. Alice replies with $a \in \mathbb{F}_2^t$, and Bob replies with $b \in (\mathbb{F}_2^t)^5$.
4. The referee accepts iff b, interpreted as t $\mathbb{F}_2$ assignments, satisfies $\mathcal{C}$ in superposition, and $b_i = a$.

This is a projection game with perfect completeness and soundness 79/80.

Using Theorem 3.9, we get the following Label-Cover construction. Note that it is important that we use Theorem 3.9 in [92] instead of the original version in [93], because the answer size is non-constant and it is important that the rate at which soundness decreases is independent of that.

Theorem 8.11. *There exists a reduction that takes a* E3-Sat *instance of size n, and outputs a* Label-Cover *instance $(U, V, E, L, R, \Pi, \Gamma)$ with the following properties:*

- *The bipartite graph (U, V, E) has size $\exp((\log n)^{(2+o(1))b})$, and the reduction runs in time $\exp((\log n)^{(2+o(1))b})$.*
- *The label set $R = \mathbb{F}_2^{m_r}$, $L = \mathbb{F}_2^{m_l}$, where $m_l, m_r = (\log n)^{(2+o(1))b}$.*
- *For each $v \in V$, there is a set of quadratic $\mathbb{F}_2$ equations, each involving 5 of the m_r coordinates of the labeling of v. The set of valid labelings $\Gamma(v)$ are those that satisfy all quadratic equations.*
- *For each edge $e \in E$, there is a set $S_e \subseteq [m_r]$, such that $\pi_e : \mathbb{F}_2^{m_r} \to \mathbb{F}_2^{m_l}$ is defined as $\pi_e(r) = r_{S_e}$.*
- *If the* E3-Sat *instance is satisfiable, then there is a labeling that satisfies all quadratic equation constraints for all vertices $v \in V$, and all projection constraints for all edges.*
- *If the* E3-Sat *instance is unsatisfiable, then for any odd integer $l < (\log n)^b$, any labeling $\sigma^{(1)}, \ldots, \sigma^{(l)}$ for the vertices in U and V, the following does not hold simultaneously:*
  - *For each $v \in V$, and for each equation q associated with v, the assignment given by $\sigma^{(1)}(v), \ldots, \sigma^{(l)}(v)$ satisfy q in superposition.*
  - *For at least $2^{-(\log n)^{(2+o(1))b}}$ fraction of the edges $e = \{u, v\}$, we have $\pi_e(\sigma^{(j)}(v)) = \sigma^{(j)}(u)$, $\forall j \in [l]$.*

We now convert the above into a Label-Cover instance with matrix label and rank soundness constraint.

Theorem 8.12. *There exists a reduction that takes a* E3-Sat *instance of size n, and outputs a* Label-Cover *instance $(U, V, E, L, R, \Pi, \Gamma)$ with the following properties:*

- *The bipartite graph* (U, V, E) *has size* $\exp((\log n)^{(2+o(1))b})$*, and the reduction runs in time* $\exp((\log n)^{(2+o(1))b})$*.*
- *The label sets are matrices* $R = \mathbb{F}_2^{(m_r+1)\times(m_r+1)}$*,* $L = \mathbb{F}_2^{(m_l+1)\times(m_l+1)}$*, where* $m_l, m_r = (\log n)^{(2+o(1))b}$*.*
- *For each* $v \in V$*, there is a set of homogeneous linear* $\mathbb{F}_2$ *equations involving entries of the labeling of* v*. The set of valid labelings* $\Gamma(v)$ *consists of matrices that satisfy all the associated linear equations.*
- *For each edge* $e \in E$*, there is a set* $S_e \subseteq [m_r + 1]$*,* $1 \in S_e$*, such that* $\pi_e : \mathbb{F}_2^{(m_r+1)\times(m_r+1)} \to \mathbb{F}_2^{(m_l+1)\times(m_l+1)}$ *is defined as* $\pi_e(r) = r_{S_e}$*.*
- *If the* E3-Sat *instance is satisfiable, then for each* $u \in U$*, there is a labeling* $M_u = x_u \otimes x_u$ *where* $x_{u,1} = 1$*, and for each* $v \in V$*, there is a labeling* $M_v = x_v \otimes x_v$ *where* $x_{v,1} = 1$*, such that for all* $v \in V$*,* $M_v \in \Gamma(v)$*, and for all* $e \in E$*,* $\pi_e(M_v) = M_v|_{S_e} = M_u$*.*
- *If the* E3-Sat *instance is unsatisfiable, then for any labeling* σ *for the vertices in* U *and* V*, the following does not hold simultaneously:*
  - *For each* $v \in V$*, the matrix* $\sigma(v)$ *is pseudo-quadratic, has* $\operatorname{rank}(\sigma(v)) \leq (\log n)^b/2$*, and is valid* $\sigma(v) \in \Gamma(v)$*.*
  - *For at least* $2^{-(\log n)^b}$ *fraction of the edges* $e = \{u, v\}$*, we have* $\pi_e(\sigma(v)) = \sigma(u)$*.*

Proof. We start with the Label-Cover instance from the previous theorem.

The underlying bipartite graph of the new instance is exactly the same. The parameters m_r and m_l are the same as before. The labels for $u \in U$ in the new instance are now matrices from $\mathbb{F}_2^{(m_l+1)\times(m_l+1)}$, and the labels for $v \in V$ are from $\mathbb{F}_2^{(m_r+1)\times(m_r+1)}$. The constraints for labelings for vertices in $v \in V$ are the following:

1. The matrix label M is symmetric, and for $i = 2, \ldots, m_r + 1$, we have $M_{i,i} = M_{1,i} = M_{i,1}$. These are all homogeneous linear constraints. Note that if in addition we have $M_{1,1} = 1$, then we get that M is pseudo-quadratic. Here, however, we do not include the latter constraint as it is not homogeneous. In fact, this will be handled by the inner verifier.
2. For each quadratic constraint in the previous instance, we include the linearized version of it in the new instance. That is, term $x_i x_j$ is replaced by entry $(i + 1, j + 1)$ of the matrix, term x_i is replaced by entry $(1, i + 1)$, and constant 1 is replaced by entry $(1, 1)$.

