New Spectral Techniques in Algorithms, Combinatorics, and Coding Theory: The Kikuchi Matrix Method

Peter Manohar

CMU-CS-24-142
August 2024

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Venkatesan Guruswami, Co-Chair (UC Berkeley)
Pravesh K. Kothari, Co-Chair (IAS & Princeton University)
Ryan O'Donnell
Uriel Feige (Weizmann Institute)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by: the National Science Foundation under the 2019 GRFP Fellowship program; the National Science Foundation under award numbers CCF1563742, CCF1908125, CCF2211971 and CCF1814603; the David and Lucille Packard Foundation under award number 200529094A; the ARCS Foundation; and a Cylab Presidential Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Spectral Methods, Constraint Satisfaction Problems, Locally Decodable Codes

To my wife Magdalen

Abstract

In this thesis, we present a new method to solve algorithmic and combinatorial problems by (1) reducing them to bounding the maximum, over $x \in \{-1,1\}^n$, of homogeneous degree-q multilinear polynomials, and then (2) bounding the maximum value attained by these polynomials by analyzing the spectral properties of appropriately chosen induced subgraphs of Cayley graphs on the hypercube (and related variants) called "Kikuchi matrices".

We will present the following applications of this method.

(1) Designing algorithms for refuting/solving semirandom and smoothed instances of constraint satisfaction problems;

(2) Proving Feige's conjectured hypergraph Moore bound on the extremal girth vs. density trade-off for hypergraphs;

(3) Proving a cubic lower bound for 3-query locally decodable codes and an exponential lower bound for 3-query locally correctable codes.

Acknowledgments

First and foremost, I would like to thank my wife Magdalen for all her invaluable support throughout the past years.

I also want to thank my advisors, Venkatesan Guruswami and Pravesh K. Kothari, for their mentorship and support. I would especially like to thank Pravesh for being a constant source of optimism and motivation, and for helping me prove [KM24a] as a wedding present! I also would like to thank the rest of my thesis committee, Ryan O'Donnell and Uriel Feige, for attending my thesis proposal and defense, and for providing suggestions that improved this thesis.

I am very grateful to Alessandro Chiesa, who was responsible for introducing me to research many years ago back in 2015, when I was an undergraduate at UC Berkeley. My talks, writing, and research would not be where they are today without your diligent mentorship. I also want to thank Yi-Ren Ng for the time I did research in his computer graphics lab, even though it was rather brief. I would also like to thank Luca Trevisan, who sadly passed away too soon just a few months ago, for his mentorship and guidance, as well as his amazing blog, which has been a great source of ideas and inspiration these past years.

I also want to thank Madhur Tulsiani, as well as Yury Makarychev and Siddharth Bhandari, for their mentorship during Summer 2023 when I was an intern at TTIC.

I would also like to thank my great collaborators, as well as my many friends and outstanding members of the CS Theory community whom I've had the pleasure to interact with these past years: Omar Alrabiah, Mitali Bafna, Ainesh Bakshi, Guy Blanc, Jun-Ting (Tim) Hsieh, Max Hopkins, Rahul Ilango, Siqi Liu, Sidhanth Mohanty, Jonathan Mosheiff, Shivam Nadimpalli, Pedro Paredes, Kevin Pratt, Nic Resch, João Riberio, Igor Shinkar, Nick Spooner, Shashank Srivastava, Thuy-Duong (June) Vuong, Jeff Xu, Goran Žužić, and many others; the research community wouldn't be the same without you. I also want to especially thank Guy Blanc for all the great times that we spent together at conferences across the years.

I also want to thank the other professors and my friends within the CMU CSD community for making this department a great place to be a PhD student: Anupam Gupta, Ryan O'Donnell, Justine Sherry, and Danny Sleator, as well as my CSD friends Jatin Arora and Brian Hu Zhang. I am especially thankful to Anupam for his immaculate talk advice, which has been shared with countless other students, and for his wise, sage-like presence and guidance (and also for selling me his car!). I am also thankful to Jatin, for the great times we had watching Champions League (even if you are a Barça fan...), and to Brian, for the many hours we spent together watching and discussing Formula 1 and Star Wars together.

I would like to give a big shoutout to Patricia Loring, Pravesh's administrative assistant, as well as the rest of the CMU CSD administrative staff, for handling all of my administrative tasks quickly and efficiently, and for keeping the department running smoothly. You're the best!

I am deeply thankful for my high school friends, Eric Chen and Joy Li. Eric, thanks for the amazing speech you gave at my wedding. Joy, I'm glad you finally got those cats you always wanted; Tako and Vinnie are adorable!

I would also like to thank my friends Sudeep Dasari and Jason Zhang, along with their fiancées, Varsha Venkat and Helen Jiang, and the rest of the RoboFantasy crew, for all the great times we enjoyed 7 hours of commercial-free football while watching me lose my weekly fantasy matchup because Josh Allen "only" put up 35 points.

I am very grateful to my wife's parents, Ted Dobson and Susan Cook, along with her siblings

Blue and Alcuin — it's an honor to be a part of your great family. I would like to especially thank Susan for the hard work she puts in every year to make me feel at home when we visit them in Slovenia for Christmas, and Alcuin for truly endless discussions about Minecraft and for working tirelessly to ensure that my wedding was spotted lanternfly-free!

Finally, I would like to thank my parents, Aneesh and Elizabeth, and my brother, Nathan, for all their love and support, and also for instilling within me a love for science and mathematics. This was where my PhD journey began.

Contents

Chapter 1

Introduction

Spectral methods — understanding eigenvectors, eigenvalues, and related linear algebraic properties — have a rich history in algorithm design, forming the backbone of the field of spectral graph theory [HLW06, Spi19]. For example, spectral expander graphs, a ubiquitous object in theoretical computer science with numerous applications such as the construction of good error-correcting codes [SS94], are graphs whose expansion (a combinatorial quantity) is characterized by the eigenvalues of its adjacency matrix. In the past 30 years, there have been remarkable advances in designing algorithms through the use of spectral methods. Such algorithms typically construct a carefully chosen matrix from the input, and analyze its eigenvectors and eigenvalues to find solutions [AKS98, GK01]. Notable examples of spectral algorithms include algorithms for problems such as max cut [Tre09], graph partitioning [McS01], community detection in networks [Abb18], graph sparsification [SS08], and fast linear equation solving [ST11], the latter of which has led to the recent development of a near-linear time algorithm for maximum flow [CKL+22]. Spectral algorithms are often used in average-case algorithm design, a setting where the input to the algorithm is drawn from a (problem-specific) random distribution. This is because the randomness of the input causes the matrix constructed by the algorithm to be random, and so one can analyze the spectral properties of the matrix (and thereby prove correctness of the algorithm) by using the toolkit of random matrix theory.

Spectral methods arise somewhat naturally in the context of graphs, as one can associate a graph G to its adjacency matrix or Laplacian matrix and analyze their eigenvectors/eigenvalues. This makes such methods rather natural to employ when studying computational problems involving graphs such as clique, or instances of arity 2 constraint satisfaction problems (CSPs) such as 2-SAT or 2-XOR, which have an underlying graph structure. However, when studying a more complex CSP such as 3-SAT (since 2-SAT is in P while 3-SAT is NP-complete), the natural object that arises is a 3-uniform *hypergraph*, rather than a graph, and this makes designing spectral algorithms for such problems comparatively more challenging. For example, one could attempt to design an algorithm by naturally associating a 3-uniform hypergraph with a 3-tensor and then computing its injective tensor norm, but this approach immediately runs into issues, as computing (or even approximating!) the injective tensor norm is a notoriously challenging task (see [Bha19]).

Nonetheless, spectral methods give very simple and beautiful algorithms for many problems. However, eigenvectors and eigenvalues of matrices are notoriously brittle properties: small perturbations to a matrix can change this structure quite substantially. As a result, many spectral

algorithms, in particular algorithms for average-case variants of foundational computational problems like 3-SAT or clique, are quite brittle as well. For the example, of, say, clique, the classic spectral algorithm succeeds with high probability when given a graph drawn from the Erdős-Rényi distribution $G(n, 1/2)$, but the algorithm will fail with high probability if one allows an adversarial addition/deletion of $O(n)$ edges to the graph. One can interpret this brittleness as showing that these algorithms are *overfitting* to the particular choice of input distribution (e.g., $G(n, 1/2)$) and thus fail to generalize to other input distributions, even those that are merely small deviations from the initial choice of distribution that should intuitively not affect the behavior of a "good" algorithm.

Contributions of this thesis. In this thesis, we present a new collection of spectral techniques to solve algorithmic and combinatorial problems over *hypergraphs*. Our techniques give a general method to bound the maximum, over $x \in \{-1, 1\}^n$, of a (problem-dependent) homogeneous degree-q multilinear polynomial f — a very general problem with many applications — by analyzing the spectral norm of a family of appropriately chosen induced subgraphs (and related variants) of weighted Cayley graphs on the hypercube $\{0, 1\}^n$. The techniques that we introduce are *robust* and allow us to obtain good bounds on $\max_{x\in\{-1,1\}^n} f(x)$ even for polynomials sampled from *semirandom* or *smoothed* input distributions: distributions where the sampled polynomial f has a significant amount of adversarial structure. This is unlike typical spectral methods, which are usually brittle and ill-suited to give good bounds in these more adversarial settings.

In more detail, we map the natural q-tensor associated to the polynomial f to a (hierarchy of) matrices where the spectral norm of the ℓ-th level matrix in the hierarchy yields progressively tighter upper bounds on $\max_{x\in\{-1,1\}^n} f(x)$ as ℓ increases. These matrices, first introduced in a work of [WAM19] to design an algorithm for Gaussian tensor PCA, are called "Kikuchi matrices" or the "Kikuchi hierarchy", and hence we call our approach the *Kikuchi matrix method*.

In Chapter 2, we will give a technical overview of Kikuchi matrices and how to use them to certify bounds on $\max_{x\in\{-1,1\}^n} f(x)$; in the course of this overview, we will demonstrate our basic approach along with two key ideas, row bucketing and row pruning, that allow us to construct spectral certificates that bound $\max_{x\in\{-1,1\}^n} f(x)$ even for polynomials f that have significant adversarial structure and correlated randomness. Next, in Chapter 3 we will formally define notation and concepts that we will use in the thesis. The remainder of the thesis is divided up into three parts, where we will discuss the results that we have shown thus far using our "Kikuchi matrix method". The results will be presented and organized as follows.

Part I: Algorithms for Semirandom and Smoothed Constraint Satisfaction Problems:

- Chapter 5: Algorithms for strongly refuting semirandom and smoothed CSPs. This chapter is based on [GKM22, Sections 4–7].
- Chapter 6: Existence of short refutation witnesses for smoothed CSPs below the spectral threshold. This chapter is based on [GKM22, Section 9].
- Chapter 7: Efficient algorithms to solve semirandom planted CSPs. This chapter is based on [GHKM23].

Part II: Extremal Girth vs. Density Trade-Offs for Hypergraphs:

- Chapter 9: A proof of the hypergraph Moore bound. This chapter is based on [GKM22, Section 8].

Part III: Lower Bounds for Locally Decodable and Correctable Codes:

Chapter 11: A near-cubic lower bound for 3-query locally decodable codes. This chapter is based on [AGKM23].

Chapter 12: Exponential lower bounds for 3-query locally correctable codes. This chapter is based on [KM24a, KM24b].

Finally, in Part IV we discuss open problems and directions for future work.

Chapter 2

An Overview of the Method and Key Technical Ideas

In this chapter, we will give a brief overview of Kikuchi matrices and the related spectral methods that we develop in this thesis. The problems that we will discuss here are specific instantiations of the general task of algorithmically certifying a good bound on $\mathrm{val}(f) \coloneqq \max_{x\in\{-1,1\}^n} f(x)$, where $f(x)$ is a homogeneous degree-q multilinear polynomial f in variables $x_1, \ldots, x_n$, i.e., $f(x) = \sum_{C\in\binom{[n]}{q}} b_C x_C$, where $b_C \in \mathbb{R}$ is a coefficient and x_C is the monomial $\prod_{i\in C} x_i$. More formally, we will design an algorithm that is given as input such a polynomial f, and then the algorithm efficiently computes a real number alg-val(f) such that $\mathrm{val}(f) \le \text{alg-val}(f)$ always holds. The goal is to argue that when f is chosen from a (problem-specific) family of distributions, the output alg-val(f) of the algorithm provides a meaningful bound on $\mathrm{val}(f)$. For the purpose of this chapter, we will focus on the case when q is even, i.e., the polynomial f has even degree. This case turns out to be, from a technical standpoint, substantially easier to handle.

Before we delve into the techniques, we will give some motivating examples and state the theorems that we will prove in this chapter.

Example 2.0.1 (Random and Semirandom q-XOR)**.** Let H be an arbitrary q-uniform hypergraph and let $b_C \in \{-1, 1\}$ for each $C \in H$. We think of the collection $(H, \{b_C\}_{C\in H})$ as specifying a q-XOR instance ψ with n variables and $m = |H|$ constraints, i.e., we associate each $C \in H$ and $b_C \in \{-1, 1\}$ with the q-XOR constraint $\prod_{i\in C} x_i = b_C$. Setting $f(x) = \sum_{C\in H} b_C x_C$, we see that for any assignment $x \in \{-1, 1\}^n$ to the variables, $f(x)$ simply computes the number of satisfied constraints minus the number of violated constraints. Hence, $\mathrm{val}(f) = m$ if and only if the instance is satisfiable, and $\mathrm{val}(f) \le \varepsilon m$ implies that at most $\frac{1}{2} + \frac{1}{2}\varepsilon$ fraction of constraints can be simultaneously satisfied.

A q-XOR instance is *random* if H and the b_C's are chosen at random, and it is semirandom if H is arbitrary but the b_C's are still chosen at random. The main technical contribution of Part I of this thesis is an algorithm to certify a bound on $\mathrm{val}(f)$ where f is defined via a semirandom q-XOR instance. This is the task of refutation, or certifying unsatisfiability, as an algorithm that outputs alg-val(f) such that (1) $\mathrm{val}(f) \le \text{alg-val}(f)$ holds for any f, and (2) alg-val$(f) \le \varepsilon m$ with high probability for, e.g., a random q-XOR polynomial f, is an algorithm that refutes (certifies unsatisfiability) of a random q-XOR instance with high probability. In fact, such an algorithm

strongly refutes the random instance, as it shows that only $\frac{1}{2} + \frac{1}{2}\varepsilon$ fraction of the constraints can be satisfied simultaneously.

Refuting semirandom instances of XOR, or of any constraint satisfaction problem more generally, and other related algorithmic tasks is the focus of Part I of this thesis. In this overview, we will prove our result for refuting semirandom q-XOR instances, for the case when q is even.

Theorem 2.0.2 (Refutation algorithm for semirandom q-XOR, even q)**.** *Let q be even. For every integer $\ell \geq q/2$, there is an algorithm $\mathcal{A}$ that takes as input a q-XOR polynomial $f(x) = \sum_{C \in H} b_C x_C$ in n variables $x_1, \ldots, x_n$, specified by a q-uniform hypergraph H with $m = |H|$ hyperedges and "right-hand sides" $b_C \in \{-1, 1\}$, and outputs in $n^{O(\ell)}$-time a value* alg-val$(f) \in [-m, m]$ *with the following two properties:*

(1) val$(f) \leq$ alg-val(f) *for all q-XOR polynomials f;*

(2a) If $m \geq O\left(\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{q/2} \ell \log n\right)$ and the input polynomial f is a random q-XOR polynomial, i.e., H is a random collection of m hyperedges C and each b_C is chosen from $\{-1, 1\}$ uniformly at random, then with high probability over the draw of H and the b_C's, it holds that alg-val$(f) \leq \varepsilon m$.

(2b) If $m \geq O\left(\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{q/2} \ell \log n\right)$ and the input polynomial f is a semirandom q-XOR polynomial, i.e., H is an arbitrary collection of m hyperedges C and each b_C is chosen from $\{-1, 1\}$ uniformly at random, then with high probability over the draw of the b_C's, it holds that alg-val$(f) \leq \varepsilon m$.

We note that Item (2b) above subsumes Item (2a), as it handles a more general case (H is arbitrary rather than random). We have separated out Item (2a) because we will prove Theorem 2.0.2 in two stages; we will first prove Item (2a), before generalizing the proof to handle Item (2b).

Example 2.0.3 (Locally Decodable Codes)**.** A (q, δ, ε)-locally decodable code (LDC) $C\colon \{-1,1\}^k \to \{-1,1\}^n$ is equivalent (Fact 3.3.3) to a collection of q-uniform hypergraph matchings $H_1, \ldots, H_k$,[1] each of size δn, such that for any choice of $b \in \{-1,1\}^k$, the polynomial $f_b(x) := \sum_{i=1}^{k} \sum_{C \in H_i} b_i x_C$ has value val$(f_b) \geq \varepsilon m$, where $m = \sum_{i=1}^{k} |H_i| = k\delta n$. Thus, for a particular choice of n as a function of k, q, δ, and ε, to show that no such locally decodable code exists, i.e., to prove a *lower bound*, it suffices to argue that val$(f_b) < \varepsilon m$ with high probability when $b \leftarrow \{-1,1\}^k$ is chosen at random.

The application of our method to proving lower bounds for locally decodable codes and the related stronger notion of locally correctable codes (LCCs) is the focus of Part III of this thesis. In this overview, we will give a proof of the following lower bounds for (q, δ, ε)-LDCs. We remark that, prior to this thesis, these were the best known lower bounds for LDCs (or LCCs).

Theorem 2.0.4. *Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a code that is (q, δ, ε)-locally decodable, for constant $q \geq 2$. Then, the following hold:*

(1) If q is even, $k \leq n^{1-2/q} O(\log n)/(\varepsilon^4 \delta^2)$, and

(2) If q is odd, $k \leq n^{1-2/(q+1)} O(\log n)/(\varepsilon^4 \delta^2)$.

Examples 2.0.1 and 2.0.3 have been chosen to showcase the core technical contributions of this thesis: the basic approach of our method along with two key ideas, *row bucketing/reweighting* and *row pruning*. We will start by explaining the basic approach of our method in Section 2.1, where we use it to give a simple algorithm to refute random q-XOR instances (Item (2a) of Theorem 2.0.2). Then, in Section 2.2 we will give an algorithm to refute *semirandom* q-XOR instances (Item (2b) of

[1] A hypergraph H_i is a *matching* if its constituent hyperedges are disjoint.

Theorem 2.0.2). Compared to the case of random q-XOR, the new challenge is that a semirandom instance has a worst-case hypergraph H, and overcoming this challenge requires a new technical idea: *row bucketing/reweighting*. Finally, in Section 2.3 we will prove Theorem 2.0.4. Compared to the two previous cases, the new challenge posed by Theorem 2.0.4 is that the coefficients b_C of the polynomial f from an LDC instance are not independent, and handling this issue will require a different key technical idea: *row pruning*.

2.1 The main approach and Kikuchi matrices for even q

The high-level idea of our approach is to bound $\mathrm{val}(f)$ by first expressing the polynomial f as a quadratic form on a matrix A_f, and then using the spectral norm $\|A_f\|_2$ to bound the maximum quadratic form on A_f. As one can compute $\|A_f\|_2$ in $\mathrm{poly}(N)$ time, where N is the size of the matrix A_f, this will yield an algorithm to bound $\mathrm{val}(f)$, e.g., as required in Theorem 2.0.2. For technical reasons, such a matrix is substantially easier to define and analyze when q is even, so we shall restrict ourselves to this case in this overview for simplicity.

The way we shall express f as a matrix A_f is as follows. For every monomial $x_C := \prod_{i \in C} x_i$, we "lift" it to a matrix $A_C \in \mathbb{R}^{N \times N}$ such that for each assignment $x \in \{-1,1\}^n$, there is a vector $x^{\odot \ell} \in \{-1,1\}^N$ where $(x^{\odot \ell})^\top A_C x^{\odot \ell} = D x_C$, for some positive integer D. If we have such a collection of matrices A_C, then we can clearly associate $f = \sum_C b_C x_C$ to the matrix $A_f = \sum_C b_C A_C$. We then have $Df(x) = (x^{\odot \ell})^\top A_f x^{\odot \ell}$ for all $x \in \{-1,1\}^n$ and therefore $D\,\mathrm{val}(f) = Df(x^*) \leq \|x^{*\odot \ell}\|_2^2 \|A_f\|_2 = N\|A_f\|_2$, where x^* is a maximizer.

The matrices A_C are *Kikuchi matrices*, first introduced in the work of [WAM19].

Definition 2.1.1 (Kikuchi matrices for even q)**.** Let $C \subseteq [n]$ be a set of size q, where q is even, and let $\ell \geq q/2$ be an integer. We define the matrix $A_C \in \mathbb{R}^{N \times N}$ as follows. Let $N = \binom{n}{\ell}$ and identify N with subsets $S \subseteq [n]$ of size exactly ℓ. We let $A_C(S,T) = 1$ if $S \oplus T = C$ and 0 otherwise, where $S \oplus T$ denotes the *symmetric difference* of S and T, i.e., $S \oplus T := (S \cup T) \setminus (S \cap T)$.

The integer ℓ in Definition 2.1.1 is a parameter that dictates the size of the matrix. A larger choice of ℓ yields a larger matrix, and hence a slower algorithm (as computing $\|A_f\|_2$ is $\mathrm{poly}(N)$ time, where N is the size of the matrix), but as ℓ grows the spectral norm $\|A_f\|_2$ yields a tighter bound on $\mathrm{val}(f)$ (as implicitly indicated in Theorem 2.0.2).

The matrices defined in Definition 2.1.1 have the following properties.

Proposition 2.1.2. *Let $C \subseteq [n]$ be a set of size q, where q is even, let $\ell \geq q/2$ be an integer, and let A_C be defined as in Definition 2.1.1. Then, the following hold:*

1. *A_C has at most one nonzero entry per row or column, and has* exactly $D = \binom{q}{q/2}\binom{n-q}{\ell-q/2}$ *nonzero entries;*
2. *For any $x \in \{-1,1\}^n$, let $x^{\odot \ell} \in \{-1,1\}^N$ be the vector where the S-th entry is $x_S^{\odot \ell} = \prod_{i \in S} x_i$. Then, $(x^{\odot \ell})^\top A_C x^{\odot \ell} = D x_C$.*

Proof. The fact that A_C has at most one nonzero entry per row or column follows because for a fixed C and a fixed choice of the row S, $A_C(S,T)$ is nonzero if and only if $S \oplus C$ has size exactly ℓ. One can compute the number of nonzero entries by observing that a pair (S,T) satisfies $S \oplus T = C$ if and only if S and T each contain disjoint halves of equal size of the set C, along with some shared set R of size exactly $\ell - q/2$. There are $\binom{q}{q/2}$ ways to split C into disjoint halves, followed by $\binom{n-q}{\ell-q/2}$ ways to choose the set $R \subseteq [n] \setminus C$, which gives us the number of nonzero entries D.

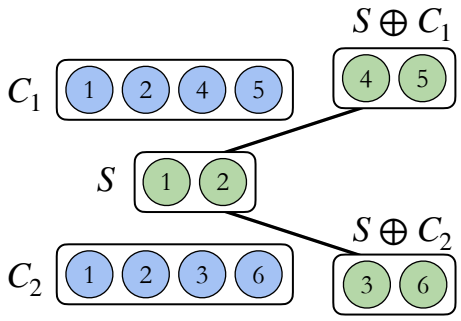

Figure 2.1: A part of the graph from the "basic spectral relaxation" when $q = 4$, or equivalently a Kikuchi graph with $\ell = q/2 = 2$. The vertices S of the graph are in green, and the hyperedges C are in blue.

Notice that here we crucially need that q is even so that we can divide the set C into two halves of equal size.

To prove Item (2), we observe that

$$(x^{\odot \ell})^\top A_C x^{\odot \ell} = \sum_{(S,T):S\oplus T=C} x_S x_T = \sum_{(S,T):S\oplus T=C} \prod_{i\in S\cap T} x_i^2 \prod_{i\in S\oplus T} x_i = \sum_{(S,T):S\oplus T=C} x_C = D x_C \,,$$

where we use that $x_i^2 = 1$ since $x \in \{-1,1\}^n$. □

With Proposition 2.1.2 in hand, we have thus shown that for $A_f := \sum_{C\in H} b_C A_C$, it holds that $D\,\mathrm{val}(f) \le N\|A_f\|_2$, and so if we set $\text{alg-val}(f) := \|A_f\|_2 \cdot N/D$, then $\mathrm{val}(f) \le \text{alg-val}(f)$ holds for all f. Hence, to prove, e.g., Item (2a) in Theorem 2.0.2, it suffices to argue that $\|A_f\|_2 \le \varepsilon m D/N$ with high probability when the hypergraph H is random, the b_C's are chosen independently from $\{-1,1\}$, and $m \ge O\left(\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{q/2} \ell \log n\right)$.

Before we prove Item (2a) in Theorem 2.0.2, we give two interpretations of the Kikuchi matrices defined in Definition 2.1.1.

Kikuchi matrices as generalizations of the "basic spectral relaxation". Definition 2.1.1 arises naturally from the viewpoint of trying to "lift" a polynomial f to a quadratic form on a matrix A_f. The setting of $\ell = q/2$ corresponds to the well-studied setting of the "basic" spectral relaxation: the matrix A_C is indexed by sets S and T of size $q/2$, and we have $A_C(S,T) = 1$ if and only if $S \cup T = C$. When $\ell = q/2$, this matrix is a flattening of the natural q-tensor associated to the polynomial f. The Kikuchi matrices in Definition 2.1.1 give a generalization of this basic matrix to larger and larger matrices, with the hope (that we will prove!) that the larger matrices will yield tighter bounds on $\mathrm{val}(f)$. A rather interesting observation is that when $\ell > q/2$, the matrix A_C is *not* a flattening of the natural q-tensor of the monomial x_C. This viewpoint can be thought of as a "bottom-up" approach, where we view the Kikuchi matrices at level ℓ as a natural generalization of the "basic" matrix at level $q/2$.