For edge e, let S_e be the set associated with its projection in the old instance, then in the new instance is defined by the set $S'_e = \{1\} \cup \{i + 1 \mid i \in S_e\}$.

The completeness case is straightforward. For the soundness case, suppose that there are pseudo-quadratic matrices M_u and M_v for each $u \in U$ and $v \in V$, such that M_v satisfies homogeneous linear constraints associated with v, $\text{rank}(M_v) \leq k$, and that for $2^{-(\log n)^b}$ fraction of the edges e, $(M_v)|_{S_e} = M_u$.

For such an edge $e = \{u, v\}$, by Lemma 8.9, there exists odd integer $l < 3/2 \cdot (\log n)^b/2 < (\log n)^b$ vectors $v_1, \ldots, v_l \in \mathbb{F}_q^{m_r+1}$, where $v_{i,1} = 1$ for $i \in [l]$, such that $M_v = \sum_{i=1}^{l} v_i \otimes v_i$, and the assignments $D_1(v_1), \ldots, D_1(v_l)$ satisfy in superposition the quadratic constraints of the old Label-Cover instance. For vertex u, we have that $\text{rank}(M_u) = \text{rank}((M_v)|_S) \leq \text{rank}(M_v)$. Also, $M_u = \sum_{i=1}^{l} v_i|_S \otimes v_i|_S$, and that $D_1(v_i)|_{S-\{1\}}$ are in the column space of $D_1(M_u)$. Therefore, for any $i \in [l]$, if we take a uniformly random vector in the column space of $D_1(M_u)$, then with probability at least $2^{-(\log n)^b/2}$, it will be equal to $(v_i')|_S$. Repeat this for all $i \in [l]$, and we have that these labelings of u all satisfy the projection constraint with probability at least $2^{-(\log n)^{2b}}$.

Overall, this labeling satisfies $2^{-(\log n)^b} 2^{-(\log n)^{2b}} = 2^{-(\log n)^{(2+o(1))b}}$ fraction of the edges in the old instance. □

8.5 Inapproximability of Hypergraph Coloring

We now compose the Label-Cover from Theorem 8.12 with Quadratic-Code inner-verifier to get inapproximability result for hypergraph coloring.

Theorem 8.13. *There is a reduction that takes as input a* E3-Sat *instance of size n, outputs a 8-uniform hypergraph H with the following properties:*

- *The size H and the running time of the reduction are both upper-bounded by* $\exp((\log n)^{(4+o(1))b})$.
- *If the* E3-Sat *instance is satisfiable, then H is 2-colorable.*
- *If the* E3-Sat *instance is unsatisfiable, then H does not have independent set of fractional size larger than* $2^{-O((\log n)^b)}$.

*In other words, it is quasi-*NP*-hard to color a 2-colorable 8-uniform hypergraph of size N with less than* $2^{(\log N)^{1/4-o(1)}}$ *colors.*

The following proof is based on a note by Girish Varma [103].

Given the Label-Cover instance from Theorem 8.12, we expect for each vertex $v \in V$ a function $f_v : \mathbb{F}_2^{(m_r+1)\times(m_r+1)} \to \mathbb{F}_2$. The expected encoding for matrix label $\sigma(v) = a_v \otimes a_v$ is $f_v(A) = \langle a_v \otimes a_v, A \rangle = a_v^T A a_v$. Let $\mathcal{H}_v \subseteq \mathbb{F}_2^{(m_r+1)\times(m_r+1)}$ be the dual of the subspace of the set of pseudo-quadratic matrices that satisfies the linear constraints associated with v. The function f_v is folded over $\mathbb{F}_2^{(m_r+1)\times(m_r+1)}/\mathcal{H}_v$.

Consider the following Boolean 8-uniform test:

- Choose $u \in U$ uniformly at random, and $v, w \in V$ uniformly and independently at random from the neighbors of u. Let $\pi, \sigma : \mathbb{F}_2^{(m_r+1)\times(m_r+1)} \to \mathbb{F}_2^{(m_l+1)\times(m_l+1)}$ be the projections corresponding to the edges (u, v) and (u, w) respectively, and let S_π and S_σ be the index set associated with them.

- Uniformly and independently sample $X_1, X_2, Y_1, Y_2 \in \mathbb{F}_2^{(m_r+1)\times(m_r+1)}$, $F \in \mathbb{F}_2^{(m_l+1)\times(m_l+1)}$, and $x, y, z, x', y', z' \in \mathbb{F}_2^{m_r+1}$. Let $e \in \mathbb{F}_2^{m_r+1}$ be the vector with only the 1-st entry 1 and the rest 0.

- Accept if and only if the following 8 values are not all equal:

$$\begin{array}{ll} f_v(X_1) & f_v(X_3) \quad \text{where } X_3 := X_1 + x \otimes y + F \circ \pi \\ f_v(X_2) & f_v(X_4) \quad \text{where } X_4 := X_2 + (x+e) \otimes z + F \circ \pi \\ f_w(Y_1) & f_w(Y_3) \quad \text{where } Y_3 := Y_1 + x' \otimes y' + F \circ \sigma + e \otimes e \\ f_w(Y_2) & f_w(Y_4) \quad \text{where } Y_4 := Y_2 + (x'+e) \otimes z' + F \circ \sigma + e \otimes e \end{array}$$

We denote by $\mathcal{T}$ the test distribution.

Let H be the output hypergraph. The vertex set of H, denoted by $\mathcal{V}(H)$, has size

$$\exp((\log n)^{(2+o(1))b}) \cdot 2^{(\log n)^{2(2+o(1))b}} = \exp((\log n)^{(4+o(1))b}) =: N\,.$$

8.5.1 Completeness

Let $y_v \otimes y_v$ for $v \in V$ and $x_u \otimes x_u$ for $u \in U$ be a perfect labeling for the Label Cover instance, with $y_{v,1} = x_{u,1} = 1$ and for each edge $e = \{u, v\} \in E$, we have $(y_v)|_{S_e} = x_u$. Consider the 2-coloring where for each $v \in V$, $f_v(X) = y_v^T X y_v = \langle X, y_v \otimes y_v \rangle$. Such a function is constant over cosets of $\mathcal{H}_v$. Let $x_1 := \langle X_1, y_v \otimes y_v \rangle$, $x_2 := \langle X_2, y_v \otimes y_v \rangle$, $y_1 := \langle Y_1, y_w \otimes y_w \rangle$, $y_2 := \langle Y_2, y_w \otimes y_w \rangle$, and $f := \langle F, x_u \otimes x_u \rangle$. Note that $\langle F, x_u \otimes x_u \rangle = \langle F, \pi_{u,v}(y_v \otimes y_v) \rangle = \langle F \circ \pi_{uv}, y_v \otimes y_v \rangle$. Also, $\langle e \otimes e, y_v \otimes y_v \rangle = \langle e, y_v \rangle = 1$. Therefore, the value of the 8 queries are

$$\begin{array}{ll} x_1 & x_1 + \langle y_v, x \rangle \langle y_v, y \rangle + f \\ x_2 & x_2 + (\langle y_v, x \rangle + 1) \langle y_v, z \rangle + f \\ y_1 & y_1 + \langle y_w, x' \rangle \langle y_w, y' \rangle + f + 1 \\ y_2 & y_2 + (\langle y_w, x' \rangle + 1) \langle y_w, z' \rangle + f + 1 \end{array}$$

We finish the proof of the completeness case by a case analysis.

If $\langle y_v, y \rangle = \langle y_w, y' \rangle = 0$, then the sum of entries in the first and third row is 1, which means that there are different values. Similarly, we conclude that if $\langle y_v, z \rangle = \langle y_w, z' \rangle = 0$, then using similar argument as above, there are different values in the second and the fourth row. The same applies to the case when $\langle y_v, x \rangle = \langle y_2, x' \rangle = 1$, and the case when $\langle y_v, x \rangle = \langle y_w, x' \rangle = 0$.

Suppose now that $\langle y_v, x \rangle = 1$ and all entries are equal. Then from the second row, we have that $f = 0$, and from the first row, we get $\langle y_v, y \rangle = 0$. By the

discussion above, we have that $\langle y_w, y' \rangle = 1$, and the third row gives us $\langle y_w, x' \rangle = 1$, but then the two entries on the last row are different.

Suppose otherwise that $\langle y_v, x \rangle = 0$ and all entries are equal. Then from the first row, we have $f = 0$, and the second row implies $\langle y_v, z \rangle = 0$. By the discussion above, we must have $\langle y_w, z' \rangle = 1$, and the last row gives $\langle y_w, x' \rangle = 0$, leaving two different entries in the third row.

Hence f_v gives a valid 2-coloring of H.

8.5.2 Soundness

Let $\delta = 2^{-(\log n)^b}$ be the soundness parameter from Theorem 8.12 and let $k = (\log n)^b/2$ be the rank upper-bound from Theorem 8.12.

Lemma 8.14. *If there is an independent set in H of relative size s, then*

$$s^8 \leq \delta + \frac{1}{2^{k/2+1}}\,.$$

Proof. Consider any set $A \subseteq \mathcal{V}(H)$ of fractional size s. For every $v \in V$, let $f_v : \mathbb{F}_2^{(m_r+1)\times(m_r+1)} \to [0,1]$ be the indicator function of A on vertices associated with v, extended such that it is constant over cosets of $\mathcal{H}_v$. The fractional size of A is given by

$$\mathop{\mathbf{E}}_{\substack{v\sim V \\ X\sim\mathbb{F}_2^{(m_r+1)\times(m_r+1)}}} [f_v(X)] \;=\; \mathop{\mathbf{E}}_{v\sim V}\left[\hat{f}_{v,0}\right]\,.$$

The set A is an independent set if and only if

$$\Theta := \mathop{\mathbf{E}}_{u,v,w}\mathop{\mathbf{E}}_{X_i,Y_i\sim\mathcal{T}}\prod_{i=1}^{4} f_v(X_i)f_w(Y_i) = 0\,. \tag{8.1}$$

Taking Fourier expansion and considering expectations over X_1, X_2, Y_1, Y_2, we get the following:

$$\begin{aligned}\Theta \;=\; \mathop{\mathbf{E}}_{u,v,w}\sum_{\alpha_1,\alpha_2,\beta_1,\beta_2\in\mathbb{F}_2^{(m_r+1)\times(m_r+1)}}\mathop{\mathbf{E}}_{F,x,x'}\Bigg[\;&\\ &\hat{f}^2_{v,\alpha_1}\mathop{\mathbf{E}}_{y}[\chi_{\alpha_1}(x\otimes y)]\chi_{\alpha_1}(F\circ\pi)\\ &\hat{f}^2_{v,\alpha_2}\mathop{\mathbf{E}}_{z}[\chi_{\alpha_2}((x+e)\otimes z)]\chi_{\alpha_2}(F\circ\pi)\\ &\hat{f}^2_{w,\beta_1}\mathop{\mathbf{E}}_{y'}[\chi_{\beta_1}(x'\otimes y')]\chi_{\beta_1}(F\circ\sigma)\chi_{\beta_1}(e\otimes e)\\ &\hat{f}^2_{w,\beta_2}\mathop{\mathbf{E}}_{z'}[\chi_{\beta_2}((x'+e)\otimes z')]\chi_{\beta_2}(F\circ\sigma)\chi_{\beta_2}(e\otimes e)\Bigg]\,.\end{aligned}$$

Denote the term inside $\mathbf{E}_{F,x,x'}[\cdot]$ as $Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)$.

For the characters involving F, we have

$$\begin{aligned}&\mathop{\mathbf{E}}_{F}\left[\chi_{\alpha_1}(F\circ\pi)\chi_{\alpha_2}(F\circ\pi)\chi_{\beta_1}(F\circ\sigma)\chi_{\beta_2}(F\circ\sigma)\right]\\ =\ &\mathop{\mathbf{E}}_{F}\left[(-1)^{\langle\pi(\alpha_1+\alpha_2),F\rangle+\langle\sigma(\beta_1+\beta_2),F\rangle}\right]\ ,\end{aligned}$$

and since $F\in\mathbb{F}_2^{(m_l+1)\times(m_l+1)}$ is chosen uniformly at random, the above is 0 unless $\pi(\alpha_1+\alpha_2)=\sigma(\beta_1+\beta_2)$.