Kikuchi matrices as induced subgraphs of Cayley graphs. Definition 2.1.1 arises naturally from the perspective of Cayley graphs on the hypercube. Given a polynomial $f = \sum_{C\in H} b_C x_C$,

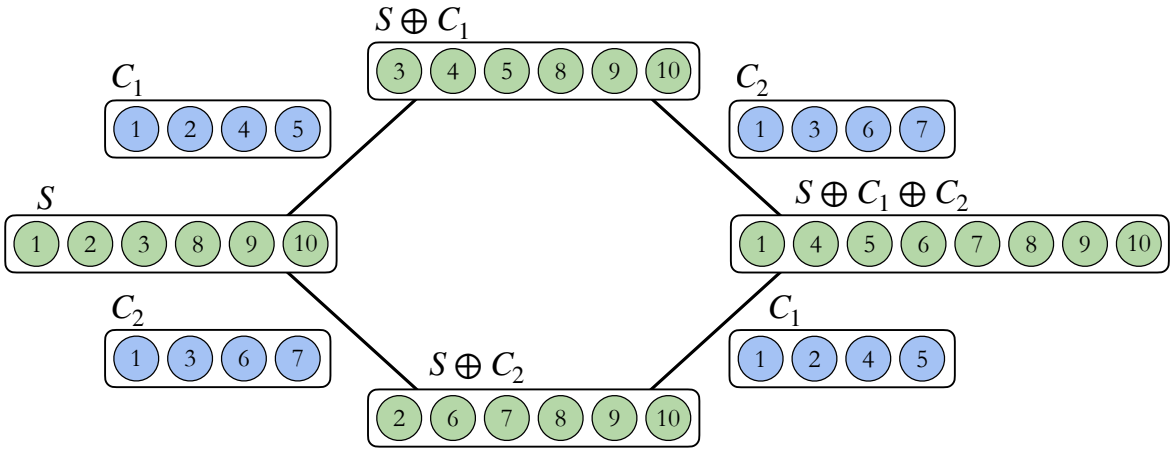

Figure 2.2: A part of the full Cayley graph when $q = 4$. The vertices S of the graph are in green, and the hyperedges C are in blue. In a Cayley graph, any pair C_1, C_2 of hyperedges (generators) along with a vertex S forms a 4-cycle.

one can associate f to a (weighted) Cayley graph on the hypercube $\{0,1\}^n$ as follows. The Cayley graph has vertices $\{0,1\}^n$, with the group operation being addition over $\mathbb{F}_2^n$, and the generators are given by the hypergraph H, i.e., for each $C \in H$ we identify C with its corresponding weight q indicator vector in $\{0,1\}^n$. Equivalently, we can identify the vertices of the graph with subsets $S \subseteq [n]$ (of any size), and we put an edge (S,T) with edge weight b_C if $S \oplus T = C$. Let B denote the adjacency matrix of this Cayley graph, i.e., B is a $2^n \times 2^n$ matrix where $B(S,T) = b_C$ if $S \oplus T = C$ for some $C \in H$, and otherwise $B(S,T) = 0$.

A well-known fact about Cayley graphs over the hypercube is that the eigenvectors of B are the character functions. Namely, for each $x \in \{-1,1\}^n$, the vector $\chi^{(x)} \in \{-1,1\}^{2^n}$, defined as $\chi_S^{(x)} = \prod_{i \in S} x_i$ for each $S \subseteq [n]$, is an eigenvector of B, and one can compute that its corresponding eigenvalue λ_x is simply $f(x)$. Thus, $\|B\|_2 = \max_{x \in \{-1,1\}^n} f(x)$. The problem is that for, e.g., the case of algorithms as in Theorem 2.0.2, computing $\|B\|_2$ requires $2^{O(n)}$ time, as B is a very large matrix, and this is no better than simply computing $f(x)$ for all $x \in \{-1,1\}^n$ via brute force.

A rather naive way to lower the size of the matrix (and thereby lower the runtime) is to simply take an induced subgraph of the full Cayley graph. Indeed, the Kikuchi matrices in Definition 2.1.1 are obtained by restricting the matrix B to the set of vertices $\{S : |S| = \ell\}$. When taking induced subgraphs, there are two potential problems that can arise. The first issue is that the induced subgraph may have no edges in it at all, as it is the induced subgraph is on a very small fraction of all 2^n vertices! In fact, this is precisely the issue that arises in the case when q is odd in Definition 2.1.1. The second issue is that the induced subgraph is no longer a Cayley graph, and in particular it will not have the same nice eigenvector/eigenvalue structure that is present in the matrix B. Thus, it might be the case that the spectral norm of the matrix from the induced subgraph no longer provides an upper bound on $\mathrm{val}(f)$. When q is odd, this is trivially the case as the induced subgraph has no edges. (Note that the spectral norm of the induced subgraph is a *lower bound* on $\|B\|_2$, the spectral norm of the full Cayley graph.) However, as we observed in Proposition 2.1.2, when q is even, neither of these two issues arise.

In contrast to the "bottom-up" viewpoint discussed previously, this viewpoint can be thought

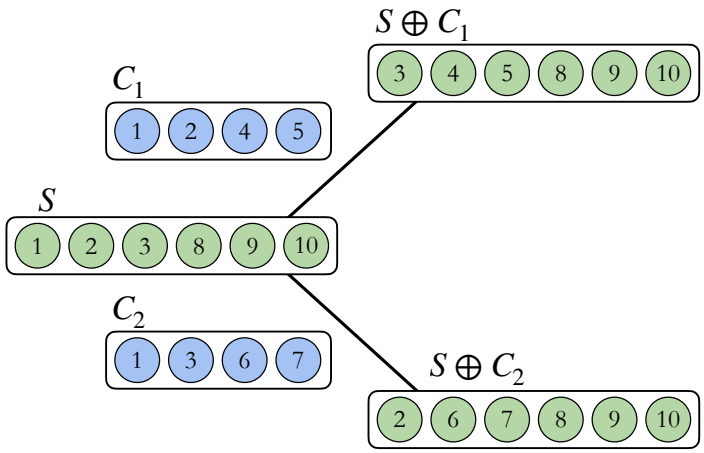

Figure 2.3: A part of the Kikuchi graph at level $\ell = 6$ when $q = 4$. The vertices S of the graph are in green, and the hyperedges C are in blue. Unlike in a Cayley graph, any pair C_1, C_2 of hyperedges (generators) along with a vertex S need not form a 4-cycle in the Kikuchi graph.

of as a "top-down" approach, where we view the Kikuchi matrices at level ℓ as a restriction of the "full" matrix at level n.

One may be wondering why we have defined Kikuchi matrices as the induced subgraph on subsets of size exactly ℓ rather than sets $\leq \ell$, as one could easily do the latter without substantially increasing the size of the matrix. It turns out that defining the matrix using all sets of size $\leq \ell$ does not result in any major differences when q is even,[2] as the contribution from the sets of size ℓ will be the dominant term in the spectral norm, so the matrix with sets of size $\leq \ell$ is essentially "no better" than the matrix with sets of size exactly ℓ as defined in Definition 2.1.1.

With this Cayley graph viewpoint, we can view Kikuchi matrices as a way to transform a q-uniform hypergraph H to a (family of) graphs, one for each choice of the parameter ℓ. An important fact is that a cycle (or even cover) in the hypergraph H — a collection of distinct hyperedges $C_1, \ldots, C_r$ such that $C_1 \oplus \cdots \oplus C_r = \emptyset$ — gives a cycle in the Cayley graph: simply take any vertex S and use the edges from the generators $C_1, \ldots, C_r$. Moreover, any cycle in the Cayley graph corresponds to a collection of (possibly not distinct) hyperedges $C_1, \ldots, C_r$ that form a cycle. This key observation forms the start of the proof of the hypergraph Moore bound in Part II, where we use Kikuchi graphs to show the existence of short cycles in sufficiently dense hypergraphs.

Refuting random q-XOR for q even. Let us now prove Item (2a) in Theorem 2.0.2. The analysis presented here is due to [WAM19], the original work that introduced the Kikuchi matrices in Definition 2.1.1. We are given as input a random q-uniform hypergraph H with $m \geq O\left(\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{q/2} \ell \cdot \log n\right)$ hyperedges with "right-hand sides" $b_C \in \{-1, 1\}$ chosen independently for each $C \in H$. As stated earlier, our goal is to argue that $\|A\|_2 \leq \varepsilon m D/N$, where $A = \sum_{C \in H} b_C A_C$

[2]When q is odd, this is a major difference; the Kikuchi matrix with sets of size ℓ is identically 0 whereas the Kikuchi matrix with sets of size $\leq \ell$ is not and can be used to bound $\mathrm{val}(f)$. However, this matrix achieves, e.g., in the setting of Theorem 2.0.2, a suboptimal bound of $m \gtrsim \frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{\lceil q/2 \rceil} \ell \log n$.

and the matrices A_C are defined in Definition 2.1.1. Because the b_C's are independent and random, the matrix A is the sum of mean 0 independent random matrices. Thus, by Matrix Khintchine (Fact 3.4.2), it follows that with high probability, we have $\|A\|_2 \le O(\sigma\sqrt{\ell \log n})$, where $\sigma^2 = \|\sum_{C\in H} A_C^2\|_2$. So, it remains to compute σ^2.

Because each A_C has at most 1 nonzero entry per row/column (Proposition 2.1.2), it follows that A_C^2 is a diagonal matrix. In fact, $\Upsilon = \sum_{C\in H} A_C^2$ is a diagonal matrix where the S-th diagonal entry is $\Upsilon_S := \{C \in H : |S \cap C| = q/2\}$, as the S-th row of A_C has a nonzero entry if and only if $|S \cap C| = q/2$. We note that by Proposition 2.1.2, $\sum_S \Upsilon_S = mD$, as $\sum_S \Upsilon_S$ is equal to the number of nonzero entries in all the A_C's, and there are m choices of C with each A_C contributing D nonzero entries. The average is simply mD/N where $N = \binom{n}{\ell}$ is the size of the matrices A_C. Now, *because the hypergraph H is random*, it holds that with high probability over S, $\max_S \Upsilon_S \le O(mD/N)$. With this fact,[3] we can now finish the proof.

Indeed, by Matrix Khintchine (Fact 3.4.2), we have shown that with high probability over H and the b_C's, it holds that

$$\|A\|_2 \le O\left(\sqrt{\frac{mD\ell \log n}{N}}\right) \le \varepsilon mD/N\,,$$

where the last inequality follows by (1) standard binomial coefficient estimates (Fact 3.6.1) to show that $D/N \sim \left(\frac{\ell}{n}\right)^{q/2}$, and (2) using that $m \ge O\left(\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{q/2} \ell \cdot \log n\right)$. This finishes the proof of Item (2a) in Theorem 2.0.2, i.e., the case of *random* q-XOR.

Summary: Method Overview

(1) For each monomial $x_C := \prod_{i\in C} x_i$ where $|C| = q$, we define a matrix $A_C \in \mathbb{R}^{N\times N}$, where $N = \binom{n}{\ell}$, and $A_C(S,T) = 1$ if $S \oplus T = C$ and 0 otherwise. The parameter ℓ controls the size of the matrix, and the matrix satisfies

$$x^{\odot\ell\top} A_C x^{\odot\ell} = D x_C$$

for any $x \in \{-1,1\}^n$, where $(x^{\odot\ell})_S = \prod_{i\in S} x_i$ and $D = \binom{q}{q/2}\binom{n-q}{\ell-q/2}$.

(2) We associate a degree-q multilinear polynomial $f(x) = \sum_{C:|C|=q} b_C x_C$ to the matrix $A_f = \sum_{C:|C|=q} b_C x_C$. For every $x \in \{-1,1\}^n$, the matrix satisfies

$$x^{\odot\ell\top} A_f x^{\odot\ell} = D f(x)\,.$$

In particular, $\mathrm{val}(f) = \max_{x\in\{-1,1\}^n} f(x) \le \frac{N}{D}\|A_f\|_2 \lesssim \left(\frac{n}{\ell}\right)^{q/2} \|A_f\|_2$.

2.2 Handling arbitrary hypergraphs with row bucketing

In this section, we will prove Item (2b) in Theorem 2.0.2, i.e., we will give an algorithm to refute *semirandom* instances of *even-arity* q-XOR. Compared to the case of *random* q-XOR with even q

[3] We will not prove this fact here, although it follows from a simple Chernoff bound (see [WAM19, Section F.1.4]). In any case, the statement proven (Item (2a) in Theorem 2.0.2) will be subsumed by Section 2.2.

handled in Section 2.1, the key challenge (and indeed, the only difference) is that we now allow the hypergraph H to be arbitrary. The purpose of this section is to explain the key idea, *row bucketing/row reweighting*,[4] that we use to handle this challenge.

More formally, a semirandom q-XOR instance f is represented as an *arbitrary* q-uniform hypergraph H with *random* right-hand sides $b_C \in \{-1,1\}$ for each $C \in H$. Our goal is to give, for any ℓ, an $n^{O(\ell)}$-time algorithm that will certify that $\mathrm{val}(f) \le \frac{1}{2} + \frac{1}{2}\varepsilon m$ where $m = |H|$ is the number of constraints, provided that $m \ge O((n/\ell)^{q/2}\ell \log n/\varepsilon^2)$.

Let us now conduct a post-mortem of the proof in Section 2.1 of Item (2a) in Theorem 2.0.2 to see where we used the randomness of the hypergraph H. Even after fixing H, the A_C's are independent random matrices, with all the randomness coming from the b_C's. Thus, we can still apply the Matrix Khintchine inequality to obtain the same bound on $\|A\|_2$. The only point in the proof where we used the randomness of the hypergraph H was to establish that $\Upsilon_S = O(mD/N) = O(\ell \log n)$ for every S. So, the proof in Section 2.1 immediately extends to semirandom instances where the instance hypergraph H is such that $\Upsilon_S = O(mD/N)$ for every S.

This bound is delicate: when $\Upsilon_S = \Omega(\ell^2)$, we obtain no non-trivial refutation guarantee and even $\Upsilon_S \sim \ell^{1.1}$ results in a suboptimal trade-off. On the other hand, in arbitrary H, Υ_S can be as large as m (but no larger). Further, this is a "real" issue and not an artifact of a potentially loose spectral norm bound from the Matrix Khintchine inequality: when Υ_S is large, so is the spectral norm of A.

Key observation: only sparse vectors cause large quadratic forms. The key observation is that even though Υ_S can be large, making $\|A\|_2$ large, the "offending" large quadratic forms are induced only by "sparse" vectors, i.e., vectors where the ℓ_2-norm is contributed by a small fraction of the coordinates. On the other hand, we only care about upper bounding quadratic forms of A on vectors $x^{\odot\ell}$ for some $x \in \{-1,1\}^n$, in particular, on vectors where all coordinates are ± 1 and are thus are maximally "non-sparse" or "flat". More formally, in order to certify a bound on $\mathrm{val}(f)$, it suffices for us to bound $\|A\|_{\infty\to 1}$, rather than $\|A\|_2$.

Row bucketing. We can formalize this observation via *row bucketing*. Let $d_0 = mD/N \sim m \cdot (\ell/n)^{q/2}$ be the average value of Υ_S. Let us partition the row indices in $N = \binom{n}{\ell}$ into multiplicatively close buckets $\mathcal{F}_0, \mathcal{F}_1, \cdots, \mathcal{F}_t$ so that for each $i \ge 1$,

$$\mathcal{F}_i = \left\{S \mid 2^{i-1}d_0 < \Upsilon_S \le 2^i d_0\right\} .$$

and $\mathcal{F}_0 = \{S \mid \Upsilon_S \le d_0\}$. Then, since $\Upsilon_S \le m$ and $d_0 \ge 1$ (as $m \ge O((n/\ell)^{q/2} \cdot \ell \log n)$), we can take $t \le \log_2 m$. Further, by Markov's inequality, $|\mathcal{F}_i| \le 2^{-i}\binom{n}{\ell} = 2^{-i}N$. For each $i, j \le t$, let $A_{i,j}$ be the matrix obtained by zeroing out all rows not in $\mathcal{F}_i$ and all columns not in $\mathcal{F}_j$ from the Kikuchi matrix A. Then, $A = \sum_{0\le i,j\le t} A_{i,j}$.

The key observation is the following: while $A_{i,j}$ has nonzero rows and columns where Υ_S is larger by a 2^i (2^j, respectively) factor than the average, we are compensated for this by a reduction in the number of nonzero rows and columns.

Let $y \in \mathbb{R}^N$ be any vector with entries in $\{-1,1\}^N$, and let $y_{\mathcal{F}_i}$ be the vector obtained by zeroing out all coordinates of y that are not indexed by elements of $\mathcal{F}_i$. Then, by Cauchy-Schwarz,

[4] Row reweighting, introduced in the work of [HKM23], is a refined version of the row bucketing method of [GKM22].

we must have:

$$\max_{y\in\{-1,1\}^N} y^\top A_{i,j} y = \max_{y\in\{-1,1\}^N} (y_{\mathcal{F}_i})^\top A_{i,j} (y_{\mathcal{F}_j}) \le \sqrt{|\mathcal{F}_i||\mathcal{F}_j|}\cdot \left\|A_{i,j}\right\|_2 .$$

We apply the Matrix Khintchine inequality in a similar manner to the previous analysis. The "variance" term grows by a factor of $\max(2^i, 2^j)$ over the bound of mD/N obtained for the random case. As a result, the spectral norm of $A_{i,j}$ is higher by a factor of $\max(2^{i/2}, 2^{j/2})$. On the other hand, the effective ℓ_2-norm of the vector drops by $2^{-(i+j)/2}$. The trade-off "breaks in our favor" and the dominating term in the bound is $A_{0,0}$, whose spectral norm is on the same order as the spectral norm of the A in the case of the previous random q-XOR analysis!

More formally, we have that with high probability over the draw of the b_C's, it holds that

$$\begin{aligned}
&\max_{y\in\{-1,1\}^N} y^\top \sum_{0\le i,j\le t} A_{i,j} y \le \sum_{0\le i,j\le t} \sqrt{|\mathcal{F}_i||\mathcal{F}_j|}\cdot \left\|A_{i,j}\right\|_2 \le \sum_{0\le i,j\le t} N\cdot 2^{-(i+j)/2}\cdot O\left(\sqrt{\frac{2^{\max(i,j)} mD\ell\log n}{N}}\right)\\
&\le N\cdot O\left(\sqrt{\frac{mD\ell\log n}{N}}\right)\sum_{0\le i,j\le t} 2^{-(i+j)/2}\cdot 2^{\max(i,j)/2} = N\cdot O\left(\sqrt{\frac{mD\ell\log n}{N}}\right)\sum_{i=0}^{t}\sum_{j=0}^{i} 2^{-j/2}\\
&\le N\cdot O\left(\sqrt{\frac{mD\ell\log n}{N}}\right)\cdot O(\log n),
\end{aligned}$$

where we use that $t \le O(\log m) \le O(\log n)$. Thus, the latter quantity is $\le \varepsilon D$ provided that $m \ge O((n/\ell)^{q/2}\ell\log^3 n/\varepsilon^2)$. Note that this is a $\log^2 n$ factor higher than the threshold in Item (2b) in Theorem 2.0.2.

Row reweighting. This row bucketing analysis loses this extra $\log^2 n$ factor because it uses "hard cut-offs" to determine the buckets. A slicker analysis, due to [HKM23], instead uses the following *row reweighting* strategy, which one can view as a smoother version of the row bucketing analysis. Let W be the diagonal matrix with S-th entry $W_S = \Upsilon_S + mD/N$, i.e., it is Υ_S plus the average of the Υ_S's, and we now consider the matrix $\tilde{A} = W^{-1/2}AW^{-1/2}$. When we use $\tilde{A}$, there are two immediate issues to resolve. First, we need to relate $\mathrm{val}(f)$ and $\|\tilde{A}\|_2$, and second, we need to bound $\|\tilde{A}\|_2$.

To handle the first issue, we observe that for any $x\in\{-1,1\}^n$, letting $y = x^{\odot\ell}$, we have that

$$f(x) = \frac{1}{mD} y^\top A y = \frac{1}{mD}(W^{1/2}y)^\top \tilde{A}(W^{1/2}y) \le \frac{1}{mD}\|\tilde{A}\|_2\cdot\|W^{1/2}y\|_2^2 \tag{2.1}$$

$$= \frac{1}{mD}\|\tilde{A}\|_2\cdot \mathrm{tr}(W) = \frac{1}{mD}\|\tilde{A}\|_2\cdot 2mD = 2\|\tilde{A}\|_2 . \tag{2.2}$$

Here, the bound $\|W^{1/2}y\|_2^2 \le \mathrm{tr}(W)$ uses that W is diagonal and that $y\in\{-1,1\}^N$.

It thus remains to bound $\|\tilde{A}\|$, which we will do using Matrix Khintchine (Fact 3.4.2). To do this, we need to compute the variance term, which is $\|\mathbb{E}[\tilde{A}^2]\|$. We have

$$\mathbb{E}[\tilde{A}^2] = \sum_{C\in H} W^{-1/2}A_C W^{-1}A_C W^{-1/2} .$$

Recall that, as we observed in Section 2.1, for a fixed $C\in H$, the matrix A_C has at most one nonzero entry per row/column. Because W is diagonal, this implies that $W^{-1/2}A_C W^{-1}A_C W^{-1/2}$

is a diagonal matrix as well, and so $\mathbb{E}[\tilde{A}^2]$ is diagonal. Furthermore, the S-th diagonal entry of $\mathbb{E}[\tilde{A}^2]$ is

$$\frac{1}{\sqrt{W_S}}\left(\sum_{T:S\oplus T\in H} A_C(S,T)\frac{1}{W_T}A_C(T,S)\right)\frac{1}{\sqrt{W_S}} = \frac{1}{W_S}\sum_{T:S\oplus T\in H}\frac{1}{W_T} = \frac{1}{\Upsilon_S+\frac{mD}{N}}\sum_{T:S\oplus T\in H}\frac{1}{\Upsilon_T+\frac{mD}{N}}$$
$$\leq \frac{1}{\Upsilon_S}\sum_{T:S\oplus T\in H}\frac{N}{mD} \leq \frac{N}{mD},$$

where the last inequality uses that for a fixed S, the number of T where $S\oplus T\in H$ is exactly Υ_S.

By applying Fact 3.4.2, we thus conclude that with probability $1-o(1)$ over the draw of the b_C's, it holds that $\|\tilde{A}\|_2 \leq O(\sqrt{\frac{N}{mD}\cdot \ell\log n})$. Thus,

$$\max_{x\in\{-1,1\}^n} f(x) \leq 2\|\tilde{A}\|_2 \leq O\left(\sqrt{\frac{N\ell\log n}{mD}}\right).$$

Using that $D/N \sim (\ell/n)^{q/2}$ (Fact 3.6.1), it follows that the right-hand side above is $\leq \varepsilon$ when $m \geq O((n/\ell)^{q/2}\ell\log n/\varepsilon^2)$, which finishes the proof.

Summary: Row Bucketing/Reweighting

(1) When H is arbitrary (the semirandom case), $\|A\|_2$ might no longer provide a good bound on $\mathrm{val}(f)$ because $\Upsilon_S = |\{C\in H : |S\cap C| = q/2\}|$ may be much larger than the average value for some S, and this quantity controls the variance of $\|A\|_2$.

(2) However, $\mathrm{val}(f)$ is controlled by quadratic forms $y^\top Ay$ for Boolean vectors $y\in\{-1,1\}^N$, whereas the "bad" quadratic forms that make $\|A\|_2$ large come from sparse vectors. In other words, it suffices to bound $\|A\|_{\infty\to 1}$ instead of $\|A\|_2$.

(3) We can bound $\|A\|_{\infty\to 1}$ by either the row bucketing strategy or the "smoother" row reweighting strategy of [HKM23]. Both approaches use that Boolean vectors $y\in\{-1,1\}^N$ are "spread" to handle the issue that the variance term Υ_S might be very large for some S's.

2.3 Handling correlated randomness with row pruning

In this section, we will prove Theorem 2.0.4. Unlike in Sections 2.1 and 2.2, in this section we will consider a random process that generates polynomials $f_b(x)$ with *correlated* coefficients b_C. We will thus develop a new technical idea, *row pruning*, that we use to handle the challenges posed by the correlated coefficients.

As we mentioned in Example 2.0.3, by standard definitions and transformations (Definition 3.3.1 and Fact 3.3.3), in order to prove Theorem 2.0.4, it suffices to prove the following lemma.

Lemma 2.3.1. *Let q, k and n be integers with $k\leq n$ and q even. Let $\delta,\varepsilon\in(0,1)$. Let $H_1,\dots,H_k$ be q-uniform hypergraph matchings on the vertex set $[n]$ with $|H_i| = \delta n$, i.e., for every $i\in[k]$, the hypergraph H_i is a collection of δn disjoint hyperedges, and each hyperedge $C\in H_i$ has size $|C| = q$.*

For each $b \in \{-1,1\}^k$, let $f_b(x) = \sum_{i=1}^k b_i \sum_{C \in H_i} \prod_{v \in C} x_v$. Suppose that $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(f_b)] \geq \varepsilon\delta nk$. Then, $k \leq n^{1-2/q} O(\log n)/(\varepsilon^2\delta^2)$.

Before we proceed with the proof of Lemma 2.3.1, let us comment on the differences between the setting of Lemma 2.3.1 compared to Theorem 2.0.2. In Theorem 2.0.2, in the *semirandom q-XOR* setting (Item (2b)), we define a polynomial $f(x) = \sum_{C \in H} b_C x_C$, where (1) the hypergraph H is *arbitrary* and (2) each b_C is *independently* chosen from $\{-1,1\}$. To compare with Theorem 2.0.2, we can view the polynomial f_b in Lemma 2.3.1 as being defined by (1) the "full" hypergraph $H := \cup_{i=1}^k H_i$, and (2) signs $b_C \in \{-1,1\}$ for each $C \in H$. However, unlike in Theorem 2.0.2, the b_C's are *no longer independent*! Indeed, the randomness is correlated; we can view the partition of the hypergraph H into the matchings $H_1, \ldots, H_k$ as partitioning the hyperedges according to their correlated signs b_C. Namely, if $C, C' \in H_i$ for some $i \in [k]$, then $b_C = b_{C'} = b_i$. On the other hand, unlike in the semirandom case, the hypergraph $H = \cup_{i=1}^k H_i$ in Lemma 2.3.1 is *not arbitrary*, as it is the union of k matchings $H_1, \ldots, H_k$. We have thus traded correlations in the "right-hand sides" b_C for additional structure in the hypergraph H. Note that the structure in H, i.e., that $H = \cup_{i=1}^k H_i$ is the union of k matchings, is crucial, as without this condition Lemma 2.3.1 is false.

Another key difference is that, unlike the algorithmic setting of Theorem 2.0.2, in Lemma 2.3.1 we are merely seeking to prove an existential statement and we do not care about the runtime of the (implicit) algorithm at all! For this reason one might expect to prove Lemma 2.3.1 by using natural probabilistic arguments. Indeed, in the case of semirandom q-XOR, one can argue that the polynomial f has low value by a simple union bound argument, and so the main difficulty in Theorem 2.0.2 is finding an *algorithm* that can certify that $\mathrm{val}(f)$ is low. In the setting of Lemma 2.3.1, the main challenge is that the polynomials f_b have significantly *limited* randomness even compared to the semirandom setting. Namely, all the constraints $C \in H_i$ share the *same* right-hand side b_i, and so there are only $k \ll n$ bits of independent randomness, which is insufficient randomness to execute a union bound argument.

Nonetheless, we can establish a good bound on the value of the polynomial f_b in Lemma 2.3.1 by constructing a subexponential-sized spectral certificate of low value. A priori, bounding the spectral value might seem like a rather roundabout route to bound $\mathrm{val}(f)$. However, shifting to this (stronger) target allows us to leverage the spectral techniques based on Kikuchi matrices. The significantly smaller amount of randomness in the polynomials f_b produced in Lemma 2.3.1, compared to, e.g., semirandom instances, poses additional technical challenges which we shall handle with a technique called *row pruning*. This technique crucially exploits the combinatorial matching structure in the hypergraphs $H_1, \ldots, H_k$.

Spectral refutations for f_b. To prove Lemma 2.3.1, we need to upper bound $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(f_b)]$, which we will do by bounding the spectral norm of an appropriately chosen Kikuchi matrix. As a first attempt, we will use the general approach outlined in Section 2.1 to define a matrix whose quadratic form is equal to $f_b(x)$. Namely, for a choice of the parameter ℓ (to be determined later) and every $b \in \{-1,1\}^k$, we define the Kikuchi matrix $A_{f_b} = \sum_{i=1}^k b_i \sum_{C \in H_i} A_C$, where A_C is defined in Definition 2.1.1. For notational convenience, we will suppress the subscript f_b and let $A := A_{f_b}$.