Let $\nu(\alpha):=\langle\alpha,e\otimes e\rangle$. Taking expectations over x,y,z,x',y',z', we have that when $\pi(\alpha_1+\alpha_2)\neq\sigma(\beta_1+\beta_2)$, $Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)=0$, and otherwise

$$\begin{aligned}&Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)\\ =\ &(-1)^{\nu(\beta_1+\beta_2)}\hat{f}^2_{v,\alpha_1}\hat{f}^2_{v,\alpha_2}\hat{f}^2_{w,\beta_1}\hat{f}^2_{w,\beta_2}\\ &\Pr_x\left[\alpha_1x=0\wedge\alpha_2x=\alpha_2e\right]\Pr_{x'}\left[\beta_1x=0\wedge\beta_2x'=\beta_2e\right]\ .\end{aligned}$$

The terms that are potentially non-zero can now be partitioned into three parts:

$$\begin{aligned}\Theta_0\ &=\ \mathop{\mathbf{E}}_{u,v,w}\sum_{\substack{\mathrm{rank}(\alpha_1+\alpha_2),\mathrm{rank}(\beta_1+\beta_2)\le k\\ \pi(\alpha_1+\alpha_2)=\sigma(\beta_1+\beta_2)\\ \nu(\beta_1+\beta_2)=0}}Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)\\ \Theta_1\ &=\ \mathop{\mathbf{E}}_{u,v,w}\sum_{\substack{\mathrm{rank}(\alpha_1+\alpha_2),\mathrm{rank}(\beta_1+\beta_2)\le k\\ \pi(\alpha_1+\alpha_2)=\sigma(\beta_1+\beta_2)\\ \nu(\beta_1+\beta_2)=1}}Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)\\ \Theta_0\ &=\ \mathop{\mathbf{E}}_{u,v,w}\sum_{\substack{\max\{\mathrm{rank}(\alpha_1+\alpha_2),\mathrm{rank}(\beta_1+\beta_2)\}>k\\ \pi(\alpha_1+\alpha_2)=\sigma(\beta_1+\beta_2)}}Term_{u,v,w}(\alpha_1,\alpha_2,\beta_1,\beta_2)\,.\end{aligned}$$

We first lower-bound Θ_0. Note that all terms in Θ_0 are positive. Consider the term corresponding to $\alpha_1=\alpha_2=\beta_1=\beta_2=0$. We have

$$\mathop{\mathbf{E}}_{u,v,w}\hat{f}^4_{v,0}\hat{f}^4_{w,0}=\mathop{\mathbf{E}}_{u}\left(\mathop{\mathbf{E}}_{v}\hat{f}^4_{v,0}\right)^2\ge\left(\mathop{\mathbf{E}}_{u,v}\hat{f}_{v,0}\right)^8\ge s^8\,.$$

Therefore $\Theta_0\ge s^8$.

For Θ_1, we have the following upper-bound

$$|\Theta_1|\le\mathop{\mathbf{E}}_{u,v,w}\sum_{\substack{\mathrm{rank}(\alpha_1+\alpha_2),\mathrm{rank}(\beta_1+\beta_2)\le k\\ \pi(\alpha_1+\alpha_2)=\sigma(\beta_1+\beta_2)\\ \nu(\beta_1+\beta_2)=1}}\hat{f}^2_{v,\alpha_1}\hat{f}^2_{v,\alpha_2}\hat{f}^2_{w,\beta_1}\hat{f}^2_{w,\beta_2}\,.\qquad(8.2)$$

Consider the following randomized labeling strategy for vertices in $u\in U$ and $v\in V$: for $v\in V$, pick (β_1,β_2) with probability $\hat{f}^2_{v,\beta_1}\hat{f}^2_{v,\beta_2}$ and set its label to

$\beta_1 + \beta_2$; for $u \in U$, pick a random neighbor v, and choose (α_1, α_2) with probability $\widehat{f}^2_{v,\alpha_1}\widehat{f}^2_{v,\alpha_2}$ and set its label to $\pi(\alpha_1 + \alpha_2)$. Due to folding, we have that β_1 and β_2 both satisfies the homogeneous linear constraints associated with v, and so does $\beta_1 + \beta_2$. Therefore the right hand side of (8.2) gives the probability that a random edge of the Label Cover is satisfied by this labeling. Thus $|\Theta_1| \le \delta$.

For Θ_2, note that if $\mathrm{rank}(\alpha) > k$, then for any fixed b, $\Pr_x[\alpha x = b] \le 1/2^{k+1}$. Therefore, for any fixed choice of u, v, w, all terms in Θ_2 have absolute value at most $1/2^{k/2+1}$. Combined with Parseval's identity, we conclude that $|\Theta_2| \le 1/2^{k/2+1}$. □

Therefore, any independent set in H has fractional size at most $2^{-\log^b n/32}$, and therefore the chromatic number of H is at least

$$2^{\log^b n/32} = \exp((\log N)^{1/(4-o(1))}).$$

Part IV

Epilogue

故知止其所不知 至矣

—— 庄子 · 齐物论

Chapter 9

Conclusions and Future Work

In this thesis, we proved new hardness results for $\text{Gap}_{1,s}$-CSP, GraphColoring and HypergraphColoring. A common theme of all these results is the construction of Probabilistically Checkable Proofs (PCPs) with perfect completeness. In all results demonstrated in the thesis, we are given certain restrictions on the PCP in terms of alphabet size, proof size, number of queries allowed and the types of verification the verifier could perform, and the goal in all these cases is to build a PCP for E3-Sat with as good soundness as possible.

In Chapter 5, the PCPs are written in Boolean alphabet, and the focus is to get the best soundness for a given number of queries.

As for the results for GraphColoring and HypergraphColoring, the main constraint is that the only type of verification allowed is to read some symbols from a given proof and check if they are not all equal. In Chapter 7, we are aiming for a 2-query PCP with constant alphabet size. Finally, in Chapter 8, the alphabet size is restricted to 2, and the number of queries is restricted to some small constant, in our case, 8.

One aspect in which our HypergraphColoring result is different from the other two is that the PCP for HypergraphColoring has soundness that decreases as the size of the input E3-Sat instance grows, whereas the soundness parameters are constants for the hardness of $\text{Gap}_{1,s}$-CSP and GraphColoring. This stronger soundness does come at a cost of having a super-polynomial-size PCP rather than polynomial size.