By Proposition 2.1.2, for any $x \in \{-1,1\}^n$ (and letting $x^{\odot \ell}$ be defined as in Proposition 2.1.2), we have that

$$f_b(x) \leq \frac{1}{D} \left\| x^{\odot \ell} \right\|_2^2 \|A\|_2 = \frac{N}{D} \|A\|_2 \leq O(1) \left(\frac{n}{\ell}\right)^{q/2} \|A\|_2 ,$$

where we have that $D/N \sim (\ell/n)^{q/2}$ by Fact 3.6.1. We thus conclude that

$$\varepsilon\delta nk \leq \mathbb{E}_b[\mathrm{val}(f_b)] \leq O(1)\left(\frac{n}{\ell}\right)^{q/2} \mathbb{E}_b[\|A\|_2], \tag{2.3}$$

where the first inequality is by assumption in Lemma 2.3.1, and so it remains to bound $\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\|A\|_2]$.

We can write $A = \sum_{i=1}^{k} b_i A_i$ as a matrix Rademacher series, where $A_i := \sum_{C\in H_i} A_C$. By the matrix Khintchine inequality (Fact 3.4.2), we have $\mathbb{E}[\|A\|_2] \leq O(\sqrt{\log N})\left\|\sum_i A_i^2\right\|_2^{1/2}$.

A combinatorial proxy for $\|A\|_2$. As mentioned in Section 2.1, we can view the matrix A_C as the adjacency matrix of a graph: the vertices are sets $S \in \binom{n}{\ell}$, and we have an edge (S,T) if and only if $S \oplus T = C$. We can thus view the matrix A_i as a graph, which is obtained by taking the *union*, over all $C \in H_i$, of the graphs A_C.

Let Δ_i be the maximum degree of any node in the Kikuchi graph A_i, and let $\Delta = \max_{1\leq i\leq k}\Delta_i$. Notice that because Δ_i is the maximum degree of any node in the graph A_i, it follows that Δ_i is the maximum ℓ_1-norm of any row/column in A_i, and hence is an upper bound on $\|A_i\|_2$. Thus, we obtain the bound $\left\|\sum_i A_i^2\right\|_2 \leq \sum_i \|A_i\|_2^2 \leq k\Delta^2$, and we conclude that the maximum degree of the A_i's naturally controls the spectral norm of A. Indeed, we have $\mathbb{E}_b[\|A\|_2] \leq \Delta \cdot O(\sqrt{k\ell\log n})$.

The quantity Δ_i arises naturally in the setting of LDC lower bounds due to the *correlated randomness* of the coefficients in the polynomial f_b. Indeed, as we saw above, the quantity Δ_i comes from the variance term $\left\|\sum_i A_i^2\right\|_2^{1/2}$ in our application of Matrix Khintchine, and this variance term arises because of the correlated randomness. As we have stated earlier, one can view the hypergraph H_i as simply grouping the hyperedges C that share the same coefficient b_i, and likewise the matrix $A_i = \sum_{C\in H_i} A_C$ simply extracts the "b_i-component" of the matrix A. One can thus view the quantity Δ_i as a means of controlling the contribution of the "b_i-component" to the overall variance term for the random matrix A.

Let us now investigate bounds on Δ. By Proposition 2.1.2, we have already observed that for each $C \in H_i$, the graph A_C is a matching with D edges. Therefore, the total number of edges in A_i is exactly $D|H_i| = D\delta n$, and so the average degree of A_i is $d_i = \delta n D/N \sim n(\ell/n)^{q/2}$. We must have $\Delta \geq d_i$ and $\Delta \geq 1$, and so we have $\Delta \gtrsim O(1)\max\{1, n(\ell/n)^{q/2}\}$. If Δ happens to be equal to this minimum possible value, then substituting it in Eq. (2.3) yields:

$$\varepsilon\delta nk \leq O(1)\left(\frac{n}{\ell}\right)^{q/2}\sqrt{k\ell\log n}\cdot\max\{1, n(\ell/n)^{q/2}\},$$

which implies that $k \leq O(\ell\log n)\cdot\max\{n^{q-2}/\ell^q, 1\}$. This is minimized at $\ell = n^{1-2/q}$ to give the bound of $k \leq \tilde{O}(n^{1-2/q})$, which is the bound we would like to show in Lemma 2.3.1.

However, there is one crucial problem: Δ_i is much larger than the average degree d_i of A_i. In fact, with some thought, one can see that $\Delta_i = \lfloor\frac{2\ell}{q}\rfloor$. This is achieved by a row S constructed by (1) choosing $C_1, \ldots, C_t \in H_i$, where $t = \lfloor\frac{2\ell}{q}\rfloor$, (2) choosing an arbitrary subset $C_j' \subseteq C_j$ of size $q/2$ from each such C_j, and then (3) setting S to be the union of the C_j''s (which are disjoint since the C_j's are disjoint as H_i is a matching) along with $\ell - \frac{tq}{2}$ arbitrary extra elements from $[n] \setminus \cup_{j=1}^{t} C_j$ to pad S so that it has size ℓ. We can also observe that if $q \geq 3$ and we substitute $\Delta = \Omega(\ell)$ into Eq. (2.3), we obtain the best lower bound by setting ℓ to be as large as possible, i.e., $\ell = \Omega(n)$. But if we do so, our "lower bound" becomes $k \leq O(n\log n)$, which is worse than the trivial $k \leq n$ bound!

Handling irregularities: row pruning. The fact that the maximum degree Δ_i is much larger than the average degree $d_i \sim n(\ell/n)^{q/2}$ of A_i is an inherent issue that we need to overcome. In handling this issue, we will need to crucially use that the H_i's are matchings; notice that we have not used this property so far, and Lemma 2.3.1 is clearly false without this assumption!

We will now make the following key observation. While there are some nodes in the graph A_i that have degree much larger than the average d_i, these nodes are "rare", and we can remove them while only deleting a small number of vertices (and therefore also edges) from A_i. Of course, a small fraction of bad rows can still cause $\|A_i\|_2$ (and also $\|A\|_2$) to be too large. But, we can now combine this with the key observation that we made in Section 2.2: to bound $\mathrm{val}(f_b)$, we only need to bound $y^\top A y$ for *Boolean* vectors $y \in \{-1,1\}^N$, or equivalently, we need to bound $\|A\|_{\infty\to 1}$ rather than $\|A\|_2$. Unlike $\|A\|_2$, the quantity $\|A\|_{\infty\to 1}$ is insensitive to deleting a small fraction of rows/columns, since ± 1-coordinate vectors when restricted to a small number of rows must have correspondingly small ℓ_2-norm. Hence, we can simply delete the bad nodes, or equivalently "zero out" the bad rows/columns, of the matrix A_i and thus obtain a matrix B_i where (1) $\|A_i - B_i\|_{\infty\to 1}$ is small, and (2) the maximum degree of B_i is $O(d_i) = O(n(\ell/n)^{q/2})$. The first property ensures that $\|A\|_{\infty\to 1} \approx \|B\|_{\infty\to 1}$ where $B := \sum_{i=1}^k b_i B_i$, and the second property implies that, by our earlier calculations, $\|B\|_2$ yields the desired bound on $\mathrm{val}(f_b)$.

To prove that only a small fraction of nodes can have a large degree in any A_i, we view the degree of any node S as a polynomial in the corresponding indicator variables $z \in \{0,1\}^n$ with $\sum_i z_i = \ell$ and use tail inequalities for low-degree polynomials (that generalize concentration of Lipschitz functions) of Kim and Vu and extensions [KV00, SS12] (Fact 3.4.3) to bound the probability that the degree of a random node S is at least $\mathrm{polylog}(n)$ times the average d_i. This relies on establishing strong bounds on the expected partial derivatives of the degree polynomial by using that the H_i's are matchings.

Indeed, the degree of a node S is at most $\mathsf{Deg}(z) := \sum_{C\in H_i} \sum_{C'\subseteq C:|C'|=q/2} z_{C'}$, where z is the 0/1 indicator vector of S. Using Fact 3.4.3, we can show that with probability $1 - 1/\mathrm{poly}(n)$, a random S has degree at most $\mathrm{polylog}(n)\cdot d_i$, provided that $\ell \gtrsim n^{1-2/q}$ (which implies that $d_i \gg 1$). Note that this crucially requires that H_i is a hypergraph matching!

We now let B be the matrix obtained by deleting ("zeroing out") each row/column S where S has degree $\geq \mathrm{polylog}(n)\cdot d_i$ in some d_i. We also write $B = \sum_{i=1}^k b_i B_i$, where the B_i's are defined similarly from the A_i's. We have that

$$\|A - B\|_{\infty\to 1} \leq N \cdot \frac{2k}{\mathrm{poly}(n)} \cdot kn \leq \frac{Nn^3}{\mathrm{poly}(n)} = \frac{N}{\mathrm{poly}(n)},$$

as (1) the total number of rows/columns removed is at most $N \cdot \frac{1}{\mathrm{poly}(n)} \cdot k$, using the concentration bound along with a union bound over all $i \in [k]$, (2) the maximum number of nonzero entries per row/column of A_i is at most δn, and (3) we have $k \leq n$. By construction, we also have that for any $i \in [k]$, each row/column of B_i has at most $d_i\,\mathrm{polylog}(n)$ nonzero entries. We thus conclude

that for $\ell \gtrsim n^{1-2/q}$,

$$\varepsilon\delta nk \le \mathbb{E}_b[\mathrm{val}(f_b)] \le \frac{1}{D}\mathbb{E}_b[\|A\|_{\infty\to 1}] \le \frac{1}{D}\mathbb{E}_b[\|A-B\|_{\infty\to 1} + \|B\|_{\infty\to 1}] \le \frac{1}{D}\left(o(N) + \mathbb{E}_b[N\|B\|_2]\right)$$
$$\le \frac{1}{D}\left(o(N) + O(N\sqrt{\log N})\left\|\sum_i B_i^2\right\|_2^{1/2}\right) \le \left(\frac{n}{\ell}\right)^{q/2}\left(o(1) + O(1)\sqrt{\ell\log n}\cdot \mathrm{polylog}(n)\sqrt{\sum_{i=1}^k d_i^2}\right)$$
$$\le \left(\frac{n}{\ell}\right)^{q/2}\left(o(1) + O(1)\sqrt{k\ell\,\mathrm{polylog}(n)}\cdot n(\ell/n)^{q/2}\right) = O\left(\sqrt{k\ell\,\mathrm{polylog}(n)}\cdot n\right)$$
$$\implies \varepsilon^2\delta^2 k \le O(\ell\,\mathrm{polylog}(n))\,.$$

Setting $\ell = n^{1-2/q}$, we thus conclude that $k \le O(n^{1-2/q}\,\mathrm{polylog}(n)/(\varepsilon^2\delta^2))$, which proves Lemma 2.3.1 up to a small loss in the $\mathrm{polylog}(n)$ factor.

Obtaining a better polylog(n) factor. The above row pruning approach is rather intuitive, but it is a bit lossy in the final $\mathrm{polylog}(n)$ factor. We can sharpen the final bound on k via a modified, but perhaps less intuitive or general, row pruning argument.

We now define the matrix B as follows. For each $i \in [k]$, we let B_i be the matrix obtained from A_i by replacing any row/column in $A_i := \sum_{C\in H_i} A_C$ with all 0's if it has at least 2 nonzero entries. We will then show that if $\ell \lesssim n^{1-2/q}$, i.e., the average degree d_i of A_i is $d_i \ll 1$, then *because H_i is a matching*, B_i has at least a constant fraction of the entries of A_C *for every* $C \in H_i$. We can then further delete entries of B_i so that each $C \in H_i$ contributes *exactly* the same number of entries to B_i; this ensures that each monomial x_C in f_b has the same contribution to the corresponding quadratic form on B. Then, we bound $\mathrm{val}(f_b)$ via $\|B\|_2$, using that $\|B_i\|_2 \le 1$ for all $i \in [k]$ as B_i has at most one nonzero entry per row/column.

In more detail, we make the following definition.

Definition 2.3.2. Let $\ell := n^{1-2/q}/c$ for some absolute constant $c \ge e^{16}$ if $q \ge 4$, and let $\ell = 1$ if $q = 2$. Note that in either case, $\ell = O(n^{1-2/q})$. Let $N := \binom{n}{\ell}$. For each q-uniform hypergraph matching H_i, let $B_i \in \mathbb{R}^{N\times N}$ denote the matrix indexed by sets $S, T \in \binom{[n]}{\ell}$ where $B_i(S,T) = 1$ if the pair (S,T) satisfies (1) $S \oplus T = C \in H_i$, and (2) $|S \oplus C'| \ne \ell$, $|T \oplus C'| \ne \ell$ for every $C' \in H_i$ with $C' \ne C$. We set $B_i(S,T) = 0$ otherwise. We let $B := \sum_{i=1}^k b_i B_i$.

As we have said, the matrices in Definition 2.3.2 are almost the same as setting $B_i = \sum_{C\in H_i} A_C$ where A_C is defined as in Definition 2.1.1; the key difference is that the definition of B_i in Definition 2.3.2 "zeros out" all rows/columns in $\sum_{C\in H_i} A_C$ that have more than one nonzero entry. Because we have removed entries from the matrix $\sum_{C\in H_i} A_C$, one might be worried that we have removed *all* the entries and the matrix B_i is identically 0. This is not the case because H_i is a matching and $\ell \lesssim n^{1-2/q}$, as we show in the lemma below.

Lemma 2.3.3. *There is an integer D' such that the following holds. Fix $i \in [k]$, and let B_i be one of the matrices defined in Definition 2.3.2. For any $C \in H_i$, the number of pairs (S,T) with $S \oplus T = C$ and $B_i(S,T) = 1$ is exactly D'. Moreover, we have that $D'/N \ge \frac{1}{2}\binom{q}{q/2}e^{-3q}\cdot(\frac{\ell}{n})^{q/2}$.*

We postpone the proof of Lemma 2.3.3, and now finish the proof of Lemma 2.3.1.

For each $x \in \{-1,1\}^n$, let $y \in \{-1,1\}^N$ be the vector where $y_S = \prod_{v \in S} x_v$. We then have that

$$\begin{aligned} y^\top B y &= \sum_{i=1}^{k} b_i (y^\top B_i y) = \sum_{i=1}^{k} b_i \sum_{C \in H_i} \sum_{(S,T): S \oplus T = C} y_S y_T \\ &= \sum_{i=1}^{k} b_i \sum_{C \in H_i} D' \cdot y_S y_T \quad \text{(by Lemma 2.3.3)} \\ &= D' \sum_{i=1}^{k} b_i \sum_{C \in H_i} \prod_{v \in C} x_v \\ &= D' f_b(x) \\ &\implies \mathbb{E}_b[\mathrm{val}(f_b)] \le \frac{N}{D'} \cdot \mathbb{E}_b[\|B\|_2], \end{aligned}$$

where the last inequality uses that $\|y\|_2^2 = N$; here, $m = \sum_{i=1}^{k} |H_i| = \delta n k$ is the total number of constraints.

It thus remains to bound $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\|B\|_2]$. As each b_i is an independent bit from $\{-1,1\}$, the matrix $B = \sum_{i=1}^{k} b_i B_i$ is the sum of k independent, mean 0 random matrices. We will use Matrix Khintchine (Fact 3.4.2) to bound $\mathbb{E}[\|B\|_2]$. We observe that $\|B_i\|_2 \le 1$ by construction, as the ℓ_1-norm of any row/column of B_i is at most 1. It then follows that $\|\sum_{i=1}^{k} B_i^2\|_2 \le \sum_{i=1}^{k} \|B_i\|_2^2 \le k$. Hence, by Fact 3.4.2, it follows that $\mathbb{E}[\|B\|_2] \le O(\sqrt{k \log N}) = O(\sqrt{k\ell \log n})$.

We thus have

$$\varepsilon \delta n k \le \mathbb{E}_{b \in \{-1,1\}^k}[\mathrm{val}(f_b)] \le \frac{N}{D'} O(\sqrt{k\ell \log n}) \le \left(\frac{n}{\ell}\right)^{q/2} \cdot O(\sqrt{k\ell \log n}) \le n \cdot O\left(\sqrt{k n^{1-2/q} \log n}\right),$$

where we use that $\ell = n^{1-2/q}/c$ and the bound on $\frac{D'}{N}$ from Lemma 2.3.3. We thus conclude that $k \le n^{1-2/q} \cdot O(\log n)/(\varepsilon^2 \delta^2)$.

It remains to prove Lemma 2.3.3.

Proof of Lemma 2.3.3. First, let $C \in H_i$ be any element. We first show that the number of pairs (S,T) with $S \oplus T = C$ and $B_i(S,T) = 1$ is independent of C. Indeed, let $C' \in H_i$ be different from C. As H_i is a matching, we have that C and C' are disjoint. Let π be an arbitrary bijection between C and C' and extend π to act on all of $[n]$ by acting as the identity on elements not in $C \cup C'$. It is simple to observe that if (S,T) is any pair satisfying the above criterion for C, then (S',T'), obtained by applying π to all elements of S and T, satisfies the criterion for C'. Hence, the number of pairs is independent of the choice of $C \in H_i$.

We note that it is clear from symmetry that D' depends only on $|H_i|$, q, and n. As $|H_i| = \delta n$ for all i, it follows that D' does not depend on i.

We now finish the proof. We have two cases. If $q = 2$, then $\ell = 1$ and $N = \binom{n}{\ell} = n$. This implies that $D' = 2$, as each H_i is a matching, so if $C = \{u,v\} \in H_i$, then $B_i(u,v) = B_i(v,u) = 1$. Thus, in this case the conclusion trivially holds.

Now, suppose $q \ge 4$. Let $C \in H_i$ be arbitrary. We first lower bound D'. We observe that $S \oplus T = C$ if and only if $S = C_S \cup Q$ and $T = C_T \cup Q$, where $C_S, C_T \subseteq C$ are disjoint subsets of size exactly $q/2$, so that $C = C_S \cup C_T$, and $Q \subseteq [n] \setminus C$ has size exactly $\ell - q/2$. It follows that if

$S \oplus T = C$ and for some $C' \neq C \in H_i$, either $|S \oplus C'| = \ell$ or $|T \oplus C'| = \ell$, then it must be the case that $|Q \cap C'| = q/2$. Hence, we have that

$$D' \geq \binom{q}{q/2}\binom{n-q}{\ell - q/2} - |H_i| \cdot \binom{q}{q/2}^2 \binom{n-2q}{\ell - q} \ .$$

Applying Fact 3.6.1, we thus have that

$$\begin{aligned} D'/N &\geq \binom{q}{q/2} e^{-3q} \left(\frac{\ell}{n}\right)^{q/2} - n \cdot \binom{q}{q/2}^2 e^{3q} \left(\frac{\ell}{n}\right)^{q} \\ &= \binom{q}{q/2} e^{-3q} \left(\frac{\ell}{n}\right)^{q/2} \left(1 - n \cdot 2^q e^{6q} \left(\frac{\ell}{n}\right)^{q/2}\right) \\ &\geq \frac{1}{2}\binom{q}{q/2} e^{-3q} \left(\frac{\ell}{n}\right)^{q/2} \ , \end{aligned}$$

where we use that $\ell \leq n^{1-2/q}/e^{16}$. □

Summary: Row Pruning

(1) In the LDC setting, $H = \cup_{i=1}^{k} H_i$ where each $C \in H_i$ shares the same coefficient b_i. The correlated coefficients make $\|A\|_2$ too large to provide a good bound on $\mathrm{val}(f)$, even when each H_i is a matching (in some sense, the "nicest" case).

(2) $\|A\|_2$ is controlled by the maximum degrees Δ_i's of the Kikuchi graphs $A_i = \sum_{C \in H_i} A_C$ for each $i \in [k]$. To get a good bound, we need $\Delta_i \approx d_i$, where the d_i's are the average degrees.

(3) Even when H_i is a matching, $\Delta_i \gg d_i$, which causes $\|A\|_2$ to be too large. However, the "bad nodes" of large degree in each A_i are rare, which one can show using Kim–Vu-style concentration bounds ([KV00, SS12], Fact 3.4.3). By deleting all of the bad nodes from A_i, we obtain a new graph B_i, and we can use $\|B\|_2$, where $B = \sum_{i=1}^{k} b_i B_i$, to bound $\mathrm{val}(f_b)$.

Chapter 3

Background and Preliminaries

3.1 Basic notation

We let $[n]$ denote the set $\{1, \ldots, n\}$. For two subsets $S, T \subseteq [n]$, we let $S \oplus T$ denote the symmetric difference of S and T, i.e., $S \oplus T \coloneqq \{i : (i \in S \wedge i \notin T) \vee (i \notin S \wedge i \in T)\}$. For a natural number $t \in \mathbb{N}$, we let $\binom{[n]}{t}$ be the collection of subsets of $[n]$ of size exactly t.

For a rectangular matrix $A \in \mathbb{R}^{m \times n}$, we let $\|A\|_2 \coloneqq \max_{x \in \mathbb{R}^m, y \in \mathbb{R}^n : \|x\|_2 = \|y\|_2 = 1} x^\top A y$ denote the spectral norm of A, and $\|A\|_{\infty \to 1} \coloneqq \max_{x \in \{-1,1\}^m, y \in \{-1,1\}^n} x^\top A y$. We note that $\|A\|_{\infty \to 1} \leq \sqrt{nm}\|A\|_2$.

Given a multiset H, we will use the notation $C \in H$ to refer to a distinct element of C, and $C \neq C'$ for $C, C' \in H$ to denote that C and C' are distinct elements in H (even if they are two different copies of the same element).

Given a set R and variables $x_1, \ldots, x_n$, we will let $x_R \coloneqq \prod_{i \in R} x_i$. In particular, $x_C \coloneqq \prod_{i \in C} x_i$.

Given a graph $G = (V, E)$ with n vertices and m edges (including self-loops[1]), we write $D_G \in \mathbb{R}^{n \times n}$ as the diagonal degree matrix, $A_G \in \mathbb{R}^{n \times n}$ as the adjacency matrix, and $L_G = D_G - A_G$ as the unnormalized Laplacian (note that the self-loops do not contribute to L_G). Furthermore, we write $\widetilde{L}_G = D_G^{-1/2} L_G D_G^{-1/2}$ to be the *normalized* Laplacian, and denote its eigenvalues as $0 = \lambda_1(\widetilde{L}_G) \leq \lambda_2(\widetilde{L}_G) \leq \cdots \leq \lambda_n(\widetilde{L}_G) \leq 2$.

For any subset $S \subseteq V$, we denote $G[S]$ as the subgraph of G induced by S, and $G\{S\}$ as the induced subgraph $G[S]$ but with self-loops added so that any vertex in S has the same degree as its degree in G.

3.1.1 Graph pruning and expander decomposition

It is a standard result that given a graph with m edges and average degree d, one can delete vertices such that the resulting graph has minimum degree εd and at least $(1 - 2\varepsilon)m$ edges. We include a short proof for completeness.

Lemma 3.1.1 (Graph pruning)**.** *Let G be an n-vertex graph with average degree d and $m = \frac{nd}{2}$ edges, and let $\varepsilon \in (0, 1/2)$. There is an algorithm that deletes vertices of G such that the resulting graph has minimum degree εd and at least $(1 - 2\varepsilon)m$ edges.*

[1] Each self-loop contributes 1 to the degree of a vertex.

Proof. The algorithm is simple: repeatedly remove any vertex with degree $< \varepsilon d$. First, we show by induction that each deletion cannot decrease the average degree. Suppose there are $n' \le n$ vertices left and average degree $d' \ge d$. Then, after deleting a vertex u with degree $d_u < \varepsilon d$, the average degree becomes $\frac{n'd'-2d_u}{n'-1} > \frac{n'd-2\varepsilon d}{n'-1} = d \cdot \frac{n'-2\varepsilon}{n'-1}$. Thus, for $\varepsilon < 1/2$, the average degree is always at least d. Furthermore, since the algorithm can delete at most n vertices, it can delete at most $\varepsilon dn = 2\varepsilon m$ edges. □

We will also need an algorithm that partitions a graph into expanding clusters such that total number of edges across different clusters is small. Expander decomposition has been developed in a long line of work [KVV04, ST11, Wul17, SW19] and has a wide range of applications. For our algorithm, we only require a very simple expander decomposition that recursively applies Cheeger's inequality.

Fact 3.1.2 (Expander decomposition)**.** *Given a (multi)graph $G = (V, E)$ with m edges and a parameter $\varepsilon \in (0, 1)$, there is a polynomial-time algorithm that finds a partition of V into $V_1, \dots, V_T$ such that $\lambda_2(\widetilde{L}_{G\{V_i\}}) \ge \Omega(\varepsilon^2/\log^2 m)$ for each $i \in [T]$ and the number of edges across partitions is at most εm.*

Proof. Fix $\lambda = c\varepsilon^2/\log^2 m$ for some constant c to be chosen later. The algorithm is very simple. Given a graph $G = (V, E)$ (with potentially parallel edges and self-loops), if $\lambda_2(\widetilde{L}_G) < \lambda$, then by Cheeger's inequality we can efficiently find a subset $S \subseteq V$ with $\mathrm{vol}(S) \le \mathrm{vol}(\overline{S})$ such that $\frac{|E(S,\overline{S})|}{\mathrm{vol}(S)} < \sqrt{2\lambda}$. Here $\mathrm{vol}(S) \coloneqq \sum_{v \in S} \deg(v)$. Then, we cut along S, add self-loops to the induced subgraphs $G[S]$ and $G[\overline{S}]$ so that the vertex degrees remain the same (each self-loop contributes 1 to the degree). This produces two graphs $G\{S\}$ and $G\{\overline{S}\}$, and we recurse on each. By construction, in the end we will have partitions $V_1, \dots, V_T$ where either V_i is either a single vertex or satisfies $\lambda_2(\widetilde{L}_{G\{V_i\}}) \ge \lambda$.

We now bound the number of edges cut via a charging argument. Consider the "half-edges" in the graph, where each edge (u, v) contributes one half-edge to u and one to v, and each self-loop counts as one half-edge. Then, $\mathrm{vol}(S)$ equals the number of half-edges attached to S. Now, imagine we have a counter for each half-edge, and every time we cut along S we add $\sqrt{2\lambda}$ to each half-edge attached to S (the smaller side). Since $E(S, \overline{S}) < \sqrt{2\lambda} \cdot \mathrm{vol}(S)$, it follows that the number of edges cut is at most the total sum of the counters. On the other hand, each half-edge can appear on the smaller side of the cut at most $\log_2 2m$ times, as each time the half-edge is on the smaller side of the cut, $\mathrm{vol}(S)$ decreases by at least a factor of 2, and $\mathrm{vol}([n]) = 2m$. So, the total sum must be $\le \sqrt{2\lambda} \cdot 2m \log_2 2m \le \varepsilon m$ for a small enough constant c. □

3.2 Hypergraphs

Definition 3.2.1 (Hypergraphs)**.** An (unweighted and undirected) hypergraph H on a vertex set $[n]$ is a collection of subsets $C \subseteq [n]$ called hyperedges. We say that a hypergraph H is *q-uniform* if $|C| = q$ for all $C \in H$, and that H is a matching if for all distinct $C, C' \in H$, C and C' are disjoint.

For a subset $Q \subseteq [n]$, we define the degree of Q in H, denoted $\deg_H(Q)$, to be $|\{C \in H : Q \subseteq C\}|$.

We will allow hypergraphs to be multisets, in which case we will use the notation $C \in H$ to refer to a distinct element of C, and $C \ne C'$ for $C, C' \in H$ to denote that C and C' are distinct elements in H (even if they are two different copies of the same set).

Definition 3.2.2 (Weighted hypergraphs)**.** A (weighted and undirected) hypergraph H on vertex set $[n]$ is a weight function $\mathrm{wt}_H \colon 2^{[n]} \to \mathbb{R}_{\geq 0}$, i.e., a function from unordered sets $C \subseteq [n]$ to $\mathbb{R}_{\geq 0}$. The hypergraph is $\leq q$-uniform if $|C| > q$ implies that $\mathrm{wt}_H(C) = 0$ and q-uniform if $|C| \neq q$ implies that $\mathrm{wt}_H(C) = 0$.

A (weighted and directed) hypergraph H on vertex set $[n]$ is a weight function $\mathrm{wt}_H \colon S \to \mathbb{R}_{\geq 0}$, where S denotes the set of all *ordered* subsets of $[n]$. The hypergraph is $\leq q$-uniform if for any ordered set $C \subseteq [n]$, $|C| > q$ implies that $\mathrm{wt}_H(C) = 0$ and q-uniform if $|C| \neq q$ implies that $\mathrm{wt}_H(C) = 0$.

For a subset $Q \subseteq [n]$, we define the degree of Q in H, denoted $\deg_H(Q)$, to be $\sum_{C \in [n]^q : Q \subseteq C} \mathrm{wt}_H(C)$, where we say that $Q \subseteq C$ if this containment holds as sets.

3.3 Locally decodable and correctable codes

We refer the reader to the survey [Yek12] for background.

A code is a map $C \colon \{-1,1\}^k \to \{-1,1\}^n$. We say that C is *linear* if the map C, when viewed as a map from $\{0,1\}^k \to \{0,1\}^n$ via the mapping $0 \leftrightarrow 1$ and $1 \leftrightarrow -1$, is a linear map. We note that for linear codes, $k = \dim(\mathcal{V})$, where $\mathcal{V}$ is the image of $\{0,1\}^k$ under the map C. We will typically let $\mathcal{L}$, as opposed to C, denote a linear code, and view $\mathcal{L}$ as a map $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$. We say that C is systematic if for every $b \in \{-1,1\}^k$, $C(b)|_{[k]} = b$. For a code $C \colon \{-1,1\}^k \to \{-1,1\}^n$, we will write $x \in C$ to denote an $x = C(b)$ for some $b \in \{-1,1\}^k$.

Locally decodable codes. A locally decodable code is a code where one can recover any bit b_i of the original message b with good confidence while only reading a few bits of the encoded string in the presence of errors.

Definition 3.3.1 (Locally Decodable Code)**.** A code $C \colon \{-1,1\}^k \to \{-1,1\}^n$ is (q, δ, ε)-locally decodable if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ with the following properties. The algorithm $\mathrm{Dec}(\cdot)$ is given oracle access to some $y \in \{-1,1\}^n$, takes an $i \in [k]$ as input, and satisfies the following: (1) the algorithm Dec makes at most q queries to the string y, and (2) for all $b \in \{-1,1\}^k$, $i \in [k]$, and all $y \in \{-1,1\}^n$ such that $\Delta(y, C(b)) \leq \delta n$, $\Pr[\mathrm{Dec}^y(i) = b_i] \geq \frac{1}{2} + \varepsilon$. Here, $\Delta(x,y)$ denotes the Hamming distance between x and y, i.e., the number of indices $v \in [n]$ where $x_v \neq y_v$.

Following known reductions [Yek12], locally decodable codes can be reduced to the following normal form, which is more convenient to work with.

Definition 3.3.2 (Normal LDC)**.** A code $C \colon \{-1,1\}^k \to \{-1,1\}^n$ is (q, δ, ε)-normally decodable if for each $i \in [k]$, there is a q-uniform hypergraph matching H_i with at least δn hyperedges such that for every $C \in H_i$, it holds that $\Pr_{b \leftarrow \{-1,1\}^k}[b_i = \prod_{v \in C} C(b)_v] \geq \frac{1}{2} + \varepsilon$.

Fact 3.3.3 (Reduction to LDC Normal Form, Lemma 6.2 in [Yek12])**.** *Let $C \colon \{-1,1\}^k \to \{-1,1\}^n$ be a code that is (q, δ, ε)-locally decodable. Then, there is a code $C' \colon \{-1,1\}^k \to \{-1,1\}^{O(n)}$ that is $(q, \delta', \varepsilon')$ normally decodable, with $\delta' \geq \varepsilon\delta/3q^2 2^{q-1}$ and $\varepsilon' \geq \varepsilon/2^{2q}$.*

We recall the lower bound for linear 2-LDCs from [GKST06].

Fact 3.3.4 (Lemma 3.3, Lemma 3.5 in [GKST06])**.** *Let $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ be a linear map, and let $G_1, \ldots, G_k$ be matchings on n vertices such that for every $b \in \{0,1\}^k$ and every $i \in [k]$ and every $(u,v) \in G_i$, it holds that $x_u + x_v = b_i$, where $x = \mathcal{L}(b)$. Suppose that $\frac{1}{k}\sum_{i=1}^k |G_i| \geq \delta n$. Then, $2\delta k \leq \log_2 n$.*

Locally correctable codes. A locally correctable code is defined similarly to a locally correctable code, except that the decoder must now recover any bit x_u of the (uncorrupted) encoded string.

Definition 3.3.5 (Locally correctable code)**.** A map $C\colon \{-1,1\}^k \to \{-1,1\}^n$ is a (q,δ,ε)-locally correctable code if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ that takes input an oracle access to some $y \in \{-1,1\}^n$ and a $u \in [n]$, and has the following properties:

(1) (q queries) For any $y \in \{-1,1\}^n$ and $u \in [n]$, $\mathrm{Dec}^y(u)$ makes at most q queries to the string y;

(2) ($(1/2+\varepsilon)$-correction with δn errors) For all $b \in \{-1,1\}^k$, $u \in [n]$, and all $y \in \{-1,1\}^n$ such that $\Delta(y, C(b)) \le \delta n$, $\Pr[\mathrm{Dec}^y(u) = C(b)_u] \ge 1/2+\varepsilon$. Here, $\Delta(x,y)$ denotes the Hamming distance between x and y, i.e., the number of indices $v \in [n]$ where $x_v \neq y_v$.

Definition 3.3.6 (Smooth LCCs [KT00])**.** A map $C\colon \{-1,1\}^k \to \{-1,1\}^n$ is a δ-smooth q-locally correctable code with completeness $1-\varepsilon$ if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ that takes input an oracle access to some $y \in \{-1,1\}^n$ and a $u \in [n]$, and has the following properties:

(1) (q queries) For any $y \in \{-1,1\}^n$ and $u \in [n]$, $\mathrm{Dec}^y(u)$ makes at most q queries to the string y;

(2) ($(1-\varepsilon)$-completeness) For all $b \in \{-1,1\}^k$, $u \in [n]$, $\Pr[\mathrm{Dec}^{C(b)}(u) = C(b)_u] \ge 1-\varepsilon$.

(3) (δ-smoothness) For all $b \in \{-1,1\}^k$, $u \in [n]$, $x = C(b)$, $v \in [n]$, $\Pr[\mathrm{Dec}^{C(b)}(u) \text{ queries } v] \le \frac{1}{\delta n}$.

We will call such codes $(q,\delta,1-\varepsilon)$-smooth LCCs.

Remark 3.3.7. Any δ-smooth q-LCC with completeness $1-\varepsilon$ is a $(q,\eta\delta,1-\varepsilon-\eta)$-LCC for any $\eta > 0$. Indeed, this follows because if we let $y \in \{-1,1\}^n$ be a corruption of a codeword $x \in C$ with $\eta\delta n$ errors, then the probability that the smooth decoder queries a corrupted entry is $\le \eta$.

Fact 3.3.8 (Systematic Nonlinear Codes, Lemma A.5, Thm A.6 in [BGT17])**.** *Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a δ-smooth q-LCC with completeness $1-\varepsilon$. Then, there is a* systematic *code $C'\colon \{-1,1\}^{k'} \to \{-1,1\}^n$ that is a δ-smooth q-LCC with completeness $1-\varepsilon$, where $k' = \Omega(k/\log(1/\delta))$.*

Like LDCs, (linear) LCCs admit a standard combinatorial characterization, formalized in the definition below.

Definition 3.3.9 (Linear LCC in normal form)**.** A linear code $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ is (q,δ)-normally correctable if for each $u \in [n]$, there is a q-uniform hypergraph matching H_u with at least δn hyperedges such that for every $C \in H_u$ and $b \in \{-1,1\}^k$, it holds that $\prod_{v\in C} x_v = x_u$ where $x = C(b)$.

Every linear LCC can be transformed into a linear LCC in normal form with only a small loss in parameters.

Fact 3.3.10 (Reduction to LCC normal form, Theorem 8.1 in [Dvi16])**.** *Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a linear code that is (q,δ,ε)-locally correctable. Then, there is a linear code $\mathcal{L}'\colon \{0,1\}^k \to \{0,1\}^{2n}$ that is (q,δ')-normally correctable, with $\delta' \ge \delta/2q$.*

Below, we define *design 3-LCCs*, which are an idealized form of linear 3-LCCs in normal form. We note that Reed–Muller codes, the best known construction of 3-LCCs, are designs (see Section 12.11).

Definition 3.3.11 (Design 3-LCCs)**.** Let $H \subseteq \binom{[n]}{4}$ denote a collection of subsets of n of size exactly 4. We say that H is a *design* if, for every pair of vertices $u \neq v \in [n]$, there exists *exactly* one $C \in H$ with $\{u,v\} \subseteq C$.

We say that such an H is a design 3-LCC of dimension k if the subspace $\mathcal{V} := \{x \in \{0,1\}^n : \sum_{v\in C} x_v = 0 \ \forall C \in H\} \subseteq \{0,1\}^n$ has dimension k.

Remark 3.3.12 (Connection between Definition 3.3.11 and Definition 3.3.9). Given a design 3-LCC H, we can construct the hypergraphs H_u for $u \in [n]$ in Definition 3.3.11 by letting $H_u := \{C \setminus \{u\} : C \in H \text{ and } u \in C\}$ be the set of $C \in H$ that contain u (and then remove u). Because H is a design, for every pair $u \neq v \in [n]$, there exists $C \in H$ containing u and v. So, there is exactly one $C' \in H_u$ containing v, which implies that H_u is a perfect 3-uniform hypergraph matching on $[n] \setminus \{u\}$, i.e., $|H_u| = \frac{n-1}{3}$.

3.4 Concentration inequalities

We will rely on the following concentration inequalities. The first is the standard rectangular Matrix Bernstein inequality.

Fact 3.4.1 (Rectangular Matrix Bernstein, Theorem 1.6 of [Tro12]). *Let $X_1, \ldots, X_k$ be independent random $d_1 \times d_2$ matrices with $\mathbb{E}[X_i] = 0$ and $\|X_i\| \leq R$ for all i. Let $\sigma^2 \geq \max(\|\mathbb{E}[\sum_{i=1}^k X_i X_i^\top]\|_2, \|\mathbb{E}[\sum_{i=1}^k X_i^\top X_i]\|_2)$. Then for all $t \geq 0$, $\Pr[\|\sum_{i=1}^k X_i\|_2 \geq t] \leq (d_1 + d_2)\exp(\frac{-t^2/2}{\sigma^2 + Rt/3})$.*

The second is the following non-commutative Khintchine inequality [LP91].

Fact 3.4.2 (Rectangular Matrix Khintchine Inequality, Theorem 4.1.1 of [Tro15]). *Let $X_1, \ldots, X_k$ be fixed $d_1 \times d_2$ matrices and $b_1, \ldots, b_k$ be i.i.d. from $\{-1, 1\}$. Let $\sigma^2 \geq \max(\|\sum_{i=1}^k X_i X_i^\top\|_2, \|\sum_{i=1}^k X_i^\top X_i\|_2)$. Then*

$$\mathbb{E}\left[\ \|\sum_{i=1}^k b_i X_i\|_2\ \right] \leq \sqrt{2\sigma^2 \log(d_1 + d_2)}\,,$$

and

$$\Pr[\|\sum_{i=1}^k b_i X_i\|_2 \geq t] \leq (d_1 + d_2)\exp(\frac{-t^2}{2\sigma^2})\,.$$

The third concentration inequality is a result for combinatorial polynomials due to Schudy and Sviridenko [SS12] that is the culmination of an influential line of work begun by Kim and Vu [KV00].

Fact 3.4.3 (Concentration of polynomials, Theorem 1.2 in [SS12], specialized). *Let $H \subseteq \binom{[n]}{t}$ be a collection of multilinear monomials of degree t in n $\{0,1\}$-valued variables, and let $f(x) := \sum_{C \in H} \prod_{i \in C} x_i$. Let $Y_1, Y_2, \ldots, Y_n$ be independent and identically distributed Bernoulli random variables with $\Pr[Y_i = 1] = \tau$. Then, for some absolute constant $R \geq 1$,*

$$\Pr[|f(Y) - \mathbb{E}f(Y)| \geq \lambda] \leq e^2 \max\left\{ \max_{r=1,2,\ldots,t} e^{-\lambda^2/\nu_0 \nu_r R^t}, \max_{r=1,2,\ldots,t} e^{-(\frac{\lambda}{\nu_r R^t})^{1/r}} \right\},$$

where, for every $r \leq t$, $\nu_r = \tau^{t-r} \max_{h_0 \subseteq [n], |h_0| = r} |\{h \in H : h \supseteq h_0\}|$.

Fact 3.4.4 (Chernoff bound). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0,1\}$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then, for any $\delta \in [0,1]$,*

$$\Pr\{|X - \mu| \geq \delta\mu\} \leq 2e^{-\delta^2\mu/3}\,.$$

Fact 3.4.5 (Matrix Chernoff [Tro15, Theorem 5.1.1]). *Let $X_1, \ldots, X_n \in \mathbb{R}^{d \times d}$ be independent, random, symmetric matrices such that $X_i \geq 0$ and $\lambda_{\max}(X_i) \leq R$ almost surely. Let $X = \sum_{i=1}^n X_i$ and $\mu =$*

$\lambda_{\max}(\mathbb{E}[X])$. *Then, for any* $\delta \in [0,1]$,

$$\Pr\{\lambda_{\max}(X) \geq (1+\delta)\mu\} \leq d \cdot \exp\left(-\frac{\delta^2\mu}{3R}\right).$$

3.5 The sum-of-squares algorithm

We briefly define the key sum-of-squares facts that we use. These facts are all taken from [BS16, FKP19].

Definition 3.5.1 (Pseudo-expectations over the hypercube)**.** A degree d pseudo-expectation $\tilde{\mathbb{E}}$ over $\{-1,1\}^n$ is a linear operator that maps degree $\leq d$ polynomials on $\{-1,1\}^n$ into real numbers with the following three properties:

1. (Normalization) $\tilde{\mathbb{E}}[1] = 1$.
2. (Booleanity) For any x_i and any polynomial f of degree $\leq d-2$, $\tilde{\mathbb{E}}[fx_i^2] = \tilde{\mathbb{E}}[f]$.
3. (Positivity) For any polynomial f of degree at most $d/2$, $\tilde{\mathbb{E}}[f^2] \geq 0$.

We note that if $\mathbb{E}$ is the expectation operator of a distribution over $\{-1,1\}^n$, then $\mathbb{E}$ is a degree d pseudo-expectation (for any d), and thus $\max_{x\in\{-1,1\}^n} f(x) \leq \max_{\tilde{\mathbb{E}}} \tilde{\mathbb{E}}[f]$, where the second max is taken over all degree d pseudo-expectations $\tilde{\mathbb{E}}$.

The SoS algorithm shows that we can efficiently maximize $\tilde{\mathbb{E}}[f]$ over degree d pseudo-expectations $\tilde{\mathbb{E}}$ for a polynomial f.

Fact 3.5.2 (Sum-of-squares algorithm, Corollary 3.40 in [FKP19])**.** *Let $f(x_1,\ldots,x_n)$ be a polynomial of degree k, where the coefficients of f are rational numbers with* poly(n) *bit complexity. Let $d \geq k$. There is an algorithm that, on input f, d, runs in time $n^{O(d)}$ and outputs a value α such that $\beta + 2^{-n} \geq \alpha \geq \beta$, where β is the maximum, over all degree d pseudo-expectations $\tilde{\mathbb{E}}$ over $\{-1,1\}^n$, of $\tilde{\mathbb{E}}[f]$.*

We now list the other key properties of pseudo-expectations that we will use. First, we note that pseudo-expectations satisfy the Cauchy-Schwarz inequality.

Fact 3.5.3 (SoS Cauchy-Schwarz inequality)**.** *Let f, g be polynomials with* $\deg(f), \deg(g) \leq d/2$*, and let $\tilde{\mathbb{E}}$ be a degree d pseudo-expectation. Then $\tilde{\mathbb{E}}[fg] \leq \sqrt{\tilde{\mathbb{E}}[f^2]\tilde{\mathbb{E}}[g^2]}$.*

Next, we observe that SoS captures Grothendieck's inequality, which we recall below.

Fact 3.5.4 (Grothendieck's inequality)**.** *Let A be an $n \times n$ matrix and let $s = \max_{Z\in\mathbb{R}^{n\times n}, Z\succeq 0, Z_{i,i}=1\forall i} \operatorname{tr}(A \cdot Z)$. Then, $s \leq K_G \|A\|_{\infty\to 1}$, where $K_G \leq 1.8$ is a universal constant independent of A.*

Fact 3.5.5 (SoS "knows of" Grothendieck)**.** *Let $A \in \mathbb{R}^{n\times n}$. Let $\tilde{\mathbb{E}}$ be a pseudo-expectation over $\{-1,1\}^n$ of degree ≥ 2. Then*

$$\tilde{\mathbb{E}}[x^\top A x] \leq K_G \|A\|_{\infty\to 1} \leq 1.8\|A\|_{\infty\to 1}.$$

Proof. Since $\tilde{\mathbb{E}}$ is a pseudo-expectation of degree ≥ 2, the pseudo-moment matrix $\tilde{\mathbb{E}}[xx^\top] \succeq 0$. Further, since $\tilde{\mathbb{E}}$ is over $\{-1,1\}^n$, $\tilde{\mathbb{E}}[x_i^2] = 1$ for every $i \in [n]$. Thus, the matrix $Z = \tilde{\mathbb{E}}[xx^\top] \succeq 0$, and has $Z_{i,i} = 1$. Applying Fact 3.5.4 completes the proof. □

Fact 3.5.6 (SoS "knows" spectral norm bounds)**.** *Let $A \in \mathbb{R}^{n\times n}$. Let $\tilde{\mathbb{E}}$ be a pseudo-expectation over $\{-1,1\}^n$ of degree ≥ 2, and let W be a symmetric PSD matrix. Then*

$$\tilde{\mathbb{E}}[x^\top A x] \leq \|W^{-1/2}AW^{-1/2}\|_2 \tilde{\mathbb{E}}[x^\top W x] \leq \|W^{-1/2}AW^{-1/2}\|_2 \cdot \operatorname{tr}(W).$$

Proof. Since $\tilde{\mathbb{E}}$ is a pseudo-expectation of degree ≥ 2, the pseudo-moment matrix $\tilde{\mathbb{E}}[xx^\top] \succeq 0$. Further, since $\tilde{\mathbb{E}}$ is over $\{-1,1\}^n$, $\tilde{\mathbb{E}}[x_i^2] = 1$ for every $i \in [n]$. Thus, the matrix $Z = \tilde{\mathbb{E}}[xx^\top] \succeq 0$, and has $Z_{i,i} = 1$.

We then have that

$$\tilde{\mathbb{E}}[x^\top Ax] = \tilde{\mathbb{E}}[(W^{1/2}x)^\top W^{-1/2}AW^{-1/2}(Wx)] \leq \|W^{-1/2}AW^{-1/2}\|_2 \tilde{\mathbb{E}}[x^\top Wx] \leq \|W^{-1/2}AW^{-1/2}\|_2 \mathrm{tr}(W)\,,$$

where the first inequality holds because for any matrix B, the matrix $\|B\|_2 \cdot \mathbb{I} - B$ is PSD, and the second inequality holds because $\tilde{\mathbb{E}}[x^\top Wx] = \sum_{i=1}^n Z_{i,i}W_i = \mathrm{tr}(W)$. □

Finally, we observe that $\tilde{\mathbb{E}}[f] \geq 0$ holds for all nonnegative f on k variables, provided that the degree d is at least $2k$.

Fact 3.5.7. *Let $f(x_1, \ldots, x_k)$ be a* non-negative *degree $\leq k$ multilinear polynomial in $x_1, \ldots, x_k$, i.e., $f(x_1, \ldots, x_k) \geq 0$ for all $x_1, \ldots, x_k \in \{-1,1\}^k$. Let $\tilde{\mathbb{E}}$ be a pseudo-expectation of degree d over $\{-1,1\}^n$, where $d \geq 2k$. Then, $\tilde{\mathbb{E}}[f] \geq 0$.*

3.6 Facts about binomial coefficients

Fact 3.6.1. *Let n, ℓ, q be positive integers such that $n/2 \geq \ell \geq q$. Then, $e^{3q}(\ell/n)^q \geq \binom{n-2q}{\ell-q}/\binom{n}{\ell} \geq e^{-3q}(\ell/n)^q$.*

Proof. We compute the ratio

$$\binom{n-2q}{\ell-q}/\binom{n}{\ell} = \frac{(n-2q)!}{(\ell-q)!(n-\ell-q)!} \cdot \frac{\ell!(n-\ell)!}{n!} = \binom{n-\ell}{q}\binom{\ell}{q}/\binom{2q}{q}\binom{n}{2q}\,.$$

This implies that

$$\binom{n-2q}{\ell-q}/\binom{n}{\ell} \leq e^{2q}\left(\frac{n-\ell}{q}\right)^q\left(\frac{\ell}{q}\right)^q \cdot 2^{-q}\left(\frac{n}{2q}\right)^{-2q} \leq e^{2q}q^{-2q}2^{-q}(2q)^{2q}\left(\frac{n-\ell}{n}\right)^q\left(\frac{\ell}{n}\right)^q \leq e^{3q}\left(\frac{\ell}{n}\right)^q\,,$$

and that

$$\binom{n-2q}{\ell-q}/\binom{n}{\ell} \geq \left(\frac{n-\ell}{q}\right)^q\left(\frac{\ell}{q}\right)^q \cdot 2^{-2q}\left(\frac{en}{2q}\right)^{-2q} = e^{-2q} \cdot \left(\frac{n-\ell}{n}\right)^q\left(\frac{\ell}{n}\right)^q \geq e^{-2q}2^{-q}\left(\frac{\ell}{n}\right)^q \geq e^{-3q}\left(\frac{\ell}{n}\right)^q\,,$$

where we use that $\ell \leq n/2$. Throughout, we use that $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$. □

Fact 3.6.2. *Let n, ℓ, q be positive integers such that $n/2 \geq \ell \geq q$. Then, $e^{2q}(\ell/n)^q \geq \binom{n}{\ell-q}/\binom{n}{\ell} \geq e^{-q}(\ell/n)^q$.*

Proof. We compute the ratio

$$\binom{n}{\ell-q}/\binom{n}{\ell} = \frac{n!}{(\ell-q)!(n-\ell+q)!} \cdot \frac{\ell!(n-\ell)!}{n!} = \binom{\ell}{q}/\binom{n-\ell+q}{q}.$$

This implies that

$$\binom{n}{\ell-q}\Big/\binom{n}{\ell} \leq \frac{e^q \ell^q}{(n-\ell+q)^q} \leq e^q \cdot \left(\frac{2\ell}{n}\right)^q \leq \frac{e^{2q}\ell^q}{n^q}, \text{ and}$$
$$\binom{n}{\ell-q}\Big/\binom{n}{\ell} \geq \frac{e^{-q} \ell^q}{(n-\ell+q)^q} \geq \frac{e^{-q}\ell^q}{n^q},$$

where we use that $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$. □

Fact 3.6.3. *Let n, r, t, ℓ be integers with $t \leq r$ and $\ell \geq r$. Then, it holds that*

$$\frac{\binom{r}{t} t! \binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \leq \left(1 + \frac{O(\ell^2)}{n}\right) n^t \frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}}.$$

Proof. First, we have that

$$\frac{\binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \leq \left(1 + \frac{O(\ell^2)}{n}\right) \frac{n^\ell}{\ell!} \cdot \frac{n^{\ell-(2r-t)}}{(\ell-(2r-t))!} \cdot \frac{(\ell-r)!}{n^{\ell-r}} \frac{(\ell-r)!}{n^{\ell-r}}$$
$$\leq \left(1 + \frac{O(\ell^2)}{n}\right) n^t \frac{(\ell-r)!}{\ell!} \cdot \frac{(\ell-r)!}{(\ell-(2r-t))!}.$$

We now observe that

$$\binom{r}{t} t! \frac{(\ell-r)!}{\ell!} \cdot \frac{(\ell-r)!}{(\ell-(2r-t))!} = \frac{r!}{(r-t)!} \cdot \frac{(\ell-r)!}{\ell!} \cdot \frac{(\ell-r)!}{(\ell-(2r-t))!}$$
$$= \frac{1}{\binom{\ell}{r}} \cdot \frac{1}{(r-t)!} \cdot \frac{(\ell-r)!}{(\ell-(2r-t))!}$$
$$= \frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}},$$

which finishes the proof. □

Part I

Algorithms for Semirandom and Smoothed Constraint Satisfaction Problems

Chapter 4

Background and Results

Four decades of work in computational complexity has uncovered strong hardness results for constraint satisfaction problems (CSPs) such as k-SAT that leave only a little room for non-trivial efficient algorithms in the *worst case*. Strong hardness of approximation [Hås01] essentially rule out (unless $\mathsf{P} = \mathsf{NP}$) any improvement over simply returning a uniformly random assignment when the input instance is *sparse* (i.e., has $m = O(n)$ constraints on n variables). While there is a polynomial time approximation scheme (PTAS) [AKK95] for maximally dense instances (e.g., with $m = O(n^k)$ constraints for k-SAT), under the exponential time hypothesis [IP01], we can already rule out polynomial time algorithms for $o(n^k)$ dense instances and more generally, $2^{n^{1-\delta}}$ time algorithms for any $\delta > 0$ for $o(n^{k-1})$ dense instances [FLP16].

Search and refutation in the average case. In sharp contrast, in well-studied *average-case* settings, there appears to be significant space for new algorithms and markedly better guarantees for CSPs. CSPs can be studied as two natural problems in such average-case settings: the problem of *refutation* — where we focus on efficiently finding witnesses of unsatisfiability for models largely supported on unsatisfiable instances, and the problem of *search* — where our goal is to find an assignment that the model guarantees is *planted* in the instance.