In terms of techniques, we demonstrated in Chapter 5 the direct sum technique, although initially used by Siu On Chan in [24] to construct PCP with only *almost perfect* completeness, can nevertheless be adapted and applied to designing PCPs with *perfect* completeness. In Chapter 7, we showed that the ideas in [67] can be extended to the direct sum PCP setting, leading to an exponential improvement in the inapproximability of GraphColoring. In Chapter 8, we presented a different way of proving hardness of a special kind of Label-Cover, first studied by Khot and Saket [77]. We also generalized some of the tools for analyzing Low-Degree-

Long-Code that might become useful in other settings.

PCPs are fascinating objects. Over the past two decades, the study of PCPs has led to the development of many mathematical tools, some of which we reviewed earlier in this thesis. For many combinations of parameters, optimal constructions of PCPs are now known. However, constructing optimal PCPs with perfect completeness remains challenging in many cases. Below, we consider some questions most closely related to the topics of this thesis.

Problem 9.1 (Hardness of $\text{Gap}_{1,s}$-k-CSP)**.** *What is the smallest s for a given k, for which $\text{Gap}_{1,s}$-k-CSP is* NP*-hard?*

Chapter 5 gives $s = 2^{\widetilde{O}(k^{1/3})}/2^k$. The best algorithm works for $s = O(k)/2^k$, and this is quite likely optimal. As discussed in detail in Chapter 5, the NP-hardness for $\text{Gap}_{1-\varepsilon,s}$-$k$-CSP is settled by Siu On Chan [24] with $s = O(k)/2^k$. An interesting result in this direction is by Tamaki and Yoshida [99]. They gave a k-query non-adaptive Long-Code test with perfect completeness and soundness $O(k)/2^k$. Given a Long-Code test, it is now a standard reduction from Unique-Games-hardness to hardness of CSP. This reduction does not work for proving hardness of Gap_s-CSP because Unique-Games does not have perfect completeness. Understanding the obstacles in converting Tamaki and Yoshida's test into a CSP hardness result may be a good starting point.

Problem 9.2 (New algorithms for $\text{Gap}_{1,s}$-CSP)**.** *Is it true that for any predicate $P \subseteq \text{E}k$-Lin, there exists constant $\varepsilon > 0$ such that $\text{Gap}_{\rho(P)+\varepsilon}$-$P$ is in* P*?*

As we discussed in Part II, it is easy to find a satisfying assignment for CSP instances that are conjunctions of linear equations using Gaussian Elimination. In Chapter 4, we proved that adding any extra accepting assignment to the Ek-Lin predicate gives a predicate that is approximation resistant even on satisfiable instances. One may ask what happens if we remove some accepting assignments from Ek-Lin, resulting in some predicate $P \subsetneq \text{E}k$-Lin.

For predicates obtained by removing a single accepting assignment from Ek-Lin, we can easily generalize the algorithm by Zwick [106], such that on satisfiable instances, the algorithm returns an assignment that satisfies at least a 3/4 fraction of the constraints, whereas the density of the predicate is strictly less than 1/2. The algorithm is based on Gaussian Elimination.

Interestingly, for some classes of predicates, Gaussian Elimination is not the only tool. Consider 1-in-k-Sat, where an input is accepted if exactly one of the input bits is 1. Guruswami and Trevisan [48] gave a factor $1/e$ approximation algorithm for satisfiable 1-in-k-Sat instances using a natural Linear Programming relaxation. It is not clear if one can do better with algorithms based on Gaussian Elimination.

Problem 9.3 (Better uniformity for hypergraph coloring hardness)**.** *Prove strong hardness for coloring 2-colorable l-uniform hypergraphs, for $3 \leq l < 8$.*

The hardness in Chapter 8 applies to 2-colorable 8-uniform hypergraph. It is therefore natural to ask whether the uniformity can be improved.

Two previous works might be relevant. In [33], Dinur, Regev and Smyth proved constant-factor hardness for 2-colorable 3-uniform hypergraphs. Their construction encodes labelings by 2-colorings of Kneser graphs instead of the usual Long-Code. It is also one of the few intriguing examples where in the soundness case, the output hypergraphs have large independent sets, even though they are not colorable with a small number of colors.

Another related result is the hardness of 3-colorable 3-uniform hypergraphs [68]. Khot used Smooth-Label-Cover and Long-Code in his construction, and the Long-Code part was replaced with Low-Degree-Long-Code recently in [42], giving a hardness of $(\log n)^{\Omega(1/\log\log\log n)}$. Generalizing the result in Chapter 8 to 3-coloring of hypergraphs is thus a natural question.

Problem 9.4 (Construction of Label-Cover with matrix labels)**.** *Are there more efficient ways to construct* Label-Cover *with matrix labels with similar guarantees as in Theorem 8.12?*

Label-Cover instances with matrix labels are used in Chapter 8 to prove hypergraph coloring hardness results. The construction in Chapter 8 is not completely satisfactory. In particular, we lost a factor 2 in the power of $(\log n)^b$ going from Theorem 8.11 to Theorem 8.12. Improving the construction here may lead to another improvement in the hardness of hypergraph coloring.

Such improvement does not seem to be easy by following our current approach. In Chapter 8, the way we construct Label-Cover with matrix labels is similar to that in [77], where we first prove Gap-CSP with superposition hardness, and then convert it into hardness for Label-Cover with matrix labels. The loss in our current construction comes from the difficulty of ensuring consistency on the edges for all $\sim (\log n)^b$ labelings.