The refutation problem has been heavily investigated in the past two decades. For *fully random* k-CSPs with uniformly random clause structure (i.e., which variables appear in each clause) and "literal pattern" (i.e., which variables appear negated in each clause), there is a polynomial-time algorithm that, with high probability over the instance, certifies that the instance is unsatisfiable, provided that m is at least $\tilde{O}(n^{k/2})$ [GL03, CGL04, AOW15, BM16, RRS17]. This threshold is far below the $\sim n^k$ hardness threshold of [FLP16] for *worst-case* instances. Furthermore, lower bounds in various restricted models [Fei02, BGMT12, OW14, MW16, BCK15, KMOW17, FPV18] provide some evidence that this threshold might be tight for polynomial time algorithms. Adding to this rich theory is the fascinating work of [FKO06] that shows that random CSPs admit polynomial-time verifiable certificates of non-trivial upper bounds on the value even when $m \sim n^{k/2-\delta_k}$ – i.e., when number of constraints are polynomially smaller than the threshold for efficient refutation.

The search problem for planted models of CSPs has also received a fair bit of attention. The setting naturally arises in the investigation of *local* one-way functions and pseudorandom generators in cryptography. Indeed, the security of the well-known one-way function proposed by Goldreich [Gol00] (also conjectured to be a pseudorandom generator [MST06, App16]) is equivalent to the hardness of recovering a satisfying assignment planted (via a carefully chosen

procedure) in a random CSP instance with an appropriate predicate. This has led to significant research on solving *fully random* planted CSPs [BHL+02, JMS07, BQ09, CCF10, FPV15]. Specifically, Feldman, Perkins and Vempala [FPV15] showed that for *fully random* planted k-CSPs with planted assignment x^*, there is a polynomial-time algorithm that, with high probability over the instance, recovers the planted assignment x^* *exactly*, provided that the instance has at least $\tilde{O}(n^{k/2})$ constraints. That is, the refutation and search versions have the same clause threshold.

***Beyond* the average case: semirandom and smoothed instances.** The phenomenal progress in average-case algorithm design notwithstanding, there is a nagging concern that the algorithms that have been developed rely too heavily on "brittle" properties of a specific random model. That is, the methods may have "overfitted" to the specific setting of random CSPs, and thus the resulting algorithms only apply in this limited setting. Unfortunately, this fear turns out to be rather well-founded — natural spectral algorithms for refuting random k-CSPs and solving the natural planted variants break down under minor perturbations, including very weak modifications to the input model such as the introduction of a vanishingly small fraction of additional clauses.

Motivated by such concerns, Blum and Spencer [BS95] and later Feige and Kilian [FK01, Fei07] and Spielman and Teng [ST03] introduced *semirandom* and *smoothed* models for optimization problems. In semirandom models, the instances are constructed by a combination of benign average-case and adversarial worst-case choices; in smoothed models, the instances are constructed by applying only a small perturbation to an otherwise worst-case input. Algorithms that succeed for such models are naturally "robust" to perturbations of the input instance.

For CSPs, a *semirandom* instance is generated by first choosing a "worst-case" clause structure and then choosing the literal negation patterns in each clause via some sufficiently random (and thus "benign") process. In the model for refutation, the literal negation patterns are chosen uniformly at random, which makes the instance unsatisfiable with high probability. In the planted model, one has to sample these negation patterns carefully to simultaneously ensure that (1) the instance has a planted assignment and (2) the negation patterns do not leak information about the planted assignment in a trivial way.

The *smoothed* model is a generalization of the semirandom model for refutation. In the smoothed model with smoothing probability p, an instance is generated by starting from an arbitrary (i.e., worst-case) instance, and then negating each literal in each clause independently with probability p. Note that when we take $p = 1/2$, we recover the semirandom model for refutation, and in both the semirandom and smoothed models, the clause structure of the instance is worst-case, with the only randomness coming from the literal negation patterns.

4.1 Refuting CSPs in semirandom and smoothed models

We will break our results into two sections, one to discuss the task of refutation and one to discuss solving CSPs in planted models. For the case of refutation, we will focus the discussion on the case of smoothed models, of which the semirandom model is a special case.

In this thesis, we develop new spectral techniques, namely the Kikuchi matrix method, that yield strong refutation algorithms for all smoothed Boolean k-CSPs with (a possibly sharp) trade-off between running time and number of constraints matching that of fully random k-CSPs [RRS17], up to polylogarithmic factors. In particular, our results show that the algorithmic

task of strong refutation in the significantly "randomness starved" setting of smoothed instances is no harder than in a fully random instance.

In Part II, we will use these same techniques to prove Feige's conjectured hypergraph Moore bound, a conjecture on the extremal girth vs. density trade-off for hypergraphs that generalizes the well-known Moore bound for graphs. Our proof uses Kikuchi matrices to give a new *spectral double counting* argument that relates subexponential-time smoothed refutation algorithms and the existence of cycles (even covers) in hypergraphs. As a corollary of the hypergraph Moore bound, we show that there are efficiently verifiable witnesses of unsatisfiability for smoothed instances of all k-CSPs with $m \sim n^{k/2-\delta_k}$ constraints, for some constant δ_k, which is polynomially smaller than the threshold at which efficient refutation algorithms exist even for random k-CSPs. This second result generalizes the work of [FKO06] for random CSPs to the semirandom and smoothed models.

Taken together, our results can be interpreted as suggesting that the worst-case picture of complexity of CSPs arises entirely because of *islands of pathology*: most instances "around" the worst-case hard ones are in fact essentially as easy as random, for both refutation algorithms as well as existence of refutation witnesses. Further, in a precise sense, the difficulty of worst-case instances can be attributed to the worst-case literal patterns, rather than the clause structure.

Our contribution is shown visually in Fig. 4.1. Fig. 4.1 plots the time vs. # constraints trade-off for refuting random and smoothed 3-SAT instances (along with the analogous trade-off for approximation schemes for worst-case instances). Our contribution is the smoothed case (blue line), which shows that smoothed 3-SAT instances can be refuted with the same trade-off as random ones (green line). We also show that there exist efficiently verifiable refutation witnesses for smoothed instances at $n^{1.4}$ constraints (purple line), matching the result for random instances due to [FKO06].

Before we formally state our results, let us recall the standard notation to talk about CSPs.

Definition 4.1.1 (k-ary Boolean CSPs)**.** A CSP instance Ψ with a k-ary predicate $P\colon \{-1,1\}^k \to \{0,1\}$ is a set of m constraints on variables $x_1, \ldots, x_n$ of the form $P(\xi(\vec{C})_1 x_{\vec{C}_1}, \xi(\vec{C})_2 x_{\vec{C}_2}, \ldots, \xi(\vec{C})_k x_{\vec{C}_k}) = 1$. Here, $\vec{C}$ ranges over a collection $\vec{H}$ of *scopes*[1] (a.k.a. clause structure) of k-tuples of n variables and $\xi(\vec{C}) \in \{-1,1\}^k$ are "literal negations", one for each $\vec{C}$ in $\vec{H}$. We let $\mathrm{val}_\Psi(x)$ denote the fraction of constraints satisfied by an assignment $x \in \{-1,1\}^n$, and we define the *value* of Ψ, $\mathrm{val}(\Psi)$, to be $\max_{x\in\{-1,1\}^n} \mathrm{val}_\Psi(x)$.

Definition 4.1.2 (Random, semirandom, and smoothed models for refutation)**.** In a *random* (sometimes, *fully random* in order to disambiguate from related models) instance, H is a collection of m uniformly random and independently chosen k-tuples and the $\xi(C)$'s are chosen uniformly at random and independently from $\{-1,1\}^k$ for each C.

In a *semirandom* instance, H is arbitrary (i.e., worst-case) and $\xi(C) \in \{-1,1\}^k$ are sampled uniformly at random and independently for each C.

In a *smoothed* instance, H is arbitrary (i.e., worst-case) and $\xi(C) \in \{-1,1\}^k$ are obtained by starting with arbitrary (i.e., worst-case) $\xi'(C) \in \{-1,1\}^k$ for each C and then for each C, i, setting $\xi(C)_i = \xi'(C)_i$ with probability 0.99 and $\xi(C)_i = -\xi'(C)_i$ with probability 0.01, independently.

We note that the semirandom model is more general than the random model, and the smoothed model is more general than the semirandom model.

[1] We additionally allow $\vec{H}$ to be a multiset, i.e., that multiple clauses can contain the same ordered set of variables.

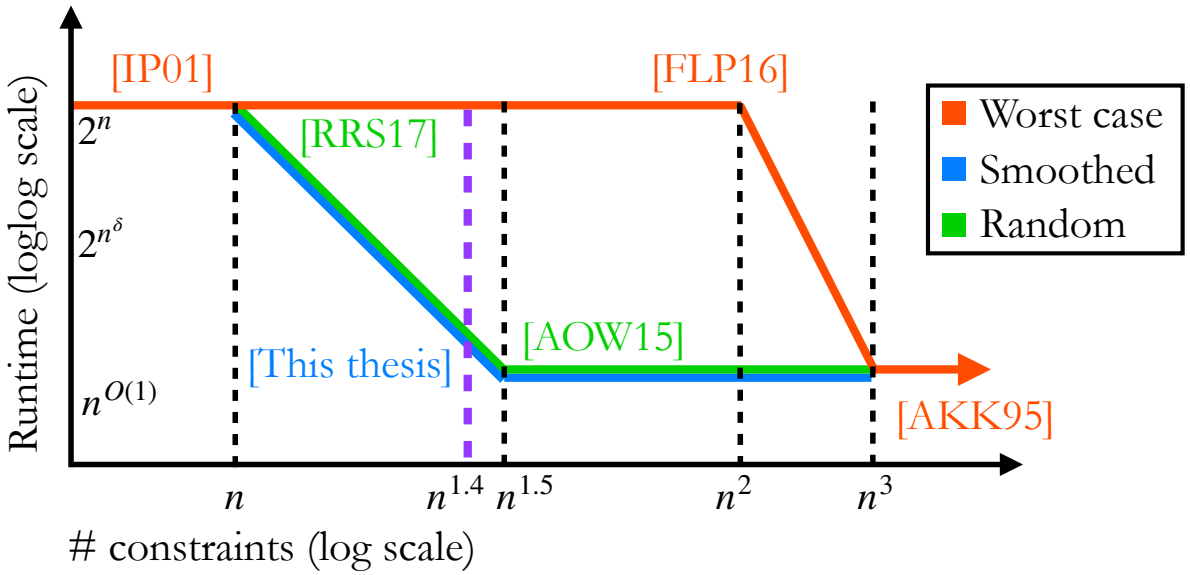

Figure 4.1: Time vs. # constraints trade-off for refuting random and smoothed 3-SAT instances, and for approximation schemes for worst-case instances. The smoothed case is our contribution. We also prove that refutation witnesses exist for smoothed instances at the purple line, i.e., $n^{1.4}$ constraints.

Definition 4.1.3 (Weak, Strong and Tight refutation algorithms)**.** A refutation algorithm takes as input a CSP instance ϕ and outputs a value alg-val$(\phi) \in [0,1]$ with alg-val$(\phi) \geq$ val(ϕ) for all ϕ. For a distribution $\mathcal{D}$ over ϕ, we say that the refutation algorithm *weakly refutes* instances drawn from $\mathcal{D}$ if with high probability over $\phi \sim \mathcal{D}$, alg-val$(\phi) < 1$. We also define *strong refutation* (alg-val$(\phi) < 1 - \delta$ for some absolute constant $\delta > 0$) and *ε-tight refutation* (alg-val$(\phi) <$ val$(\phi) + \varepsilon$, where ε is a parameter of the algorithm that can be made arbitrarily small) analogously.

4.1.1 Algorithms for refuting smoothed CSPs

Our first main result gives a (possibly sharp) trade-off between running time and number of constraints for strongly refuting *smoothed* CSP instances.

Theorem 1 (Smoothed refutation, informal Theorem 5.5.4)**.** *For every $\ell = \ell(n)$, there is a $n^{O(\ell)}$-time strong refutation algorithm for smoothed CSPs with $m \geq m_0 = \tilde{O}(n) \cdot \left(\frac{n}{\ell}\right)^{\left(\frac{t}{2}-1\right)}$ constraints. That is, for any CSP instance ϕ with $m \geq m_0$ constraints, with probability 0.99 over the smoothing ϕ_s of ϕ, the algorithm outputs* alg-val$(\phi_s) \leq 1 - \delta$ *for some absolute constant $\delta > 0$.*

Here, $t = t(P) \leq k$ is the "degree of uniformity" of P – the smallest integer $t \leq k$ such that there is no t-wise uniform distribution (Definition 5.5.3) on $\{-1,1\}^k$ supported entirely on the satisfying assignments $P^{-1}(1) \subseteq \{-1,1\}^k$.

In order to understand the trade-off described by the theorem, let us apply it to two examples.

Example 4.1.4. For k-SAT, P is the Boolean OR function. We thus have $t(P) = k$, as the uniform distribution on odd-parity strings is supported on $P^{-1}(1)$ and is $(k-1)$-wise uniform. Our result gives a polynomial time algorithm to strongly refute smoothed instances of k-SAT whenever the

number of constraints $m \geq \tilde{O}(n^{\frac{k}{2}})$. More generally, for any $\delta > 0$, in time $2^{O(n^\delta)}$ the algorithm strongly refutes smoothed instances with $\geq \tilde{O}(n^{(1-\delta)\frac{k}{2}+\delta})$ constraints.

Example 4.1.5. Consider the "Hadamard predicate" P on $k = 2^{2^q-1}$ bits where $P(x) = 1$ if and only if x is a codeword of the truncated Hadamard code, i.e., x is a truth table of a linear function, excluding the all 0's function. Hadamard CSPs naturally appear in the design of query efficient PCPs. Here, $t(P) = 3 \ll k$, so our theorem gives a polynomial-time algorithm to strongly refute smoothed instances of the Hadamard CSP with at least $\tilde{O}(n^{1.5})$ constraints, and a 2^{n^δ}-time algorithm for instances with at least $\tilde{O}(n^{1.5-\delta/2})$ constraints $\forall \delta \in (0, 1]$.

Comparison with prior results. Theorem 1 can be directly compared to works on refuting random, semirandom and smoothed (in the order of increasing generality) CSPs.

Building on [AOW15, BM16], Raghavendra, Rao and Schramm [RRS17] proved the same trade-off (up to a polylog(n) factor in m) between running time and number of constraints required as in Theorem 1 for the significantly simpler special case of *fully random* CSPs – when the clause structure and the literal patterns are chosen uniformly at random from the respective domains. Our result shows that the same trade-off holds for *smoothed* instances – i.e., with worst-case clause structure and small random perturbations of worst-case literal patterns. All known efficient refutation algorithms, including ours and that of [RRS17], can in hindsight be interpreted as an analysis of the canonical sum-of-squares (SoS) relaxation (Section 3.5) for the max k-CSP problem. For random CSPs (and thus also for the more general smoothed instances we study) the trade-off we obtain is known to be essentially tight [KMOW17, BCK15] for such "SoS-encapsulated" algorithms: this fact is often taken as evidence of sharpness of this trade-off.

Much less is known about refuting CSPs in the more general *semirandom* and *smoothed* models. Feige [Fei07] gave a *weak* refutation algorithm for refuting smoothed and semirandom instances of 3-SAT. His techniques apply to all 3-CSPs but do not seem to extend to either strong refutation or 4-CSPs. More recently, in a direct precursor to this work, Abascal, Guruswami and Kothari [AGK21] gave a polynomial time algorithm for refuting *semirandom* instances of all CSPs – thus obtaining one of the extreme points (corresponding to $\ell = O(1)$) in the trade-off in Theorem 1 above. Theorem 1 relies on a key idea from their work (row bucketing) along with several new ideas discussed below.

Algorithms for refuting *semirandom* k-XOR. Our main technical result is an algorithm for *tight* refutation of *semirandom* instances of k-XOR. Theorem 1 then follows by a simple blackbox reduction (see Section 5.5) that relies on a dual polynomial introduced in [AOW15]. For the special case of k-XOR, an instance ϕ is completely described by an arbitrary k-uniform instance hypergraph H and a collection of "right-hand sides" $b_C \in \{-1, 1\}$, one for each $C \in H$; in the notation of Definition 4.1.1, we have $b_C = \prod_{i=1}^{k} \xi(C)_i$. One can associate to ϕ a homogeneous degree k polynomial $\phi(x)$ on the hypercube $\{-1, 1\}^n$:

$$\phi(x) = \frac{1}{m} \sum_{C \in H} b_C \prod_{i \in C} x_i \,.$$

This polynomial $\phi(x)$ computes the "advantage over 1/2" of an assignment x. That is, the value of the associated instance is $\frac{1}{2} + \frac{1}{2} \max_{x \in \{-1,1\}^n} \phi(x)$. Tight refutation corresponds to certifying that $\phi(x) \leq \epsilon$ for arbitrary $\epsilon > 0$.

Theorem 4.1.6 (Tight refutation of semirandom k-XOR, informal Theorem 5.3.1)**.** *For every $k \in \mathbb{N}$ and $\ell = \ell(n)$ and every $\epsilon > 0$, there is a $n^{O(\ell)}$ time ϵ-tight refutation algorithm for homogeneous degree k*

polynomials that succeeds with probability at least 0.99 *over the draw of the coefficients i.i.d. uniform on* $\{-1,1\}$*, whenever the associated hypergraph* H *has* $m \geq n\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \cdot \mathrm{poly}(\frac{\log n}{\epsilon})$ *hyperedges.*

In particular, for every $\delta > 0$*, we obtain a* $2^{O(n^\delta)}$*-time* ϵ*-tight refutation algorithm for semirandom* k*-XOR instances with* $m \gg \tilde{O}(n) \cdot n^{(1-\delta)(\frac{k}{2}-1)}\mathrm{poly}(\frac{1}{\epsilon})$*-constraints.*

Prior works and brief comparison of techniques. The trade-off above (up to polylog(n) factors in m) matches the one obtained for refuting fully random k-XOR [RRS17]. Our techniques, however, necessarily need to be significantly different, as the analysis in [RRS17] (and related works it built on [CGL04, BM16, AOW15]) crucially rely on the randomness of the hypergraph H. In particular, the refutation in [RRS17] uses the spectral norm of a certain "symmetric tensor power" of the canonical matrix obtained from the instance. They analyze this matrix using a technical tour-de-force argument using the trace moment method.[2] A couple of follow-up works have attempted to simplify the analyses in [RRS17]. Wein, Alaoui and Moore [WAM19] succeeded in giving a simpler proof (introducing the *Kikuchi matrix*, a variant of which is central to this work) for the case of random k-XOR for *even* k, and they also suggest that a natural generalization of their Kikuchi matrix for random odd k will work (their suggestion does not pan out, as we prove in Section 5.6). In a recent work, Ahn [Ahn20] simplified some aspects of the analysis of the "symmetric tensor power" matrix in the analysis of [RRS17]. To summarize, the tools in prior works on random CSPs for analyzing the spectra of relevant correlated random matrices seem to use the randomness of the hypergraph both heavily and in a rather opaque manner.

For the more general setting of semirandom k-XOR refutation, the best known result [AGK21] obtained an extreme point in the trade-off (i.e., the case of $\ell = O(1)$). That work analyzes the $\infty \to 1$-norm of the canonical matrix associated with the CSP instance. In this special case when $\ell = O(1)$, it turns out that handling 3-XOR instances allows deriving all larger k as a corollary. For the case of 3-XOR, their analysis relies on a new *row bucketing* step according to the *butterfly degree* of a pair of vertices (a new notion that they define), along with a certain pseudo-random vs structure decomposition for arbitrary 3-uniform hypergraphs associated with the 3-XOR instance.

To prove Theorem 4.1.6, we build on [AGK21] and introduce a few new tools. For even k, the Kikuchi matrix of [WAM19] analyzed using the row bucketing idea (with an appropriate generalization of the butterfly degree) of [AGK21] yields a correct trade-off (see Section 2.2). The case of odd k turns out to be significantly more challenging (as has always been the case in CSP refutation) and needs new ideas. We introduce a variant of the Kikuchi matrix for this purpose. Unlike the case of even k (and the algorithm in [AGK21]), the spectral norm of this matrix is provably too large to yield a refutation, even for *random* instances. Indeed, this is why the strategy suggested by [WAM19] fails, as we show in Section 5.6. Instead, we use the row pruning strategy (Section 2.3) and refute the instance using the spectral norm of a matrix obtained by pruning away appropriately chosen rows. We then show that the number of pruned rows is not too large, and so does not contribute too much to the $\infty \to 1$-norm of the full matrix.

The row pruning step motivates a definition of *regularity*, a collection of natural pseudorandom properties that relate to *well-spreadness* in the intersection structure of the hyperedges in the instance hypergraph.[3] We then show that the hyperedges in every k-uniform hypergraph can be decomposed, via a *regularity decomposition* lemma, into k'-uniform hypergraphs for $k' \leq k$,

[2] Just the technical argument in [RRS17] runs over 20 pages!

[3] This is closely related to the notion of spread encountered in recent work on the sunflower conjecture [ALWZ20, Rao23].

along with some "error" hyperedges, such that (i) each of the k'-uniform hypergraphs satisfies regularity, and (ii) refuting all of these k'-XOR instances provides a refutation for the original instance. We explain our row pruning and the regularity decomposition steps in more detail in Section 5.1.

4.1.2 Refutation witnesses for smoothed CSPs below the spectral threshold

In a one-of-a-kind result, Feige, Kim and Ofek [FKO06] (henceforth, FKO) proved that with high probability over the draw of a fully random 3-SAT instance ψ, there is a polynomial size *witness* that weakly refutes ψ if ψ has $m \sim \tilde{O}(n^{1.4})$ constraints. Formally, there is a polynomial time *non-deterministic* refutation algorithm that succeeds in finding a refutation with high probability over the drawn of a fully random 3-SAT instance with $m \sim \tilde{O}(n^{1.4})$ constraints. On the other hand, all known polynomial time *deterministic* refutation algorithms require the input random instance to have $\Omega(n^{1.5})$ constraints – this bound is often called the *spectral threshold*. The fastest known refutation algorithm [RRS17] for instances with $\sim n^{1.4}$ constraints runs in time $2^{n^{0.2}}$, matching the SoS lower bound [KMOW17]. Thus, intriguingly, the FKO result shows the existence of polynomial time verifiable refutation witnesses (i.e., certificates of an upper bound of $1 - o_n(1)$ on the value) at a constraint density at which there are no known $2^{n^{o(1)}}$-time refutation algorithms. Does such a "gap" between thresholds for existence vs efficient computability of refutation witnesses persist for semirandom and smoothed instances, i.e., instances with *worst-case* constraint hypergraphs?

In 2008, Feige [Fei08] made an elegant conjecture on the existence of even covers in sufficiently dense hypergraphs. This conjecture can be interpreted as generalizing to hypergraphs the classical Moore bound on the girth of graphs with a given number of edges. If true, Feige's conjecture implies that the FKO result holds for all semirandom and smoothed CSP instances – in particular, the FKO result does not rely on the properties of the underlying hypergraph at all.

In Part II of this thesis, we will prove this conjecture. Combining this result with our smoothed refutation algorithms (Theorem 1), we immediately obtain a generalization of the FKO result that yields a polynomial time non-deterministic refutation algorithm for smoothed instances of all k-ary CSPs with number of constraints m polynomially below the spectral threshold of $n^{k/2}$.

Theorem 2 (Informal Theorem 6.0.2)**.** *There is a* non-deterministic *polynomial time algorithm that weakly refutes smoothed instances of any k-CSP with $m \geq m_0 = \tilde{O}(n^{\frac{k}{2} - \frac{k-2}{2(k+8)}})$-constraints. For the special case of $k = 3$, $m_0 = \tilde{O}(n^{1.4})$.*

4.2 Solving planted CSPs in semirandom models

In this section, we discuss our results for solving CSPs in planted models.

In this thesis, we give an algorithm for the *search* variant of CSPs in the semirandom setting. Our result gives an efficient algorithm for solving semirandom planted CSPs that succeeds in finding the planted assignment whenever the number of constraints exceeds $\tilde{O}(n^{k/2})$ — the *same* threshold at which polynomial time algorithms exist for the refutation problem for random (and semirandom) instances.

Theorem 3 (Algorithm for planted CSPs, informal Theorem 4)**.** *There is an efficient algorithm that takes as input a k-CSP Ψ and outputs an assignment x with the following guarantee: if Ψ is a semirandom*

planted k-CSP with $m \geq \tilde{O}(n^{k/2})$ constraints, then with high probability over Ψ, the output x satisfies $\mathrm{val}_\Psi(x) \geq 1 - o(1)$, i.e., x satisfies $1 - o(1)$-fraction of the constraints in Ψ.

We note that in the semirandom setting, it is not possible to efficiently recover an assignment that satisfies *all* of the constraints without being able to do so even when $m = O(n)$.[4] This is because it is easy to construct a semirandom instance Ψ that is the "union" of two disjoint instances Ψ_1 and Ψ_2, where Ψ_1 and Ψ_2 use disjoint sets of $n/2$ variables, but Ψ_1 only has $m_1 \sim O(n)$ clauses (and Ψ_2, therefore, contains almost all of the $m \sim n^{k/2}$ clauses). Thus, the guarantee in Theorem 3 of satisfying a $1 - o(1)$-fraction of constraints is qualitatively the best we can hope for.

Search vs. refutation. It is natural to compare Theorem 3 to the problem of *refuting* semirandom CSPs discussed in Section 4.1 [AGK21, GKM22, HKM23]. For average-case optimization problems, techniques for refuting random instances can typically be adapted to solving the search problem in the related planted model. This can be formalized in the *proofs to algorithms* paradigm [BS14, FKP19] where spectral/SDP-based refutations can be transformed into "simple" (i.e., "captured" within the low-degree sum-of-squares proof system) efficient certificates of near-uniqueness of optimal solution — that is, every optimal solution is close to the planted assignment. Unfortunately, this intuition breaks down even in the simplest setting of semirandom 2-XOR where there can be multiple maximally far-off solutions that satisfy as many (or even more) constraints as the planted assignment. Such departure from uniqueness also breaks algorithms for recovery [FPV15] that rely on the top eigenvector of a certain matrix built from the instance being correlated with the planted assignment. In the semirandom setting, one can build instances where the top eigenspace of such matrices is the span of the multiple optimal solutions and has dimension $\omega(1)$ (searching for a Boolean vector close to the subspace is, in general, hard in super-constant dimensional subspaces).

Our key insight. Our starting point is a new, efficiently checkable certificate of the unique identifiability of the planted solution for noisy planted k-XOR (i.e., where each equation in a satisfiable k-sparse linear system is corrupted independently with some fixed constant probability) whenever the constraint hypergraph satisfies a certain weak expansion property. For random graphs in case of 2-XOR (and generalizations to multiple community *stochastic block models*), such certificates (in the form of explicit dual solutions to a semidefinite program) were shown to exist in [ABH16, MNS15]; these two works independently discovered the threshold for exact recovery for 2-community SBMs.

Our certificate naturally yields an efficient algorithm for *exactly* recovering the planted assignment in noisy k-XOR instances whenever the constraint hypergraph satisfies a deterministic weak expansion property and has size exceeding the refutation threshold $\sim n^{k/2}$. Finally, we use expander decomposition procedures to decompose the input constraint hypergraph into pieces that satisfy the above condition. This is done in a manner that further allows us to find a good assignment via a consistent patching scheme to combine solutions across all the pieces in our decomposition.

[4] Achieving this would break a hardness assumption for the search problem analogous to Feige's random 3-SAT hypothesis for the refutation problem [Fei02].

4.2.1 Our semirandom planted model and results

Before formally stating our results, we define the semirandom planted model that we work with and explain some of the subtleties in the definition. Our model is the natural one that arises if we wish to enforce independent randomness (for each clause) in the literal negations, while still fixing a particular satisfying assignment. Recall that we have formally defined a k-CSP in Definition 4.1.1.

Definition 4.2.1 (Semirandom planted k-ary Boolean CSPs)**.** Let $P\colon \{-1,1\}^k \to \{0,1\}$ be a predicate. We say that a distribution Q over $\{-1,1\}^k$ is a *planting distribution for* P if $\Pr_{y \leftarrow Q}[P(y) = 1] = 1$.

We say that an instance Ψ with predicate P is a *semirandom planted instance* with *planting distribution* Q if it is sampled from a distribution $\Psi(\vec{H}, x^*, Q)$ where

(1) the scopes $\vec{H} \subseteq [n]^k$ and planted assignment $x^* \in \{-1,1\}^n$ are arbitrary, and

(2) $\Psi(\vec{H}, x^*, Q)$ is defined as follows: for each $\vec{C} \in \vec{H}$, sample literal negations $\xi(\vec{C}) \leftarrow Q(\xi(\vec{C}) \odot x^*_{\vec{C}})$, where "$\odot$" denotes the element-wise product of two vectors. That is, $\Pr[\xi(\vec{C}) = \xi] = Q(\xi \odot x^*_{\vec{C}})$ for each $\xi \in \{-1,1\}^k$. Then, add the constraint $P(\xi(\vec{C})_1 x_{\vec{C}_1}, \xi(\vec{C})_2 x_{\vec{C}_2}, \ldots, \xi(\vec{C})_k x_{\vec{C}_k}) = 1$ to Ψ.

Notice that because Q is supported only on satisfying assignments to P, it follows that if $\Psi \leftarrow \Psi(\vec{H}, x^*, Q)$, then x^* satisfies Ψ with probability 1.

A (fully) random planted CSP, e.g., as defined in [FPV15], is generated by first sampling $\vec{H} \leftarrow [n]^k$ uniformly at random, and then sampling $\Psi \leftarrow \Psi(\vec{H}, x^*, Q)$. The difference in the semirandom planted model is that we allow $\vec{H}$ to be *worst-case*.

Notice that in Definition 4.2.1, there are some choices of Q for which the planted instance becomes easy to solve. In the case of, e.g., 3-SAT, one could set the planting distribution Q to be uniform over all 7 satisfying assignments, which results in the literal negations in each clause being chosen uniformly conditioned on x^* satisfying the clause. However, by simply counting how many times the variable x_i appears negated versus not negated and taking the majority vote, we recover x^* with high probability [BHL+02, JMS07] (see Section 7.7).

Instead of sampling clauses uniformly from all those satisfied by x^*, one can create more challenging distributions, e.g., ones where true and false literals appear in equal proportion. Such distributions are termed "quiet plantings" and have been studied extensively [JMS07, KZ09, CCF10, KMZ12]. Our semirandom model follows definitions in [FPV15, FPV18] and is a general planted model with respect to a *planting distribution* Q, which unifies various plantings studied in the past.

Unlike in the case of random planted CSPs, we cannot hope to recover the planted assignment x^* exactly in the semirandom setting. Indeed, the scopes $\vec{H}$ may not use some variable x_i at all, and so we cannot hope to recover x^*_i! Thus, our goal is instead to recover an assignment x that has nontrivially large value, ideally value $1 - \varepsilon$ for arbitrarily small ε. Our result, stated formally below, gives an algorithm to accomplish this task.

Theorem 4 (Formal Theorem 3)**.** *Let $k \in \mathbb{N}$ be constant. There is a polynomial-time algorithm that takes as input a k-CSP Ψ and outputs an assignment x with the following guarantee. If Ψ is a semirandom planted k-CSP with $m \geq c^k n^{k/2} \cdot \frac{\log^3 n}{\varepsilon^9}$ constraints drawn from $\Psi(\vec{H}, x^*, Q)$, then with probability $1 - 1/\mathrm{poly}(n)$ over Ψ, the output x of the algorithm has $\mathrm{val}_\Psi(x) \geq 1 - \varepsilon$. Here, c is a universal constant.*

In particular, setting $\varepsilon = 1/\mathrm{polylog}(n)$*, if* $m \geq \tilde{O}(n^{k/2})$*, then with high probability over* $\Psi \leftarrow \Psi(\vec{H}, x^*, Q)$*, the algorithm outputs* x *with* $\mathrm{val}_\Psi(x) \geq 1 - o(1)$*.*

Theorem 4 shows that one can *nearly* solve a semirandom planted k-CSP at the same $\tilde{O}(n^{k/2})$ threshold as done in the random case [FPV15], matching the same $\tilde{O}(n^{k/2})$ threshold as for semirandom refutation (Theorem 1, [AGK21, GKM22, HKM23]). However, as explained earlier (and will be discussed further in Section 7.1), there are several unanticipated technical hurdles to overcome in the semirandom planted setting that are not present in the semirandom refutation setting, and this causes many of the natural approaches that "springboard off" the refutation case to fail. Curiously enough, for the special case of $k = 2$ there *is* a simple reduction from search to refutation for the case of 2-XOR, which we will describe in Section 7.1.1, but the same approach for k-XOR encounters a hardness barrier for $k \geq 3$, as we will discuss in Section 7.1.2.

Theorem 4 also breaks Goldreich's candidate pseudorandom generators [Gol00] and its variants [App16],[5] when they have $\tilde{\Omega}(n^{k/2})$ stretch and *any* k-hypergraph (not just a random one). In fact, not only does Theorem 4 break the PRG, it also gives an algorithm that nearly *inverts* it.

Noisy planted k-XOR. Similar to work on random planted CSPs [FPV15] and the refutation setting [AOW15, RRS17, AGK21, GKM22, HKM23], our proof of Theorem 4 goes through a reduction to noisy k-XOR. Our algorithm achieves very strong guarantees in the noisy k-XOR case, as we now explain. We define the noisy k-XOR model below and then state our result.

Definition 4.2.2 (Noisy planted k-XOR)**.** Let $H \subseteq \binom{[n]}{k}$ be a k-uniform hypergraph on n vertices, let $x^* \in \{-1,1\}^n$, and let $\eta \in [0, 1/2)$. Let $\psi(H, x^*, \eta)$ denote the distribution on k-XOR instances over n variables $x_1, \ldots, x_n \in \{-1, 1\}$ obtained by, for each $C \in H$, adding the constraint $\prod_{i \in C} x_i = \prod_{i \in C} x_i^*$ with probability $1 - \eta$, and otherwise adding the constraint $\prod_{i \in C} x_i = -\prod_{i \in C} x_i^*$. In the latter case, we say that the constraint C is *corrupted* or *noisy*.

We call ψ a *noisy planted* k-XOR instance if it is sampled from $\psi(H, x^*, \eta)$, for some H, x^*, and η; the hypergraph H is the constraint hypergraph, x^* is the planted assignment, and η is the noise parameter. Furthermore, we let $\mathcal{E}_\psi \subseteq H$ denote the (unknown) set of corrupted constraints.

Theorem 5 (Algorithm for noisy k-XOR)**.** *Let* $\eta \in [0, 1/2)$*, let* $k, n \in \mathbb{N}$*, and let* $\varepsilon \in (0, 1)$*. Let* $m \geq c n^{k/2} \cdot \frac{k^4 \log^3 n}{\varepsilon^5 (1-2\eta)^4}$ *for a universal constant* c*. There is a polynomial-time algorithm* $\mathcal{A}$ *that takes as input a* k*-XOR instance* ψ *with constraint hypergraph* H *and outputs two disjoint sets* $\mathcal{A}_1(H), \mathcal{A}_2(\psi) \subseteq H$ *with the following guarantees: (1) for any instance* ψ *with* m *constraints,* $|\mathcal{A}_1(H)| \leq \varepsilon m$ *and* $\mathcal{A}_1(H)$ *only depends on* H*, and (2) for any* $x^* \in \{-1,1\}^n$ *and any* k*-uniform hypergraph* H *with at least* m *hyperedges, with probability at least* $1 - 1/\mathrm{poly}(n)$ *over* $\psi \leftarrow \psi(H, x^*, \eta)$*, it holds that* $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (H \setminus \mathcal{A}_1(H))$*.*

In words, the algorithm discards a small number of constraints, and among the constraints that are not discarded, correctly identifies all (and only) the corrupted constraints. In particular, the subinstance obtained by discarding the $\lesssim (\varepsilon + \eta)m$ constraints $\mathcal{A}_1(H) \cup \mathcal{A}_2(\psi)$ is satisfiable (and a solution can be found by Gaussian elimination). Thus, Theorem 5 immediately implies that for k-XOR, the NP-hard task of deciding if ψ has value $\geq 1 - \eta$ or $\leq \frac{1}{2} + \eta$ is actually *easy* if ψ has $\sim n^{k/2}$ constraints (far below the $\sim n^k$-hardness of [FLP16]), provided that the η-fraction of corrupted constraints in the "yes" case are a *randomly chosen subset* of the otherwise arbitrary constraints.

Exact vs. approximate recovery. As alluded to above, the guarantees of Theorem 5 are much

[5] Goldreich's original PRG is essentially a planted k-CSP with a Boolean predicate P on a random hypergraph, containing both P and $\neg P$ constraints.

stronger: not only can we find a good assignment to ψ, we can break the constraints into two parts, a small fraction, $\mathcal{A}_1(H)$, where we are unable to determine the corrupted constraints,[6] and a large fraction, $H \setminus \mathcal{A}_1(H)$, where we can determine *exactly* all of the corrupted constraints, $\mathcal{A}_2(\psi)$. Moreover, this partition depends only on the hypergraph H and is *independent of the noise*. We remark that it is not immediately obvious that this guarantee is achievable even for exponential-time algorithms, as x^* may not be the globally optimal assignment with constant probability. This strong guarantee of Theorem 5 is in fact required for the reduction from Theorem 4 to Theorem 5; the weaker (and more intuitive) guarantee of approximate recovery — obtaining an assignment of value $1 - \eta - o(1)$ for the noisy XOR instance — is insufficient for the reduction.

One can view Theorem 5 as an algorithm that extracts almost all the information about the planted assignment x^* encoded by the instance ψ. Indeed, notice that even if $\eta = 0$, the instance ψ only determines x^* "up to a linear subspace."[7] Namely, if we let $y \in \{-1, 1\}^n$ be any solution to the system of constraints $\prod_{i \in C} y_i = 1$ for $C \in H$, then $y \odot x^*$ is also a planted assignment for ψ: formally, $\psi(H, x^*, \eta) = \psi(H, y \odot x^*, \eta)$ as distributions. So, aside from the εm constraints that are discarded, with high probability over ψ the algorithm determines the uncorrupted right-hand sides $\prod_{i \in C} x^*_i$ for every remaining constraint, which is all the information about the planted assignment x^* encoded in the remaining constraints.

The importance of relative spectral approximation. As a key technical ingredient in the algorithm, we uncover a *deterministic* condition — relative spectral approximation of the Laplacian of a graph (associated with the input instance) by a certain correlated random sample from it — which when satisfied implies uniqueness of the SDP solution (Lemma 7.1.4). In Lemma 7.1.5 and Lemma 7.4.7, we establish such spectral approximation guarantees.

This spectral approximation property is the key ingredient in our certificate of unique identifiability of the planted assignment in a noisy k-XOR instance (see Section 7.1.4 for details) and allows us to *exactly* recover the planted assignment for 2-XOR instances where the constraint graph G is a weak spectral expander (i.e., spectral gap $\gg 1/\mathrm{poly}\log n$) (Lemma 7.1.4), and forms the backbone of our final algorithm. We note that our spectral approximation condition can be seen as an analog of (and is, in fact, stronger than) the related spectral norm upper bound property that underlie the refutation algorithm of [AGK21].

This process of extracting a "deterministic property of random instances sufficient for the analysis" is an important conceptual theme underlying recent progress on semirandom optimization, and manifests as, e.g., the notion of "butterfly degree" in [AGK21], "hypergraph regularity" or spreadness in [GKM22] in the context of semirandom CSP refutation, and biclique number bounds in the context of planted clique [BKS22].

[6] Note that discarding a small fraction of constraints is necessary in the semirandom setting, as ψ may contain many disconnected constant-size subinstances where it is not possible, even information-theoretically, to exactly identify the corrupted constraints with $1 - o(1)$ probability.

[7] A k-XOR constraint $x_{C_1} \cdots x_{C_k} = b_C \in \{-1, 1\}$ can be equivalently written as a linear equation $x'_{C_1} + \cdots + x'_{C_k} = b'_C$ over $\mathbb{F}_2$, where we map $+1$ to 0 and -1 to 1.

Chapter 5

Algorithms for Strongly Refuting Smoothed CSPs

In this chapter, we will prove Theorem 1 using the Kikuchi matrix method. As we will show in Section 5.5, to prove Theorem 1, it suffices to prove Theorem 4.1.6. We note that in the case of even k, Theorem 4.1.6 is Theorem 2.0.2, which we have already proven in Section 2.2. Thus, the majority of this chapter will focus on proving Theorem 4.1.6 when k is odd.

In Section 5.1 we will give an overview of the proof for k odd, and then in Sections 5.2 to 5.4, we will present the full proof of Theorem 4.1.6. Finally, in Section 5.5, we will use Theorem 4.1.6 to prove Theorem 1.

5.1 Proof overview: refuting semirandom k-XOR for odd k

The case of odd arity XOR refutation is lot more challenging. Even in the well-studied special case of random k-XOR and the special case of $\ell = O(1)$ (i.e., polynomial time refutation), the case of odd k turns out to be significantly more challenging than the even case. So, let us start by focusing on the case of random 3-XOR first.

Analogous to the case of even k, we would like to begin by finding a simpler argument (compared to [RRS17]) for the special case of *random* 3-XOR using some appropriate variant of the Kikuchi matrix. In fact, [WAM19] attempted this by introducing a variant of the Kikuchi matrix, and suggested an explicit approach (see Section F.1 of [WAM19]) to prove that the spectral norm of that matrix yields a refutation, but unfortunately this approach does not work (see Section 5.6). Indeed, unlike the case of even k, we do not know of any reasonable variant of the Kikuchi matrix whose spectral norm yields a refutation for even *fully random* 3-XOR instances with the expected trade-off.

Instead, we will introduce a variant of the Kikuchi matrix and use it to give a refutation algorithm for *random* 3-XOR instances by relying not on the spectral norm (which is too large) but, instead, the spectral norm of a "pruned" version of the matrix. In other words, to handle the case of random 3-XOR, we will need a variant of the "basic approach" (Section 2.1) along with the "row pruning" method (Section 2.3). Finally, we refute semirandom k-XOR for odd k by adding two new ideas to this approach: *regularity decomposition* and *row bucketing* (Section 2.2).

Bipartite 3-XOR. The Kikuchi matrix we introduce relates directly to a polynomial obtained by

applying the standard "Cauchy-Schwarz trick" to the input polynomial. Consider the polynomial $\psi(x) = \frac{1}{m}\sum_{C\in H} b_C x_C$ associated with a 3-XOR instance described by a 3-uniform hypergraph H with m hyperedges and "right-hand sides" b_C's. Here, for a set R we define $x_R := \prod_{i\in R} x_i$, and in particular, $x_C = \prod_{i\in C} x_i$. For each $C \in H$, let $C_{\min}$ be the minimum indexed element in C (using the natural ordering on $[n]$). Then,

$$\max_{x\in\{-1,1\}^n} \psi(x) \le \max_{x,y\in\{-1,1\}^n} \frac{1}{m}\sum_{C\in H} b_C y_{C_{\min}} x_{C\setminus C_{\min}},$$

where each y_u is formally a new variable, but we think of y_u as equal to x_u. Let us reformulate this expression a bit: let $H_u = \{C \mid C' = (C,u) \in H, C'_{\min} = u\}$. Then,

$$\max_{x\in\{-1,1\}^n} \psi(x) \le \max_{x,y\in\{-1,1\}^n} \frac{1}{m}\sum_{u\in[n]} y_u \sum_{C\in H_u} b_{u,C} x_C .$$

One can think of the RHS as the polynomial associated with a *bipartite* instance of the 3-XOR problem on $2n$ variables, since every constraint uses one y variable and two x variables. Our refutation algorithm works for such bipartite instances more generally.

For such a bipartite instance, using the Cauchy-Schwarz inequality, we can derive:

$$\left(\frac{1}{m}\sum_{u\in[n]} y_u \sum_{C\in H_u} b_{u,C} x_C\right)^2 \le \frac{n}{m^2}\sum_u \sum_{C,C'\in H_u} b_{u,C} b_{u,C'} x_C x_{C'}$$
$$= \frac{nm}{m^2} + \frac{n}{m^2}\sum_u \sum_{C\neq C'\in H_u} b_{u,C} b_{u,C'} x_C x_{C'} := \frac{n}{m} + f(x) \quad (5.1)$$

The first term on the RHS is $\le \epsilon^2/2$ if $m \ge 2n/\epsilon^2$. The second term produces a $\le$ 4-XOR instance.

We thus end up with a 4-XOR instance — an even arity instance — albeit with significantly less randomness than required in the argument from Section 2.1. So, we need some different tools to refute such instances. The first of this is the following variant of the Kikuchi matrix that is designed specifically for "playing well" with the symmetries produced by the squaring step above.

Our Kikuchi matrix. Our Kikuchi matrix is indexed by subsets of size ℓ on a universe of size $2n$, corresponding to two labeled copies of each of the original n x variables. For each $C \in H$, let $C^{(1)}$ be the subset of $[n]\times[2]$ where every variable is labeled with "1", and similarly for $C^{(2)}$. This trick is done to ensure that the clauses $x_{C^{(1)}} x_{C'^{(2)}}$ form a 4-XOR instance, as now $C^{(1)}$ and $C'^{(2)}$ *by definition* cannot intersect.

For even k, the "independent" pieces in the Kikuchi matrix were the matrices A_C, one for each $C \in H$. For odd k, the independence pieces will be A_u, one for each y_u because of the loss of independence due to the Cauchy-Schwarz step above.

Definition 5.1.1 (Kikuchi Matrix, 3-XOR)**.** Let $N = \binom{[2n]}{\ell}$. For every $u \in [n]$, let $A_u \in \mathbb{R}^{N\times N}$ be defined as follows: for each $S,T \subseteq [n]\times[2]$ of size ℓ, we will set $A_u(S,T)$ to be nonzero if there are $C,C' \in H_u$ such that $S \oplus T = C^{(1)} \oplus C'^{(2)}$ and $1 = |S\cap C^{(1)}| = |S\cap C'^{(2)}| = |T\cap C^{(1)}| = |T\cap C'^{(2)}|$. That is, $A_u(S,T)$ is nonzero if each of S,T contain one variable from each of $C^{(1)}$ and $C'^{(2)}$. In that case, we will set $A_u(S,T) = b_{u,C}\cdot b_{u,C'}$. Finally, set $A = \sum_u A_u$.

Equivalently, $A_u(S,T)$ is nonzero if there are $C, C' \in H_u$ such that the 1-labeled (respectively, 2-labeled) elements in S, T have symmetric difference C (C', respectively). This construction is important for the success of our row pruning step (which we will soon discuss) and at the same time ensures that every pair (C, C') of constraints in H_u contributes an equal number of nonzero entries in the Kikuchi matrix A. We note that if we do not introduce the 2 copies of each variable, the number of times a pair (C, C') appears in the matrix would depend on $|C \cap C'|$.

The quadratic forms of A relate to the value of the underlying 4-XOR instance: for $D = 4\binom{2n-4}{\ell-2}$,

$$\mathrm{val}(\phi)^2 \leq \frac{n}{m} + \mathrm{val}(f) \leq \frac{n}{m} + \frac{n}{m^2 D}\left(\max_{z \in \{-1,1\}^N} z^\top A z\right).$$

Bounding $z^\top A z$. In the even arity case, we were able to obtain a refutation at this point by simply using the spectral norm of A to bound the right-hand side above. However, this turns out to provably fail here. To see why, let us define the relevant notion of degree — the count of the number of nonzero entries in each row of A_u:

$$\deg(S) = |\{C, C' \in H_u \mid |S \cap C^{(1)}| = |S \cap C'^{(2)}| = 1\}|$$

Because A_u is itself a random matrix, rather than a random sign times a fixed matrix, we cannot apply the Matrix Khintchine inequality (Fact 3.4.2) anymore. We can, however, still apply the related Matrix Bernstein inequality (Fact 3.4.1), but if we do so, the upper bound on $\|A_u\|_2$ for all u is at least as large as $\sim \max_S \sqrt{\deg(S)}$ and it is not too hard to show that there are S for which this bound is at least ℓ. As a result, the best possible spectral norm upper bound that we can hope to obtain on A is $\Omega(\ell \log_2 N) = \tilde{\Omega}(\ell^2)$, a bound that gives us no non-trivial refutation algorithm.

Row pruning. The key observation that "rescues" this bad bound is the key observation that we made in Section 2.3: $\deg(S)$ cannot be large for too many rows. To see why, consider the random variable that selects a uniformly random $S \in \binom{[2n]}{\ell}$ and outputs $\deg(S)$. This can be well approximated (for our purposes) by a random set where every element is included independently with probability $\sim \ell/2n$. The expectation of $\deg(S)$ on this distribution is $O(1)$. By relying on the fact that $|C \cap C'| = \emptyset$ in H_u for almost all pairs with high probability, $\mathrm{Var}[\deg(S)] = O(1)$. A Chernoff bound yields that the fraction of S for which $|\{C \in H_u \mid |S \cap C| > O(\log n)\}|$ is inverse polynomially small in n. A union bound on all u then shows the fraction of rows that are "bad" for any u is at most an inverse polynomial.

It turns out we can ignore such "bad" rows with impunity. This is because, as we observed in Sections 2.2 and 2.3, we are interested in certifying upper bounds on quadratic forms of A over "flat" vectors again and we can argue that removing "bad" rows cannot appreciably affect them. For the "residual matrix", we can now apply the Matrix Bernstein inequality and finish off the proof!

Extending to semirandom 3-XOR. Looking back, the previous analysis uses that the graphs H_u's obtained from the random 3-uniform hypergraph H satisfy a "spread" condition: there are few to none distinct pairs $C, C' \in H_u$ such that $C \cap C' \neq \emptyset$. This notion of *regularity* is the precise pseudorandom property of H that is enough for our argument (i.e. the row pruning step) above to go through. This immediately poses an issue for the row pruning step, as unlike the case of LDCs highlighted in Section 2.3, where the constituent hypergraphs H_u's were matching, here the H_u's are arbitrary and need not be regular!

For the case of 3-XOR, such a regularity property is relatively easy to ensure by a certain ad hoc argument: if too many pairs $C, C' \in H_u$ happen to share a variable, then, "resolving" them yields a system of 2-XOR constraints. Refutation in the special case of 2-XOR is easy using the Grothendieck inequality; this has been observed in several works, including [Fei07, AGK21]. Indeed, this was roughly the strategy employed in the recent work [AGK21] for the case of $\ell = O(1)$ for semirandom k-XOR. In fact, in the $\ell = O(1)$ regime, it turns out that one can reduce k-XOR for all k to the case of 3-XOR and get the right trade-off; thus, such a decomposition for 3-XOR is enough for the argument of [AGK21] to go through for all k.

A second issue is that the variance term in the application of Matrix Bernstein may become large. This is analogous to the issue with the variance term that appears in the even k case (Section 2.2), which we handled earlier using *row bucketing/reweighting*. The execution here is essentially the same, but now requires bucketing with respect to a different combinatorial parameter called the butterfly degree (generalizing a similar notion in [AGK21]) that controls the variance term in the odd k setting.

5.1.1 Refuting semirandom k-XOR for $k > 3$: hypergraph regularity

When $\ell \gg O(1)$, the case of higher arity k does not reduce to $k = 3$. Once again, working through the case of random k-XOR inspires our more general argument. We work with a generalization of the Kikuchi matrix introduced in the previous section for the case of $k = 3$. When analyzing the row pruning step, we need a significantly stricter notion of *regularity* — we call this (ϵ, ℓ)-regularity — for our row pruning argument to go through.

Hypergraph regularity decomposition. Roughly speaking the notion of (ϵ, ℓ)-regularity (indexed by the parameter ℓ and an accuracy bound ϵ) we need demands that for each subset $Q \subseteq [n]$, the number of hyperedges $C \in H_u$ such that $Q \subseteq C$ is bounded above by an appropriate function of m, n and ℓ. Random hypergraphs H satisfy such a regularity property naturally.

In order to handle arbitrary hypergraphs, we introduce a new *regularity decomposition* for hypergraphs. Our regularity decomposition is based on a certain *bipartite contraction* operation that takes a bipartite hyperedge $(u, C) \in H$ and a subset $Q \subseteq C$ and replaces it with $((u, Q), C \setminus Q)$. This operation should be thought of as "merging" all the elements in Q and u into a new single element (u, Q) and obtaining a smaller arity hyperedge in a variable extended space.

We give a greedy (and efficient) algorithm that starts from a k-uniform hypergraph and repeatedly applies bipartite contraction operations to obtain a sequence of k'-uniform hypergraphs for $k' \leq k$ along with some "error" hyperedges, with the property that each of the k'-uniform hypergraphs produced are (ϵ, ℓ)-regular. Each of the k'-uniform hypergraphs produced is naturally associated with a k'-XOR instance related to the input k-XOR instance. We show that refuting each of these output instances yields a refutation for the original k-XOR instance.

Cauchy-Schwarz even in the even-arity setting. Unlike in the case of 3-XOR where the resulting bipartite 3-XOR instance had an equal number of y and x variables above, the bipartite k'-XOR instances produced via our regularity decomposition are *lopsided* – the number of y variables can be polynomially larger in n than the number n of the x variables. A naive bound on the number of constraints required to refute such instances is too large to yield the required trade-off, even in the case for even k.

Instead (and in contrast to all previous works on CSP refutation), we show that an appropriate application of the "Cauchy-Schwarz" trick above to even-arity k-XOR instances allows us to "kill" the y_u's appearing in the polynomial, leaving us with only a polynomial in the x_i's. This is a rather different usage of the technique; in prior works (and as in the case of 3-XOR highlighted above), it was instead used to build the right "square" matrices for obtaining spectral refutations of the associated CSP instances when k is odd.

5.2 A hypergraph decomposition lemma

We are now ready to start the full proof of Theorem 4.1.6. A key ingredient in our proof is a *regular hypergraph decomposition* algorithm that takes an arbitrary k-uniform hypergraph and decomposes it into a $k-1$ different *regular* sub-hypergraphs (after removing a small fraction of the hyperedges). In this section, we present this decomposition step. We first introduce some notation, and then explain the decomposition.

Definition 5.2.1 (Uniform hypergraphs)**.** A k-uniform hypergraph H on n vertices is a collection H of subsets of $[n]$ of size exactly k. For a set $Q \subseteq [n]$, we define $\deg(Q) := |\{C \in H : Q \subseteq C\}|$.

Remark 5.2.2. We will *not* assume that H is simple, i.e., H can be a multiset. For simplicity, we will abuse notation and let $C \in H$ refer to an element of the *multiset* H. We will say that $C \neq C'$ if C and C' are different elements of the multiset H, even if C and C' are equal as sets, i.e., they are distinct copies of the same element in the underlying set of H. As an example, we use the above definition of $\deg(Q)$ to refer to the number of $C \in H$ with $Q \subseteq C$, *counted with multiplicity*. We encourage the reader to assume that H is simple, and then observe that nothing changes if H is a multiset, and definitions are changed appropriately to count multiplicities.