Problem 9.5 (Construction of Label-Cover with strong soundness/label-size tradeoffs)**.** *Prove* Label-Cover *hardness with soundness ε using as small a label set as possible.*

For Label-Cover instances with label size R, a random assignment satisfies a $1/R$ fraction of the edges. Therefore, if we want to have NP-hard Label-Cover problems with soundness ε, the label size needs to be at least $1/\varepsilon$. The construction from Parallel Repetition as given by Theorem 3.10 has label size $1/\varepsilon^c$ for some constant c. The constant c might be quite large. This could be significant in many applications because when using Long-Code together with such Label-Cover, the constant c has a significant impact on the size of the output. A smaller c gives a more efficient reduction and may even give improved hardness result for some problems such as GraphColoring following the approach in Chapter 7. For more examples, we refer to the discussion in [74]. We remark that the focus here is incomparable to that of the Sliding Scale Conjecture [17] or the Projection

Game Conjecture [84]. Roughly speaking, the Sliding Scale Conjecture and the Projection Game Conjecture ask for Label-Cover hardness with soundness ε, instance size poly(n) and $1/\varepsilon^m$, where ε is any $\varepsilon \geq 1/n^c$ for some constant $c > 0$. For applications such as those in Part III, the constant m may have a significant impact on the size of the output instance and thus affect the hardness ratio we get.

For Label-Cover with imperfect completeness, Khot and Safra gave a construction that achieves soundness $1/q$ with label size $O(q^6)$ in [74]. In [24], Chan gave a construction that achieves soundness $1/q$ with label size $O(q^2)$, although the size of the instance in Chan's reduction is much worse than [74].

Problem 9.6 (Efficient construction of Smooth-Label-Cover)**.** *Are there any alternative ways to construct* Smooth-Label-Cover*, other than that presented in the proof of Theorem 3.12?*

We used Smooth-Label-Cover to obtain inapproximability result in Chapter 5. It is also useful in applications such as hypergraph coloring [68].

Most constructions of Smooth-Label-Cover are similar to that presented in Theorem 3.12. One major shortcoming of this construction is that there is a significant blow-up in label size, making it unsuitable for certain hardness of approximation application.

The Label-Cover in [77] has very good smoothness property and is obtained in a very different way. The construction uses low degree polynomials over large fields, and smoothness follows directly from Schwartz-Zippel lemma. Constructions using low degree polynomials are usually not as straightforward as the one in Theorem 3.12, but it could be a good alternative for efficient construction of Smooth-Label-Cover.

Bibliography

[1] Noga Alon, Pierre Kelsen, Sanjeev Mahajan, and Ramesh Hariharan. Approximate hypergraph coloring. *Nord. J. Comput.*, 3(4):425–439, 1996. 6.2

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. 3

[3] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proc. 51st Ann. IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 563–572. IEEE Comp. Soc. Press, 2010. 3.3.1

[4] Sanjeev Arora and Eden Chlamtáč. New approximation guarantee for chromatic number. In *Proc. 38th Ann. ACM Symposium on Theory of Computing (STOC'06)*, pages 215–224. ACM Press, 2006. 6.1

[5] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In *Proc. 40th Ann. ACM Symposium on Theory of Computing (STOC'08)*, pages 21–28. ACM Press, 2008. 3.3.1

[6] 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, 1998. 3.8, 8.1

[7] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. 3.8, 8.1

[8] Per Austrin. Balanced max 2-sat might not be the hardest. In *Proc. 39th Ann. ACM Symposium on Theory of Computing (STOC'07)*, pages 189–197. ACM Press, 2007. 3.3.1

[9] Per Austrin and Johan Håstad. On the usefulness of predicates. *ACM Trans. Computation Theory*, 5(1):1, 2013. 4.1, 5.3

[10] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Comput. Complexity*, 18(2):249–271, 2009. 2.4.4, 3.2, 3.3.1

[11] László Babai. Trading group theory for randomness. In *Proc. 17th Ann. ACM Symposium on Theory of Computing (STOC'85)*, pages 421–429. ACM Press, 1985. 1.3

[12] Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proc. 44th Ann. ACM Symposium on Theory of Computing (STOC'12)*, pages 307–326. ACM Press, 2012. 3.3.1

[13] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. In *Proc. 53rd Ann. IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 370–379. IEEE Comp. Soc. Press, 2012. 3.3.1, 6.2, 8.1

[14] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:59, 2014. 3.3.1

[15] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Trans. Information Theory*, 42(6):1781–1795, 1996. 2.5, 2.5

[16] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998. 1.3, 2.5

[17] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proc. 25th Ann. ACM Symposium on Theory of Computing (STOC'93)*, pages 294–304. ACM Press, 1993. 9

[18] Bonnie Berger and John Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(3):459–466, 1990. 6.1

[19] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *Proc. 51st Ann. IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 488–497. IEEE Comp. Soc. Press, 2010. 6.2

[20] Avrim Blum and David R. Karger. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Inf. Process. Lett.*, 61(1):49–53, 1997. 6.1

[21] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549 – 595, 1993. 2.5

[22] Aline Bonami. Étude des coefficients de fourier des fonctions de $l^p(g)$. *Annales de l'institut Fourier*, 20(2):335–402, 1970. 2.30

[23] Siu On Chan. Approximation resistance from pairwise independent subgroups. *Electron. Colloq. on Comput. Complexity (ECCC)*, 19:110, 2012. 5.1

[24] Siu On Chan. Approximation resistance from pairwise independent subgroups. In *Proc. 45th Ann. ACM Symposium on Theory of Computing (STOC'13)*, pages 447–456. ACM Press, 2013. 2.4.4, 3.2, 5, 5, 5.1, 5.2, 5.3, 6, 7, 7, 9, 9, 9

[25] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Trans. Algorithms*, 5(3):32:1–32:14, 2009. 3.2, 5

[26] Hui Chen and Alan M. Frieze. Coloring bipartite hypergraphs. In *Integer Programming and Combinatorial Optimization, 5th International IPCO Conference, Vancouver, British Columbia, Canada, June 3–5, 1996, Proceedings*, pages 345–358, 1996. 6.2

[27] Eden Chlamtáč. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Proc. 48th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 691–701. IEEE Comp. Soc. Press, 2007. 6.1

[28] Andrea E. F. Clementi and Luca Trevisan. Improved non-approximability results for minimum vertex cover with density constraints. *Theor. Comput. Sci.*, 225(1-2):113–128, 1999. 7.1

[29] Irit Dinur and Venkatesan Guruswami. PCPs via low-degree long code and hardness for constrained hypergraph coloring. In *Proc. 54th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'13)*, pages 340–349. IEEE Comp. Soc. Press, 2013. 2.3, 2.15, 2.3, 2.3, 2.3, 2.5, 2.50, 2.51, 2.5, 2.53, 2.54, 2.55, 6.2, 8.1