Our decomposition lemma will decompose a uniform hypergraph into *bipartite* hypergraphs, which we introduce.

Definition 5.2.3 (Bipartite hypergraphs)**.** A p-bipartite t-uniform hypergraph on n vertices is a collection $\{H_u\}_{u\in[p]}$, where each H_u is a collection of subsets of $[n]$ of size exactly $t-1$. We call each H_u, or just u, a *partition* of the bipartite hypergraph. A set $C \in H_u$ corresponds to the hyperedge (u, C). For a set $Q \subseteq [n]$ and $u \in [p]$, we define $\deg_u(Q) := |\{C \in H_u : Q \subseteq C\}|$. When p is clear from context or not relevant, we just use the terminology "bipartite t-uniform hypergraph".

One should think of a bipartite hypergraph $\{H_u\}_{u\in[p]}$ as a hypergraph H on two sets of vertices, $[p]$ and $[n]$, where each hyperedge $(u, C) \in H$ contains one vertex $u \in [p]$ and $k-1$ vertices in $[n]$; for $u \in [p]$, the $(k-1)$-uniform hypergraph H_u contains all hyperedges C such that the hyperedge (u, C) is in the hypergraph H.

Definition 5.2.4 (Hypergraph regularity)**.** We say that a p-bipartite k-uniform hypergraph $\{H_u\}_{u\in[p]}$ is (ε, ℓ)-regular if $\deg_u(Q) \leq \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1-|Q|}, 1)$ for all $Q \subseteq [n]$ of size at most $k-1$ and all $u \in [p]$. For convenience, we will say $\{H_u\}_{u\in[p]}$ is regular when ε, ℓ are clear from context.

Remark 5.2.5 (Regularity is a pseudorandom property). Informally speaking, a collection of k-tuples is regular if the number of k-tuples in H_u that all contain a fixed set of size j is appropriately upper bounded. It is not hard to show that if $H = \cup_{u\in[p]} H_u$ is a *uniformly random* bipartite hypergraph with $p = n$ partitions and $m = \ell(\frac{n}{\ell})^{\frac{k}{2}}$ random k-tuples, then with high probability,

for every $u \in [p], Q$, $\deg_u(Q) \le \max(\frac{m}{pn^{|Q|}}, 1) \cdot O(\log n) \le \max((\frac{n}{\ell})^{\frac{k}{2}-1-|Q|}, 1) \cdot O(\log n)$, which is the same condition of regularity, up to the $O(\log n)$ extra factor. Thus, regularity can be seen as a (weak) pseudorandom property of a bipartite hypergraph.

Next, we define a notion of hypergraph decomposition that we call a bipartite contraction.

Definition 5.2.6 (Bipartite contractions)**.** Let H be a k-uniform hypergraph on n vertices. We say that a pair of subsets (Q, C') (of $[n]$) is a *contraction* of the hyperedge $C \in H$ if $C = Q \cup C'$ and Q, C' are disjoint. It is sometimes useful to think of this pair as denoting a set of size $1 + k - |Q|$, where the first "element" of the set is the entire *set* Q, and the remaining $k - |Q|$ elements come from the set $C \setminus Q$.

A *bipartite contraction* of H is a collection of $k-1$ bipartite hypergraphs $\{H_u^{(t)}\}_{u \in [p^{(t)}]}$ for $t = 2, \dots, k$, along with a set $H^{(1)}$ of "discarded edges" where:

(1) each $\{H_u^{(t)}\}_{u \in [p^{(t)}]}$ is a bipartite t-uniform hypergraph,

(2) each $u \in [p^{(t)}]$ corresponds to a subset $Q_u \subseteq [n]$ of size $k + 1 - t$ (it is possible that $Q_u = Q_{u'}$ for distinct u, u'),

(3) every hyperedge in any $H_u^{(t)}$ is a bipartite contraction of some hyperedge in H, i.e., for every t and any $u \in [p^{(t)}]$ and $R \in H_u^{(t)}$, the set $Q_u \cup R = C$ for some $C \in H$, so that the hyperedge (Q_u, R) is a contraction of C,

(4) every hyperedge C is contracted exactly once, i.e., for each $C \in H$, either $C \in H^{(1)}$ or there exists unique t, $u \in [p^{(t)}], R \in H_u^{(t)}$ such that $Q_u \cup R = C$.

Our hypergraph contraction lemma shows that for any k-uniform hypergraph H, we can efficiently find a bipartite contraction of H such that each of the resulting bipartite hypergraphs is regular.

Lemma 5.2.7 (Hypergraph contraction lemma)**.** *Let H be a k-uniform hypergraph on n vertices with $k \ge 2$ and $|H| = m$. Then, there is a bipartite contraction of H such that*

(1) $m^{(1)} := |H^{(1)}| \le \frac{n}{k\varepsilon^2} \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}$.

(2) For $t \ge 2$, each bipartite t-uniform hypergraph $\{H^{(t)}\}_{u \in [p^{(t)}]}$ is

(a) (ε, ℓ)-regular,

(b) $|H_u^{(t)}| = m^{(t)}/p^{(t)} = \lfloor \frac{1}{\varepsilon^2} \max((\frac{n}{\ell})^{t-\frac{k}{2}-1}, 1) \rfloor$ *for all $u \in [p^{(t)}]$, where $m^{(t)} := \sum_{u \in [p^{(t)}]} |H_u^{(t)}|$.*

Further, given H, the decomposition itself can be computed by an algorithm running in time $O(n^k |H|^2)$.

Observe that the lemma does not assume any lower bound on m. Indeed if m is too small then we will have $m^{(t)} = 0$ for all $t \ge 2$.

Proof of Lemma 5.2.7. We prove Lemma 5.2.7 by analyzing the following greedy algorithm to construct the bipartite contraction. Before stating the formal algorithm, we first explain the high level idea of the algorithm, as it is very simple.

If H does not have enough hyperedges, then we set $H^{(1)} = H$ and are done. Otherwise, there must be some "violating" set Q: namely, a set Q where $\deg(Q)$ is above a threshold τ (related to the definition of regularity). We choose a "maximal" such violating Q, i.e., no set containing Q is a violation, and then (1) remove an arbitrary τ hyperedges of the form $Q \cup C$ from H, (2) take bipartite contractions $(Q, C \setminus Q)$ of all such hyperedges, and (3) add them all to $H_u^{(k+1-|Q|)}$ where u is "new" partition where $Q_u := Q$. Notice that we may pick the same Q more than once since we only decrease $\deg(Q)$ by τ in one such step. We repeatedly fix such violations greedily until

we cannot and stop. Notice that this procedure is "one-shot" – we do not recursively operate on the $H_u^{(t)}$'s produced, as (we will show) that they are guaranteed to (ϵ, ℓ)-regular by the design of our decomposition procedure.

We now state and analyze the greedy algorithm.

Algorithm 5.2.8.

Given: A k-uniform hypergraph H over n vertices, where $m = |H|$.
Output: A bipartite contraction $\{\{H_u^{(t)}\}_{u\in[p^{(t)}]}\}_{t=2,\dots,k}$ of H.
Operation:

1. **Initialize:** $p^{(t)} = 0$ for $t = 2, \dots, k$.
2. **Fix violations greedily**:
   (a) Find a maximal nonempty violating Q. That is, find $Q \subseteq [n]$ of size $1 \le |Q| \le k-1$ such that $\deg(Q) = |\{C \in H : Q \subseteq C\}| > \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-|Q|}, 1)$, and $\deg(Q') \le \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-|Q'|}, 1)$ for all $Q' \supsetneq Q$.
   (b) Let $q = |Q|$. Let $u = 1 + p^{(k+1-q)}$ be a new "label", and define H' to be an arbitrary subset of $\{C \in H : Q \subseteq C\}$ of size exactly $\lfloor \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-q}, 1)\rfloor$. Let Q be the set Q_u associated with u, and define $H_u^{(k+1-q)} := \{C \setminus Q : C \in H'\}$.
   (c) Set $p^{(k+1-q)} \leftarrow 1 + p^{(k+1-q)}$, and $H \leftarrow H \setminus H'$.
3. If no such Q exists, then put the remaining hyperedges in $H^{(1)}$.

First, we argue that $m^{(1)}$ is small. By construction, $H^{(1)}$ is the set of remaining hyperedges when the inner loop terminates, and so we must have $\deg(\{i\}) \le \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}, 1) = \frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}$ for every $i \in [n]$; we abuse notation and let deg only count hyperedges remaining in H. We then have $\sum_{i\in[n]}\deg(\{i\}) = k|H^{(1)}|$, as every $C \in H^{(1)}$ is counted exactly k times in the sum. Hence, $m^{(1)} \le \frac{n}{k\varepsilon^2}\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}$.

We now argue that for each t, the bipartite hypergraphs $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$ have the desired properties. Fix $t \in \{2, \dots, k\}$. By construction, each $H_u^{(t)}$ has the same size, namely $\lfloor \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1)\rfloor$. It then follows that $m^{(t)} := \sum_{u\in[p^{(t)}]}|H_u^{(t)}| = p^{(t)} \cdot \lfloor \frac{1}{\varepsilon^2}\max\left(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1\right)\rfloor$, and so $p^{(t)} \le \varepsilon^2 m^{(t)}$ and $|H_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}}$. This proves property (b) in Item (2).

It remains to show property (a), that $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$ is (ε, ℓ)-regular. To see this, let $u \in [p^{(t)}]$, and let Q_u be the set associated with the label u. Note that we must have $|Q_u| = k + 1 - t$. Let H' denote the set of constraints in H at the time when u and $H_u^{(t)}$ are added to the bipartite hypergraph. Namely, we have that for every $C \in H_u^{(t)}$, $Q_u \cup C \in H'$. Now, let $R \subseteq [n]$ be a nonempty set of size at most $t-1$. First, observe that if $R \cap Q_u$ is nonempty, then we must have $\deg_u(R) = 0$ (this degree is in the hypergraph $H_u^{(t)}$). Indeed, this is because $C \cap Q_u = \emptyset$ for all $C \in H_u^{(t)}$. So, we can assume that $R \cap Q_u = \emptyset$. Next, we see that $\deg_u(R) \le \deg_{H'}(Q_u \cup R)$ (where $\deg_{H'}$ is the degree in H'), as $Q_u \cup C \in H'$ for every $C \in H_u^{(t)}$. Because Q_u was maximal whenever it was processed in our decomposition algorithm and $Q_u \subsetneq Q_u \cup R$ as R is nonempty

and $R \cap Q_u = \emptyset$, it follows that

$$\deg_{H'}(Q_u \cup R) \le \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-|Q_u \cup R|}, 1) = \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{\frac{k}{2}-|Q_u|-|R|}, 1)$$
$$= \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1-|R|}, 1) \le \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{\frac{t}{2}-1-|R|}, 1),$$

where the last inequality follows because $t - \frac{k}{2} - 1 - |R| \le \frac{t}{2} - 1 - |R|$ always holds, as $t \le k$. This finishes the proof.

Finally, when $R = \emptyset$, we trivially have $\deg_u(\emptyset) = |H_u^{(t)}| = \lfloor \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1) \rfloor \le \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1) \le \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{\frac{t}{2}-1}, 1)$, where we use again that $t - \frac{k}{2} \le \frac{t}{2}$ as $t \le k$.

To argue the runtime bound, we simply observe that each iteration takes $O(|H|n^k)$ time via brute-force, and there are clearly at most $|H|$ iterations. □

5.3 Refuting semirandom sparse polynomials over the hypercube

In this section, we describe an algorithm to tightly refute semirandom instances of homogenous, multilinear degree-k polynomials. Concretely, our algorithm takes as input a homogenous, multilinear degree-k polynomial ϕ in n variables $x_1, \dots, x_n$ and outputs a correct upper bound on $\mathrm{val}(\phi) := \max_{x \in \{-1,1\}^n} \phi(x)$. Whenever the coefficients of the polynomial are generated from independent random probability distributions on $[-1, 1]$ and the (multi-)hypergraph of coefficients has sufficiently many hyperedges, with high probability, the algorithm outputs a value that is smaller than a target ϵ. The guarantees of our algorithm are captured by the theorem below.

Theorem 5.3.1 (Refuting semirandom sparse polynomials)**.** *Let $k \in \mathbb{N}$ and $\ell : \mathbb{N} \to \mathbb{N}$ be a function such that $2(k-1) \le \ell(n) \le n$. There is an algorithm that takes as input a homogeneous, multilinear polynomial ϕ in n variables $x_1, x_2, \dots, x_n$ of total degree k specified by a k-uniform multi-hypergraph H and a collection of rational numbers $\{b_C\}_{C \in H}$:*

$$\phi(x) = \frac{1}{m} \sum_{C \in H} b_C \cdot \prod_{i \le k} x_{C_i}, \tag{5.2}$$

and the algorithm outputs a value $\text{alg-val}(\phi) \in [-1, 1]$ *in time $n^{O(\ell)}$ satisfying the following:*

(1) $1 \ge \text{alg-val}(\phi) \ge \mathrm{val}(\phi)$.

(2) There is an absolute constant $\Gamma > 0$ such that if $n^{\log_2 n} \ge |H| = m \ge m_0 = \Gamma^k \cdot \left(\frac{n}{\ell}\right)^{\frac{k}{2}} \ell \cdot \frac{\log_2 n}{\varepsilon^5}$ and the b_C's are independent, mean 0 random variables supported in $[-1, 1]$, then with probability $1 - 1/\mathrm{poly}(n)$ over the draw of b_C's, it holds that $\text{alg-val}(\phi) \le \varepsilon + 2^{-n}$.

Moreover, our algorithm is "captured" by the canonical degree 2ℓ sum-of-squares relaxation of polynomial maximization problem over the hypercube. Specifically, under the same hypothesis on ϕ as above, for every pseudo-expectation $\tilde{\mathbb{E}}$ of degree $\ge 2\ell$ over $\{-1, 1\}^n$, it holds that $\tilde{\mathbb{E}}[\phi] \le \epsilon$.

As is the case in Section 5.2, we will *not* assume that H is simple, and we will adopt the same notational conventions as in Remark 5.2.2.

5.3.1 Regular bipartite polynomials

Our proof of Theorem 5.3.1 goes via a reduction to refuting sparse polynomials with additional structure that we call *bipartite* polynomials. Bipartite polynomials can be seen as a generalization of partitioned 2-XOR instances introduced in [AGK21]. We next present this class of polynomials and identify a *regularity* property of such polynomials that will be a key technical ingredient in our algorithm.

Definition 5.3.2 (p-bipartite polynomials)**.** Let $k \in \mathbb{N}$. A p-bipartite polynomial ψ is a homogeneous degree k polynomial in $p+n$ variables $y = \{y_u\}_{u\in[p]}$ and $x = \{x_j\}_{j\in[n]}$ defined by

$$\psi(y,x) = \frac{1}{m}\sum_{u=1}^{p} y_u \sum_{C\in H_u} b_{u,C} x_C\,,$$

where $\{H_u\}_{u\in[p]}$ is a p-bipartite k-uniform hypergraph (Definition 5.2.3), $b_{u,C} \in [-1,1]$ for every $C \in H$, $x_C := \prod_{i\in C} x_i$, and $m := \sum_{u\in[p]}|H_u|$. The *value* of ψ, denoted by $\mathrm{val}(\psi)$, is $\max_{y\in\{-1,1\}^p, x\in\{-1,1\}^n} \psi(y,x)$. Note that $\mathrm{val}(\psi) \in [-1,1]$ always. We also note that ψ is a homogeneous degree 1 polynomial in y.

Definition 5.3.3 (Regular p-bipartite polynomials)**.** We say that a p-bipartite polynomial ψ is (ε,ℓ)-regular if the underlying p-bipartite k-uniform hypergraph $\{H_u\}_{u\in[p]}$ is (ε,ℓ)-regular (Definition 5.2.4). When ε,ℓ are clear from context, we will simply say that ψ is regular.

The bulk of the technical work in proving Theorem 5.3.1 is in analyzing a refutation algorithm for regular instances of p-bipartite polynomials encapsulated in the following theorem.

Theorem 5.3.4 (Refuting regular bipartite polynomials)**.** *Let $k \in \mathbb{N}$. For any $\ell : \mathbb{N} \to \mathbb{N}$ with $2(k-1) \le \ell(n) \le n$ for all $n \in \mathbb{N}$, there is an algorithm with the following properties: the algorithm takes as input a p-bipartite, homogeneous, polynomial $\psi = \psi(y,x)$ in variables $y = \{y_u\}_{u\in[p]}$ and $x = \{x_i\}_{i\in[n]}$ of total degree k:*

$$\psi(y,x) = \frac{1}{m}\sum_{u=1}^{p} y_u \sum_{C\in H_u} b_{u,C} x_C\,,$$

specified by a collection of $(k-1)$-uniform hypergraphs $\{H_u\}_{u\in[p]}$ and rational numbers in $[-1,1]$ $\{b_{u,C}\}_{u\in[p],C\in H_u}$. The algorithm runs in time $(p+n)^{O(\ell)}$ time and outputs $\text{alg-val}(\psi) \in [-1,1]$ satisfying the following:

1. *For every ψ, $\text{alg-val}(\psi) \ge \mathrm{val}(\psi)$.*
2. *Whenever ψ and $b_{u,C}$'s satisfy:*
   (a) *ψ is (ε,ℓ)-regular,*
   (b) *$|H_u| \le \frac{2m}{p}$ for all $u \in [p]$,*
   (c) *$m \ge \max\left\{\Gamma^k \cdot \left(\frac{n}{\ell}\right)^{\frac{k-1}{2}} \sqrt{p\ell\log n}\cdot\frac{1}{\varepsilon^3}, \frac{p}{\varepsilon^2}\right\}$, where Γ is an absolute constant, and*
   (d) *Each $b_{u,C}$ is chosen uniformly at random from $\{-1,1\}$.*

   Then with probability $1 - 1/\mathrm{poly}(n)$ over the draw of $b_{u,C}$'s, $\text{alg-val}(\psi) \le O(\varepsilon) + 2^{-n}$.

Further, our algorithm is "captured" by the sum-of-squares algorithm of degree 2ℓ: for every pseudo-expectation $\tilde{\mathbb{E}}$ in variables x, y of degree 2ℓ over $\{-1,1\}^{p+n}$, $\tilde{\mathbb{E}}[\psi(x,y)] \le O(\varepsilon)$.

We defer the proof of Theorem 5.3.4 to Section 5.4.

5.3.2 Reduction to regular bipartite polynomials

We now use Lemma 5.2.7 along with Theorem 5.3.4 to complete the proof of Theorem 5.3.1 by analyzing the following algorithm:

Main Refutation Algorithm

Algorithm 5.3.5.

Given: A polynomial ϕ specified by a k-uniform multi-hypergraph H over n vertices and rational numbers $\{b_C\}_{C\in H}$.

Output: A value alg-val $\in [-1,1]$.

Operation:

1. Apply the decomposition algorithm from Lemma 5.2.7 to construct bipartite hypergraphs $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$ for $2 \le t \le k$, and a set of discarded edges $H^{(1)}$.
2. For every t, $u \in [p^{(t)}]$ and for every hyperedge $C \in H_u^{(t)}$, set $b_{u,C} = b_{Q_u \cup C}$.
3. For $2 \le t \le k$, apply the refutation algorithm for regular bipartite polynomials from Theorem 5.3.4 to the degree t $p^{(t)}$-bipartite polynomial specified by the bipartite hypergraph $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$ and $b_{u,C}$'s to obtain alg-val$_t$. Set alg-val$_1 = 1$.
4. Output alg-val $= \frac{1}{m}\sum_{t=1^k} m^{(t)} \cdot$ alg-val$_t$, where $m^{(t)} = \sum_{u\in[p^{(t)}]} |H_u^{(t)}|$.

Proof of Theorem 5.3.1 from Lemma 5.2.7 and Theorem 5.3.4. First, without loss of generality we will assume that $\varepsilon \le \frac{1}{\sqrt{2}}$, so that $\frac{1}{\varepsilon^2} \ge 2$. This is without loss of generality, as it only changes the universal constant in Theorem 5.3.1.

For each t and $u \in [p^{(t)}]$, let $Q_u \subseteq [n]$ denote the subset of size $k+1-t$ associated to u, and let ψ_t be the polynomial associated with the t-uniform (ε,ℓ)-regular bipartite hypergraph $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$ obtained from the hypergraph H specifying the input polynomial ϕ by applying the decomposition algorithm from Lemma 5.2.7. Thus, ψ_t is a polynomial in the $p^{(t)}+n$ variables $\{y_u^{(t)}\}_{u\in[p^{(t)}]} \cup \{x_i\}_{i\in[n]}$, and $\psi_t(\{y_u^{(t)}\}_{u\in[p^{(t)}]}, x) := \frac{1}{m^{(t)}}\sum_{u\in[p^{(t)}]} y_u^{(t)} \prod_{C\in H_u^{(t)}} b_{Q_u\cup C} x_C$. We then have that

$$\phi(x) = \frac{1}{m}\sum_{t=2}^{k} m^{(t)}\psi_t(\{x_{Q_u}\}_{u\in[p^{(t)}]}, x) + \frac{1}{m}\sum_{C\in H^{(1)}} b_C x_C\,. \tag{5.3}$$

Indeed, this follows immediately from the definition of a bipartite contraction, because when we substitute x_{Q_u} for y_u for some $u \in [p^{(t)}]$, then $y_u x_C = x_{Q_u\cup C} = x_{C'}$ for $C' \in H$.

Let alg-val$_t$ = alg-val(ψ_t) be the output of the refutation algorithm from Theorem 5.3.4 applied to ψ_t. Then, val$(\psi_t) \le$ alg-val$_t$. Thus, using (5.3), val$(\phi) \le \frac{1}{m}\sum_{t=1}^{k} m^{(t)}$alg-val$_t$ = alg-val.

Next, if for some t, $m^{(t)} \le \varepsilon m$, then using the trivial bound of alg-val$(\psi_t) \le 1$ yields $m^{(t)}$alg-val$(\psi_t) \le \varepsilon m$. Note that in particular, $m^{(1)} \le \varepsilon m$ always holds, as $m \ge \frac{1}{\varepsilon^3}\left(\frac{n}{\ell}\right)^{\frac{k}{2}} \cdot \ell$ and $m^{(1)} \le \frac{n}{k\varepsilon^2}\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}$.

Now, suppose that for some t, $m^{(t)} \ge \varepsilon m$. We now prove that in this setting, $m^{(t)} \ge \Gamma^t \cdot \left(\frac{n}{\ell}\right)^{\frac{t-1}{2}}\sqrt{p^{(t)}\ell} \cdot \frac{(\log_2 n)^{2t+0.5}}{\varepsilon^3}$. We know that $m^{(t)} = p^{(t)} \cdot \lfloor \frac{1}{\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1)\rfloor$. Hence, it suffices to

show

$$\varepsilon m \geq \Gamma^{2t} \cdot \left(\frac{n}{\ell}\right)^{t-1} \ell \cdot \frac{\log_2 n}{\varepsilon^6} \cdot \frac{1}{\frac{1}{2\varepsilon^2}\max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1)},$$

where we use that $\lfloor \frac{1}{\varepsilon^2} \max(\left(\frac{n}{\ell}\right)^{t-\frac{k}{2}-1}, 1) \rfloor \geq \lfloor \frac{1}{\varepsilon^2} \rfloor \geq \frac{1}{2\varepsilon^2}$ as $\frac{1}{\varepsilon^2} \geq 2$.

Hence, for $t \geq \frac{k}{2} + 1$, it suffices to have

$$\varepsilon m \geq 2\Gamma^{2t} \cdot \left(\frac{n}{\ell}\right)^{\frac{k}{2}} \ell \cdot \frac{\log_2 n}{\varepsilon^4},$$

and for $t < \frac{k}{2} + 1$, it suffices to have

$$\varepsilon m \geq 2\Gamma^{2t} \cdot \left(\frac{n}{\ell}\right)^{t-1} \ell \cdot \frac{\log_2 n}{\varepsilon^4}.$$

As $m \geq \Gamma'^k \cdot \left(\frac{n}{\ell}\right)^{\frac{k}{2}} \ell \cdot \frac{\log_2 n}{\varepsilon^5}$, for the absolute constant $\Gamma' = 2\Gamma^2$, both conditions are satisfied.

We have thus shown that if $m^{(t)} \geq \varepsilon m$, then ψ_t satisfies the conditions of Theorem 5.3.4, and so we have $m^{(t)}\text{alg-val}_t \leq \varepsilon m^{(t)} \leq \varepsilon m$ with probability $1 - 1/\text{poly}(n)$ over the draw of b_C's. By union bound over all t, we thus get that $\text{alg-val}(\phi) \leq O(k\varepsilon)$ with probability $1 - k/\text{poly}(n) \geq 1 - 1/\text{poly}(n)$ over the draw of b_C's. This completes the analysis of the second guarantee.

The running time of the algorithm is dominated by the time required to apply the refutation algorithm from Theorem 5.3.4 to each of the bipartite polynomials produced by the decomposition algorithm. This cost is bounded above by $n^{O(\ell)}$.

Finally, the fact that this algorithm is "captured" by SoS follows because Theorem 5.3.4 is "captured" by SoS and the linearity of the pseudo-expectations. □

5.4 Refuting regular bipartite polynomials

In this section, we prove Theorem 5.3.4. Our algorithm is based on the semidefinite programming relaxation of the "$\infty \to 1$"-norm of an appropriate matrix associated with the polynomial ψ. The analysis of the algorithm will naturally establish the "Further,..." part of the statement.

As in several prior works starting with [CGL04], our proof of Theorem 5.3.4 applies the "Cauchy-Schwarz" trick in order to work with an even-degree polynomial associated with ψ.

Lemma 5.4.1 (Cauchy-Schwarz trick)**.** *Let ψ be a p-bipartite, homogeneous, polynomial $\psi = \psi(y, x)$ in variables $y = \{y_u\}_{u \in [p]}$ and $x = \{x_i\}_{i \in [n]}$ of total degree k:*

$$\psi(y, x) = \frac{1}{m} \sum_{u=1}^{p} y_u \sum_{C \in H_u} b_{u,C} x_C.$$

Let f be the following polynomial obtained from ψ:

$$f(x) = \frac{p}{m^2} \sum_{u=1}^{p} \sum_{(C,C') \in H_u \times H_u, C \neq C'} b_{u,C} b_{u,C'} x_C x_{C'}.$$

Then $\text{val}(\psi)^2 \leq \frac{p}{m} + \text{val}(f)$. Further, for every pseudo-expectation $\tilde{\mathbb{E}}$ of degree $\geq 2k$ over $\{-1, 1\}^{p+n}$, $\tilde{\mathbb{E}}[\psi]^2 \leq \frac{p}{m} + \tilde{\mathbb{E}}[f]$.