[30] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Comput.*, 34(5):1129–1146, 2005. 3.3

[31] Irit Dinur, Subhash Khot, Will Perkins, and Muli Safra. Hardness of finding independent sets in almost 3-colorable graphs. In *Proc. 51st Ann. IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 212–221. IEEE Comp. Soc. Press, 2010. 7

[32] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009. 3.3, 3.3.1, 6.1

[33] Irit Dinur, Oded Regev, and Clifford D. Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005. 6.2, 9

[34] Irit Dinur and Igor Shinkar. On the conditional hardness of coloring a 4-colorable graph with super-constant number of colors. In *Proc. 13th Internat. Conference on Approximation & 14th Internat. Conference on Randomization, and Combinatorial Optimization: Algorithms and Techniques (APPROX/RANDOM'10)*, pages 138–151. Springer, 2010. 6.1

[35] Bradley Efron and Charles Stein. The jackknife estimate of variance. *Ann. Stat.*, 9(3):586–596, 1981. 2.23

[36] 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. 3.2, 3.3, 3.3, 5, 5, 5.2, 5.2

[37] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996. 7.1

[38] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41 (6):1558–1590, 2012. 5

[39] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP*-Completeness*. W. H. Freeman, 1979. 6.1

[40] Oded Goldreich. Short locally testable codes and proofs: A survey in two parts. In Oded Goldreich, editor, *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 65–104. Springer Berlin Heidelberg, 2010. 2.5

[41] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. 1.3

[42] Venkatesan Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. Super-polylogarithmic hypergraph coloring hardness via low-degree long codes. In *Proc. 46th Ann. ACM Symposium on Theory of Computing (STOC'14)*, pages 614–623. ACM Press, 2014. 6.2, 8.1, 9

[43] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering CSP is approximation resistant. *SIAM J. Comput.*, 40(3): 878–914, 2011. 3.3.1

[44] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. *SIAM J. Comput.*, 31(6):1663–1686, 2002. 6.2

[45] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM J. Discrete Math.*, 18(1):30–40, 2004. 6.1

[46] Venkatesan Guruswami, Prasad Raghavendra, Rishi Saket, and Yi Wu. Bypassing UGC from some optimal geometric inapproximability results. In *Proc. 23rd Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'12)*, pages 699–717. SIAM, 2012. 5

[47] Venkatesan Guruswami and Ali Kemal Sinop. The complexity of finding independent sets in bounded degree (hyper)graphs of low chromatic number. In *Proc. 22nd Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'11)*, pages 1615–1626. SIAM, 2011. 6.1

[48] Venkatesan Guruswami and Luca Trevisan. The complexity of making unique choices: Approximating 1-in-k SAT. In *Proc. 8th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems & 9th Internat. Workshop on Randomization and Computation (APPROX/RANDOM'05)*, pages 99–110. Springer, 2005. 9

[49] Gustav Hast. Beating a random assignment. In *Proc. 9th Internat. Workshop on Randomization and Computation (RANDOM'05)*, pages 134–145. Springer, 2005. 3.2

[50] Gustav Hast. *Beating a Random Assignment: Approximating Constraint Satisfaction Problems*. PhD thesis, KTH Royal Institute of Technology, 2005. 3.2

[51] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. 2.5, 3.1.2, 3.2, 5

[52] Johan Håstad. Every 2-CSP allows nontrivial approximation. *Comput. Complexity*, 17(4):549–566, 2008. 3.2

[53] Johan Håstad. On the approximation resistance of a random predicate. *Comput. Complexity*, 18(3):413–434, 2009. 3.2, 3.3.1

[54] Johan Håstad. On the NP-hardness of Max-Not-2. *SIAM J. Comput.*, 43(1):179–193, 2014. 4, 5, 5.3.2

[55] Johan Håstad and Subhash Khot. Query efficient PCPs with perfect completeness. *Theory of Computing*, 1(7):119–148, 2005. 1.4, 1, 5, 7, 8.3

[56] Martin Hilbert and Priscila López. The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011. 1.1, 2

[57] Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(8):141–172, 2009. 3.3

[58] Jonas Holmerin. Vertex cover on 4-regular hyper-graphs is hard to approximate within $2-\varepsilon$. In *Proc. 34th Ann. ACM Symposium on Theory of Computing (STOC'02)*, pages 544–552. ACM Press, 2002. 6.2

[59] Sangxia Huang. Approximation resistance on satisfiable instances for predicates strictly dominating parity. *Electron. Colloq. on Comput. Complexity (ECCC)*, 19:40, 2012. 1.4, 5

[60] Sangxia Huang. Approximation resistance on satisfiable instances for predicates with few accepting inputs. In *Proc. 45th Ann. ACM Symposium on Theory of Computing (STOC'13)*, pages 457–466. ACM Press, 2013. 1.4

[61] Sangxia Huang. Improved hardness of approximating chromatic number. In *Proc. 16th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems & 17th Internat. Workshop on Randomization and Computation (APPROX/RANDOM'13)*, pages 233–243. Springer, 2013. 1.4, 6.1

[62] Sangxia Huang. Approximation resistance on satisfiable instances for sparse predicates. *Theory of Computing*, 10(14):359–388, 2014. 1.4

[63] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. 6.1

[64] Ken-ichi Kawarabayashi and Mikkel Thorup. Combinatorial coloring of 3-colorable graphs. In *Proc. 53rd Ann. IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 68–75. IEEE Comp. Soc. Press, 2012. 6.1

[65] Ken-ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with $o(n^{1/5})$ colors. In *Proc. 31st Symp. Theoretical Aspects of Comp. Sci. (STACS'14)*, pages 458–469. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. 6.1

[66] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000. 6.1

[67] Subhash Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *Proc. 42nd Ann. IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 600–609. IEEE Comp. Soc. Press, 2001. 6.1, 7, 7, 7.1, 9

[68] Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Proc. 43rd Ann. IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 23–32. IEEE Comp. Soc. Press, 2002. 3.3, 5, 6.2, 9, 9

[69] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th Ann. ACM Symposium on Theory of Computing (STOC'02)*, pages 767–775. ACM Press, 2002. 3.2, 3.3.1, 3.19, 3.21

[70] Subhash Khot. On the unique games conjecture (invited survey). In *Proc. 25th IEEE Conf. on Computational Complexity (CCC'10)*, pages 99–121. IEEE Comp. Soc. Press, 2010. 3.3.1

[71] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. 2.4.4, 3.3.1

[72] Subhash Khot and Dana Moshkovitz. NP-hardness of approximately solving linear equations over reals. *SIAM J. Comput.*, 42(3):752–791, 2013. 5

[73] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. 3.3.1

[74] Subhash Khot and Muli Safra. A two-prover one-round game with strong soundness. *Theory of Computing*, 9(28):863–887, 2013. 9

[75] Subhash Khot and Rishi Saket. SDP integrality gaps with local l_1-embeddability. In *Proc. 50th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'09)*, pages 565–574. IEEE Comp. Soc. Press, 2009. 3.3.1

[76] Subhash Khot and Rishi Saket. Hardness of finding independent sets in almost q-colorable graphs. In *Proc. 53rd Ann. IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 380–389. IEEE Comp. Soc. Press, 2012. 7

[77] Subhash Khot and Rishi Saket. Hardness of coloring 2-colorable 12-uniform hypergraphs with $\exp(log^{\Omega(1)}n)$ colors. In *Proc. 55th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'14)*, pages 206–215. IEEE Comp. Soc. Press, 2014. 6.2, 8.1, 8.2, 8.2, 8.4, 9, 9, 9

[78] Subhash Khot and Rishi Saket. Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proc. 25th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'14)*, pages 1607–1625. SIAM, 2014. 6.2

[79] Subhash Khot, Madhur Tulsiani, and Pratik Worah. A characterization of strong approximation resistance. In *Proc. 46th Ann. ACM Symposium on Theory of Computing (STOC'14)*, pages 634–643. ACM Press, 2014. 3.3.1

[80] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into l_1. In *Proc. 46th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 53–62. IEEE Comp. Soc. Press, 2005. 3.3.1

[81] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into l_1. *J. ACM*, 62(1):8, 2015. 3.3.1

[82] Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *J. ACM*, 41(1):99–113, 2001. 6.2

[83] Konstantin Makarychev and Yury Makarychev. Approximation algorithm for non-boolean max-k-csp. *Theory of Computing*, 10(13):341–358, 2014. 3.2

[84] Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory of Computing*, 11(826):501–515, 2015. 9

[85] Elchanan Mossel. Gaussian bounds for noise correlation of functions and tight analysis of long codes. In *Proc. 49th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'08)*, pages 156–165. IEEE Comp. Soc. Press, 2008. 2.4.3, 2.35, 2.4.3, 2.36

[86] Elchanan Mossel. Gaussian bounds for noise correlation of functions. *Geom. Funct. Anal.*, 19(6):1713–1756, 2010. 2.2, 2.5, 2.2, 2.6, 2.2, 2.7, 2.23, 2.4.4, 5, 5.3.3, 5.16

[87] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *Ann. of Math.*, 171(1):295–341, 2010. 2.4.4, 5

[88] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 2.4, 2.4.4

[89] Ryan O'Donnell and John Wright. A new point of NP-hardness for unique games. In *Proc. 44th Ann. ACM Symposium on Theory of Computing (STOC'12)*, pages 289–306. ACM Press, 2012. 2.4.4, 5

[90] Ryan O'Donnell and Yi Wu. Conditional hardness for satisfiable 3-CSPs. In *Proc. 41st Ann. ACM Symposium on Theory of Computing (STOC'09)*, pages 493–502. ACM Press, 2009. 2.4.2, 2.4.2, 2.31, 3.3.1, 4, 4.2, 4.2, 4.2, 4.2, 4.3, 4.4, 4.15, 5

[91] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th Ann. ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254. ACM Press, 2008. 2.4.4, 3.3.1

[92] Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011. 3.3, 3.9, 8.4

[93] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. 3.3, 8.4

[94] Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2 – 44, 1997. 1

[95] Rishi Saket. Hardness of finding independent sets in 2-colorable hypergraphs and of satisfiable CSPs. In *Proc. 29th IEEE Conf. on Computational Complexity (CCC'14)*, pages 78–89. IEEE Comp. Soc. Press, 2012. 6.2

[96] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proc. 32nd Ann. ACM Symposium on Theory of Computing (STOC'00)*, pages 191–199. ACM Press, 2000. 3.2, 5, 5.2, 7

[97] Alex Samorodnitsky and Luca Trevisan. Gowers uniformity, influence of variables, and PCPs. *SIAM J. Comput.*, 39(1):323–360, 2009. 3.2, 5

[98] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th Ann. ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226. ACM Press, 1978. 3.1.1

[99] Suguru Tamaki and Yuichi Yoshida. A query efficient non-adaptive long code test with perfect completeness. *Random Structures & Algorithms*, 45(4):703–723, 2014. 9

[100] Linqing Tang. Conditional hardness of approximating satisfiable Max 3CSP-q. In *Proc. 20th Internat. Symp. Algorithms and Computation (ISAAC'09)*, pages 923–932. Springer, 2009. 5

[101] Luca Trevisan. Approximating satisfiable satisfiability problems. *Algorithmica*, 28(1):145–172, 2000. 3.2

[102] Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proc. 33rd Ann. ACM Symposium on Theory of Computing (STOC'01)*, pages 453–461. ACM Press, 2001. 7, 7.1, 7.4, 7.1

[103] Girish Varma. A note on reducing uniformity in khot-saket hypergraph coloring hardness reductions. *CoRR*, abs/1408.0262, 2014. 6.2, 8.5

[104] Cenny Wenner. Circumventing d-to-1 for approximation resistance of satisfiable predicates strictly containing parity of width at least four. *Theory of Computing*, 9(23):703–757, 2013. 2.8, 2.4.2, 2.33, 2.4.4, 4, 5, 5, 5.3, 5.3.2, 5.3.2, 5.3.2, 5.3.3, 5.15

[105] Avi Wigderson. A new approximate graph coloring algorithm. In *Proc. 14th Ann. ACM Symposium on Theory of Computing (STOC'82)*, pages 325–329. ACM Press, 1982. 6.1

[106] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'98)*, pages 201–210. SIAM, 1998. 3.2, 5, 9