Proof. Fix an assignment in $\{-1,1\}$ to the y_u's and x_i's. We then have

$$\begin{aligned}\psi^2(y,x) &= \left(\frac{1}{m}\sum_{u=1}^{p} y_u \sum_{C\in H_u} b_{u,C}x_C\right)^2 \le \frac{1}{m^2}\left(\sum_{u=1}^{p} y_u^2\right)\left(\sum_{u=1}^{p}\left(\sum_{C\in H_u} b_{u,C}x_C\right)^2\right)\\ &\le \frac{p}{m^2}\cdot\sum_{u=1}^{p}\sum_{C\in H_u} b_{u,C}^2 x_C^2 + \frac{p}{m^2}\sum_{u\le p}\sum_{(C,C')\in H_u\times H_u, C\neq C'} b_{u,C}b_{u,C'}x_C x_{C'}\\ &\le \frac{p}{m} + \frac{p}{m^2}\sum_{u=1}^{p}\sum_{(C,C')\in H_u\times H_u, C\neq C'} b_{u,C}b_{u,C'}x_C x_{C'},\end{aligned}$$

where the first inequality above uses the Cauchy-Schwarz inequality, the second uses that $y_u^2 = 1$ for every u, and the third uses that $b_{u,C}^2 \le 1$ and $x_C^2 = 1$. Further, observe that by using the SoS version of the Cauchy-Schwarz inequality (Fact 3.5.3) and the fact that $\tilde{\mathbb{E}}$ is over $\{-1,1\}^{p+n}$, we see that the above also holds for all degree $d \ge 2(k-1)$ pseudo-expectations $\tilde{\mathbb{E}}$.

Taking the maximum over x and y on both sides then yields that $\mathrm{val}(\psi)^2 \le \frac{p}{m} + \mathrm{val}(f)$. Taking the maximum over all pseudo-expectations $\tilde{\mathbb{E}}$ on $\{-1,1\}^{p+n}$ and using Fact 3.5.3 yields that $\tilde{\mathbb{E}}[\psi]^2 \le \tilde{\mathbb{E}}[\psi^2] \le \frac{p}{m} + \tilde{\mathbb{E}}[f]$. □

5.4.1 The initial Kikuchi matrix

As Lemma 5.4.1 shows, it suffices to upper bound $\mathrm{val}(f)$. Our certificate of an upper bound on $\mathrm{val}(f)$ is based on an appropriate variant of the Kikuchi matrix of [WAM19]. The definition of the final matrix that we use is rather technical, so we will first define a simpler Kikuchi matrix that will be helpful for intuition and in the analysis. Our final matrix will be obtained by keeping a carefully chosen subset of the entries of the initial matrix.

To define the initial matrix, it is convenient to think of having two clones of each of the n possible "x" variables. For every i, we will use $(i,1)$ and $(i,2)$ to denote the two clones of the i-th variable below. For any set $C \subseteq [n]$, we will use $C^{(1)}$ to denote the set $\{(i,1) \mid i \in C\}$, i.e., the clause C using the first type of clones, and $C^{(2)}$ to be the clause C using the second type of clones. Recall that for any sets S, T, let $S \oplus T$ denote the symmetric difference of the two sets. More generally, let $S_1 \oplus S_2 \oplus \cdots \oplus S_t$ denote the set of all elements that occur in an odd number of different S_i's.

Definition 5.4.2 (Our initial Kikuchi Matrix)**.** Let $\ell \in \mathbb{N}$ and let $N := \binom{2n}{\ell}$.

Fix a p-bipartite k-uniform hypergraph $\{H_u\}_{u\in[p]}$. For each $u \in [p]$, define the $N \times N$ matrix A_u, indexed by sets $S \subseteq [n]\times[2]$ of size ℓ, as follows. For any two sets $S, T \subseteq [n]\times[2]$ of size ℓ and sets $C \neq C' \in H_u$ of size $k-1$, we say that $S \overset{C,C'}{\leftrightarrow} T$ if

1. $S \oplus T = C^{(1)} \oplus C'^{(2)}$,
2. k is odd, and $|S \cap C^{(1)}| = |S \cap C'^{(2)}| = |T \cap C^{(1)}| = |T \cap C'^{(2)}| = \frac{k-1}{2}$, or,
3. k is even, and $|S \cap C^{(1)}| = |T \cap C'^{(2)}| = \frac{k}{2}$ and $|S \cap C'^{(2)}| = |T \cap C^{(1)}| = \frac{k-2}{2}$, or,
4. k is even, and $|S \cap C^{(1)}| = |T \cap C'^{(2)}| = \frac{k-2}{2}$ and $|S \cap C'^{(2)}| = |T \cap C^{(1)}| = \frac{k}{2}$.

Note that $C^{(1)} \oplus C'^{(2)} = C^{(1)} \cup C'^{(2)}$, as $C^{(1)}$ and $C'^{(2)}$ are disjoint by construction.

For $C \neq C' \in H_u$, we define

$$A_{u,C,C'}(S,T) = \begin{cases} 1 \text{ if } S \overset{C,C'}{\leftrightarrow} T, \\ 0 \text{ otherwise.} \end{cases}$$

We then set

$$A_u = \sum_{C \neq C' \in H_u} b_{u,C} b_{u,C'} A_{u,C,C'} \,. \tag{5.4}$$

Note that the sum is over pairs of different elements C, C' of the multiset H (which may nonetheless be equal as sets).

Our (overall) Kikuchi matrix A for the polynomial f is defined as

$$A := \sum_{u=1}^{p} A_u \,. \tag{5.5}$$

The matrix A allows us to write f as a quadratic form, as the following lemma shows.

Lemma 5.4.3. *Let $N := \binom{2n}{\ell}$ and let A be the Kikuchi matrix in Definition 5.4.2 associated with an arbitrary p-bipartite ψ specified by a bipartite hypergraph H and coefficients $\{b_{u,C}\}_{u\in[p],C\in H}$. For any $x \in \{-1,1\}^n$, let $x^{\odot\ell} \in \{-1,1\}^N$ be the vector where the S-th entry of $x^{\odot\ell}$ is $x_S := \prod_{b\in[2]} \prod_{(i,b)\in S} x_i$. Then,*

$$(x^{\odot\ell})^\top A x^{\odot\ell} = \frac{m^2 D}{p} \cdot f(x) \tag{5.6}$$

for D as defined in Eq. (5.9).

Furthermore, since $x^{\odot\ell}$ has ±1-valued entries, for any symmetric PSD matrix $W \succeq 0$, it holds that $\mathrm{val}(f) \leq \frac{p}{m^2 D} \|W^{-1/2} A W^{-1/2}\|_2 \cdot \mathrm{tr}(W)$. *Moreover, for every pseudo-expectation $\tilde{\mathbb{E}}$ of degree $\geq 2\ell$ over $\{-1,1\}^n$,*

$$\tilde{\mathbb{E}}[f] = \frac{p}{m^2 D} \tilde{\mathbb{E}}[(x^{\odot\ell})^\top A x^{\odot\ell}] \leq \frac{p}{m^2 D} \|W^{-1/2} A W^{-1/2}\|_2 \cdot \mathrm{tr}(W) \,.$$

Proof. To see (5.6), observe that by definition of A, if k is odd then every pair (C, C') in H_u with $C \neq C'$ appears exactly $\binom{k-1}{\frac{k-1}{2}}^2 \binom{2n-2(k-1)}{\ell-(k-1)} = D$ times when we expand the LHS. This is because we can choose S by first picking its size $\frac{k-1}{2}$ intersection with $C^{(1)}$ and its intersection with $C'^{(2)}$ ($\binom{k-1}{\frac{k-1}{2}}^2$ choices) and then picking the rest of the set ($\binom{2n-2(k-1)}{\ell-(k-1)}$ choices), and this also completely determines T. A similar calculation yields the value of D when k is even, and so Eq. (5.6) then follows. This is the place where we crucially use the "clones" of the variables to ensure that each pair (C, C') appears the same number of times on the LHS. Without this trick, the number of times a pair (C, C') appears would instead depend on $|C \cap C'|$.

The "furthermore" follows by Fact 3.5.6. □

Below, we summarize the definitions that we have made so far.

Key Notation

1. The input polynomial ψ

$$\psi(y,x) = \frac{1}{m}\sum_{u=1}^{p} y_u \sum_{C\in H_u} b_{u,C}x_C\,, \tag{5.7}$$

is (ε,ℓ)-regular, and p-bipartite, homogeneous of total degree k and is described by a collection of $(k-1)$-uniform hypergraphs $\{H_u\}_{u\in[p]}$ one for every $u \in [p]$ and a collection of rationals $\{b_{u,C}\}_{u\in[p],C\in H_u}$.

2. The polynomial f obtained after the Cauchy-Schwarz trick applied to ψ:

$$f(x) = \frac{p}{m^2}\sum_{u=1}^{p}\sum_{(C,C')\in H_u\times H_u, C\neq C'} b_{u,C}b_{u,C'}x_Cx_{C'}\,, \tag{5.8}$$

is homogeneous of total degree $2(k-1)$. Furthermore, $\mathrm{val}(\psi)^2 \le \mathrm{val}(f) + \frac{p}{m} \le \mathrm{val}(f) + \varepsilon^2$.

3. The Kikuchi matrix $A = \sum_u A_u$ of f is an $N\times N$ matrix for $N = \binom{2n}{\ell}$. The entries of A are indexed by sets $S,T \subseteq [n]\times[2]$ of size ℓ and the entry $A_u(S,T)$ is nonzero (and equal to $b_{u,C}b_{u,C'}$) if and only if $S \overset{C,C'}{\leftrightarrow} T$ for some distinct pair $C,C' \in H_u$. Each pair (C,C') from H_u contributes D nonzero entries in A where

$$D = \begin{cases} \binom{k-1}{\frac{k-1}{2}}^2\binom{2n-2(k-1)}{\ell-(k-1)} & \text{if } k \text{ is odd} \\ 2\binom{k-1}{\frac{k}{2}}\binom{k-1}{\frac{k-2}{2}}\binom{2n-2(k-1)}{\ell-(k-1)} & \text{if } k \text{ is even.}\end{cases} \tag{5.9}$$

Furthermore, $\mathrm{val}(f) \le \frac{p}{m^2D}\|W^{-1/2}AW^{-1/2}\|_2\cdot\mathrm{tr}(W)$ for any symmetric PSD matrix W.

5.4.2 Proof plan

Using Lemma 5.4.3, our task reduces to finding a symmetric PSD matrix W such that $\|W^{-1/2}AW^{-1/2}\|_2\cdot\mathrm{tr}(W) \le \frac{m^2D\varepsilon^2}{p}$ whenever $b_{u,C}$'s are chosen independently at random from $\{-1,1\}$. Our proof proceeds in three conceptual steps:

1. **Row pruning (Section 2.3).** It turns out that the matrix A is not quite sufficient for the analysis to go through. Specifically, there can be rows in the matrix A_u that have ℓ_1-norm that is much larger than the average of $\frac{m^2D}{pN}$. The first step of the proof is to remove rows in each A_u that have too large ℓ_1-norm and show that, by furthermore deleting an extra small set of entries, we are left with a matrix $B = \sum_{u=1}^{p} B_u$ that satisfies all the properties of Lemma 5.4.3 and each B_u has rows with bounded ℓ_1-norm. This is somewhat delicate and crucially relies on regularity of the H_u's. We will prove this by computing conditional first moments, a strategy that is due to [Yan24] and is a generalization of the edge deletion method of row pruning in [HKM23]. The original proof in [GKM22] used a careful application of the celebrated Schudy-Sviridenko polynomial concentration inequality for combinatorial polynomials [SS12].

2. **Row bucketing/reweighting (Section 2.2).** The row pruning ensures that no row has a large ℓ_1-norm in any single B_u. Taking inspiration from spectral analyses of combinatorial random matrices, one might expect that the spectral norm of B after row pruning is upper bounded. However, this turns out not to be true when the H_u's are arbitrary regular hypergraphs. Instead, we show that by reweighting by a careful choice of the PSD matrix W, we can make all the rows/columns have roughly equal contribution to the "variance term" in the reweighted matrix $W^{-1/2}BW^{-1/2}$, which will make it have a good spectral norm. This row reweighting strategy is due to [HKM23], which is a smoother version of the row bucketing strategy employed in [GKM22].

3. **Spectral norm bound.** Our final step involves proving a spectral norm upper bound on $\|W^{-1/2}BW^{-1/2}\|_2$. This is the only step where we use randomness of the right-hand sides b_C's.

Let us now proceed with the details of each of the three steps above.

In the row pruning step, we prove the following lemma.

Lemma 5.4.4 (Row pruned Kikuchi matrices)**.** *Let A be the Kikuchi matrix associated with the polynomial f obtained from an (ϵ, ℓ)-regular p-bipartite polynomial ψ of total degree k defined by $(k-1)$ uniform hypergraphs $\{H_u\}_{u\in[p]}$. Let $\Delta = c^k \frac{1}{\epsilon^4}$ for a sufficiently large absolute constant c. Then, for each $u \in [p]$ and each pair $C \neq C' \in H_u$, there exists a matrix $B_{u,C,C'} \in \{0,1\}^{N\times N}$ such that*

(1) The matrix $B_{u,C,C'}$ is a "subset" of the matrix $A_{u,C,C'}$. Namely, for any pair (S,T), if $B_{u,C,C'}(S,T) = 1$, then $A_{u,C,C'}(S,T) = 1$, and if $A_{u,C,C'}(S,T) = 0$ then $B_{u,C,C'}(S,T) = 0$.

(2) The matrix $B_{u,C,C'}$ has exactly *$\frac{1}{2}D$ nonzero entries.*

(3) The matrix $\sum_{C\neq C'\in H_u} B_{u,C,C'}$ has maximum row/column ℓ_1-norm at most Δ.

Similarly to Definition 5.4.2, we let $B_u := \sum_{C\neq C'\in H_u} B_{u,C,C'} b_{u,C} b_{u,C'}$ and $B := \sum_{u\in[p]} B_u$.

We note that the above properties of the matrices $B_{u,C,C'}$ imply that Lemma 5.4.3 holds for the matrix B as well if we replace D with $\frac{1}{2}D$.

The reweighting matrix uses the following definition, which is a combinatorial notion that bounds the ℓ_1-norm of rows in B_u.

Definition 5.4.5 (Combinatorial proxy for the row ℓ_1-norm in B_u)**.** For $u \in [p]$ and $S \in \binom{2n}{\ell}$, we let $d_u(S) := \sum_T |B_{u,C,C'}(S,T)|$. We also define $d(S) := \sum_{u\in[p]} d_u(S)$.

Remark 5.4.6. We note that $d_u(S)$ is an upper bound on the ℓ_1-norm of the S-th row in B_u, with the difference being that B_u is a *random* matrix (with randomness coming from the $b_{u,C}$'s), and so the ℓ_1-norm of the S-th row may be lower depending on the draw of the $b_{u,C}$'s if H_u is a multigraph.

We also have that $\sum_S d(S) \leq \frac{2m^2D}{p}$. This is because this simply counts the total number of nonzero entries across all the $B_{u,C,C'}$'s, and there are exactly $D/2$ nonzero entries in each matrix, and there are p choices for u and $|H_u|^2 \leq 4m^2/p^2$ choices for $C \neq C' \in H_u$.

The reweighting and spectral norm bound steps are captured via the following lemma.

Lemma 5.4.7. *Let A be the Kikuchi matrix associated with the polynomial f obtained from an (ϵ, ℓ)-regular p-bipartite polynomial ψ of total degree k defined by $(k-1)$ uniform hypergraphs $\{H_u\}_{u\in[p]}$ and coefficients $\{b_{u,C}\}_{u\in[p],C\in H_u}$. Let B be the pruned Kikuchi matrix defined in Lemma 5.4.4, and let W be the diagonal PSD matrix where W_S is $d(S) + \frac{m^2D}{pN}$.*

Then, with probability $1 - 1/\mathrm{poly}(n)$ *over the draw of* $b_{u,C}$*'s, it holds that*

$$\|W^{-1/2}BW^{-1/2}\|_2 \leq O\left(\sqrt{\frac{pN\ell \log n}{m^2 D}} + \Delta \frac{pN\ell \log n}{m^2 D}\right).$$

We now finish the proof assuming Lemmas 5.4.4 and 5.4.7.

Finishing the proof of Theorem 5.3.4. We have already shown that

$$\mathrm{val}(f) \leq \frac{2p}{m^2 D}\|W^{-1/2}BW^{-1/2}\|_2 \cdot \mathrm{tr}(W),$$

where B is defined by Lemma 5.4.4.

Thus, our refutation algorithm operates as follows. First, we construct the matrices $A_{u,C,C'}$, and then we construct the matrices $B_{u,C,C'}$ (which exist and are well-defined, by Lemma 5.4.4). Then, we compute the matrix W and $\frac{2p}{m^2 D}\|W^{-1/2}BW^{-1/2}\|_2$, to obtain an upper bound on $\mathrm{val}(f)$. We note that because the spectral norm is a real number, we can only compute it to an additive error of $2^{-O(n)}$, Finally, we use Lemma 5.4.1 to compute an upper bound on $\mathrm{val}(\psi)$.

It thus remains to argue that with probability $1 - 1/\mathrm{poly}(n)$, the output of the idealized algorithm (namely, ignoring the 2^{-n} error from real number computation) is $O(\varepsilon)$, i.e., it produces an upper bound of $O(\varepsilon)$ on $\mathrm{val}(\psi)$.

By Lemma 5.4.7, we have that with probability $1 - 1/\mathrm{poly}(n)$ over the draw of $b_{u,C}$'s, it holds that

$$\|W^{-1/2}BW^{-1/2}\|_2 \leq O\left(\sqrt{\frac{pN\ell \log n}{m^2 D}} + \Delta \frac{pN\ell \log n}{m^2 D}\right).$$

Therefore, our algorithm certifies that

$$\mathrm{val}(f) \leq \frac{2p}{m^2 D}\|W^{-1/2}BW^{-1/2}\|_2 \cdot \mathrm{tr}(W).$$

We have that $\mathrm{tr}(W) \leq O\left(\frac{m^2 D}{p}\right)$, as $\sum_S d(S) \leq \frac{2m^2 D}{p}$, because for each $u \in [p]$ and pair $C \neq C' \in H_u$ (of which there are at most $4m^2/p$), the pair (C, C') contributes exactly $D/2$ entries, each of magnitude at most 1.

It thus follows that we certify that

$$\mathrm{val}(f) \leq O\left(\sqrt{\frac{pN\ell \log n}{m^2 D}} + \Delta \frac{pN\ell \log n}{m^2 D}\right)$$

By Fact 3.6.2, we have that $N/D \leq 2^{O(k)} \cdot \left(\frac{n}{\ell}\right)^{k-1}$. Recall that $\Delta = c^k \varepsilon^{-4}$, for some absolute constant c. Because we have $m \geq \Gamma^k \cdot \left(\frac{n}{\ell}\right)^{\frac{k-1}{2}} \sqrt{p\ell \log n} \cdot \varepsilon^{-3}$ for a sufficiently large constant Γ, it follows that we have $\mathrm{val}(f) \leq O(\varepsilon^2)$.

Finally, we have Lemma 5.4.1 that $\mathrm{val}(\psi)^2 \leq \frac{p}{m} + \mathrm{val}(f)$. As $m \geq p/\varepsilon^2$, it follows that we certify that $\mathrm{val}(\psi)^2 \leq O(\varepsilon^2)$, i.e., $\mathrm{val}(\psi) \leq O(\varepsilon)$. This finishes the proof. □

5.4.3 Row pruning

In order to implement our row pruning step and prove Lemma 5.4.4, we will define *bad* rows/columns of A_u for each u. The following key definition abstracts out the property (of the hypergraphs defining the input polynomial) that decides which rows are bad:

Definition 5.4.8 (Butterfly Degree)**.** Let H_u be a $(k-1)$-uniform hypergraph on $[n]$. For any $C, C' \in H_u$, let

$$\mathcal{R}_{(C,C')} = \left\{ R \subseteq [n] \times [2] \;\middle|\; |R| = k-1,\ \left\{ |R \cap C^{(1)}|, |R \cap C'^{(2)}| \right\} = \left\{ \left\lceil \frac{k-1}{2} \right\rceil, \left\lfloor \frac{k-1}{2} \right\rfloor \right\} \right\}.$$

For any $S \subseteq [n] \times [2]$, and $(k-1)$-uniform hypergraph H_u on $[n]$, the *butterfly degree* of S in H_u is defined by:

$$\gamma_u(S) = \sum_{(C,C') \in H_u \times H_u, C \neq C'} \sum_{R \in \mathcal{R}_{(C,C')}} \mathbf{1}(S \cap (C^{(1)} \cup C'^{(2)}) = R).$$

For a collection of $(k-1)$-uniform hypergraphs H_u on $[n]$ for $u \in [p]$, the *total butterfly degree* of S is defined by $\gamma(S) = \sum_{u \in [p]} \gamma_u(S)$.

We note that the notion of total butterfly degree above generalizes the notion of butterfly degree studied in [AGK21]; the original notion of "butterfly degree" is so named because it counts numbers of butterfly-shaped graphs.

The following lemma shows that the butterfly degree characterizes the maximum ℓ_1-norm of the rows of the Kikuchi matrix A_u.

Lemma 5.4.9 (Butterfly Degree and the ℓ_1-norm of rows of the Kikuchi Matrix)**.** *Let H_u be a $(k-1)$-uniform hypergraph on $[n]$ and A_u be the associated matrix in Definition 5.4.2. Then, for any $S \subseteq [n] \times [2]$, we have:*

$$\gamma_u(S) \geq \sum_{T} \sum_{C \neq C' \in H_u} |A_{u,C,C'}(S,T)|.$$

Proof. If k is odd, we observe that $\gamma_u(S)$ is the number pairs $(C, C') \in H_u \times H_u$ with $C \neq C'$ such that $|S \cap C^{(1)}| = |S \cap C'^{(2)}| = \frac{k-1}{2}$, and if k is even, $\gamma_u(S)$ is the number of pairs such that $|S \cap C^{(1)}| = \frac{k}{2}$ and $|S \cap C'^{(2)}| = \frac{k-2}{2}$ or $|S \cap C^{(1)}| = \frac{k-2}{2}$ and $|S \cap C'^{(2)}| = \frac{k}{2}$. The lemma now follows. □

We now identify "bad rows" in A_u as those that have too large total butterfly degrees.

Definition 5.4.10 (Δ-Bad rows in A_u)**.** We define the set of Δ-bad rows in A to be:

$$\mathcal{B}_u := \{S : \gamma_u(S) > \Delta\}.$$

Note that the set $\mathcal{B}_u$ *does not* depend on the values of the $b_{u,C}$'s.

Observe that by Lemma 5.4.9, every row that is not bad has an ℓ_1-norm that is bounded by Δ. The following lemma bounds the expectation of $\gamma_u(S)$ over the rows S where S is a nonzero row in $A_{u,C,C'}$. We defer the proof of Lemma 5.4.11 to the end of this subsection.

Lemma 5.4.11 (Conditional first moment of $\gamma_u(S)$)**.** *Let A be the Kikuchi matrix associated with the polynomial f obtained from an (ϵ, ℓ)-regular p-bipartite polynomial ψ of total degree k defined by $(k-1)$ uniform hypergraphs $\{H_u\}_{u \in [p]}$. Let $u \in [p]$. For $C \neq C' \in H_u$, let $\mathcal{U}_{u,C,C'}$ denote the uniform distribution over* nonzero *rows in $A_{u,C,C'}$. Then, $\mathbb{E}_{S \leftarrow \mathcal{U}_{u,C,C'}}[\gamma_u(S)] \leq 2^{O(k)} \varepsilon^{-4}$.*

By Markov's inequality, this immediately implies the following corollary, which bounds the number of entries that are deleted for a particular pair (C, C') by the row deletion process.

Corollary 5.4.12 (Row pruned Kikuchi matrices)**.** *Let A be the Kikuchi matrix associated with the polynomial f obtained from an (ϵ, ℓ)-regular p-bipartite polynomial ψ of total degree k defined by $(k-1)$ uniform hypergraphs $\{H_u\}_{u\in[p]}$. Let $u \in [p]$. Let $\mathcal{B}_u$ is the set of Δ-bad rows in A_u for*

$$\Delta = c^k \frac{1}{\varepsilon^4}\,, \tag{5.10}$$

where c is an absolute constant. Then, for each $C \neq C' \in H_u$, the number pairs (S, T) with $S, T \notin \mathcal{B}_u$ such that $A_{u,C,C'}(S, T) = 1$ is at least $\frac{1}{2}D$.

In particular, for each pair $C \neq C' \in H_u$, there exists a symmetric matrix $B_{u,C,C'} \in \{0,1\}^{N\times N}$ such that

(1) The matrix $B_{u,C,C'}$ is a "subset" of the matrix $A_{u,C,C'}$. Namely, for any pair (S, T), if $B_{u,C,C'}(S, T) = 1$, then $A_{u,C,C'}(S, T) = 1$, and if $A_{u,C,C'}(S, T) = 0$ then $B_{u,C,C'}(S, T) = 0$.

(2) The matrix $B_{u,C,C'}$ has exactly *$\frac{1}{2}D$ nonzero entries.*

(3) For every S, $\sum_T \sum_{C\neq C'\in H_u} B_{u,C,C'}(S, T) \leq \Delta$.

Proof of Corollary 5.4.12 from Lemma 5.4.11. Fix $C \neq C' \in H_u$. We observe that, because c is a large enough absolute constant, by applying Markov's inequality and using Lemma 5.4.11, the probability that a row $S \leftarrow \mathcal{D}_{u,C,C'}$ has $\gamma_u(S) \geq \Delta$ is at most 0.01. Let $A'_{u,C,C'}$ be the matrix obtained by (1) starting with the matrix $A_{u,C,C'}$, and (2) "zeroing out" all rows/columns in $\mathcal{B}_u$, i.e., setting $A'_{u,C,C'}(S, T) = 0$ if $S \in \mathcal{B}_u$ or $T \in \mathcal{B}_u$.

By the above, this can remove at most $2 \cdot 0.01 \cdot D$ nonzero entries from $A_{u,C,C'}$, so $A'_{u,C,C'}$ has at least $0.98 \cdot D$ nonzero entries. We then let $B_{u,C,C'}$ be an arbitrary matrix obtained by taking a subset of exactly $\frac{1}{2}D$ of the nonzero entries of $A'_{u,C,C'}$. Because $A_{u,C,C'}$ is symmetric, the set of bad rows and bad columns is the same, and so $A'_{u,C,C'}$ is symmetric. Thus, we can also make $B_{u,C,C'}$ be symmetric as well.

We have clearly found symmetric matrices $B_{u,C,C'}$ that satisfy the first two properties. To show the last property, we observe that for any row S, we have either $S \in \mathcal{B}_u$, in which case the S-th row of $B_{u,C,C'}$ is zero, or else $S \notin \mathcal{B}_u$, in which case we have

$$\sum_T \sum_{C\neq C'\in H_u} B_{u,C,C'}(S, T) \leq \sum_T \sum_{C\neq C'\in H_u} A_{u,C,C'}(S, T) \leq \gamma_u(S) \leq \Delta\,,$$

where we use Lemma 5.4.9 and the observation that if $B_{u,C,C'}$ has a nonzero entry, then so does $A_{u,C,C'}$. □

It remains to prove Lemma 5.4.11, which we do now.

Proof of Lemma 5.4.11. Let $C \neq C' \in H_u$, and let $\mathcal{U}_{u,C,C'}$ be the uniform distribution over the rows in $A_{u,C,C'}$ that contain a nonzero entry. Note that by definition, if a row S has a nonzero entry, then it has exactly one nonzero entry, and there are exactly D nonzero entries in $A_{u,C,C'}$, so there

are exactly D rows with nonzero entries. If k is even, we have

$$\mathbb{E}_{S\leftarrow\mathcal{U}_{u,C,C'}}[\gamma_u(S)] = \frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{C''\neq C'''\in H_u}|\{S : S\in\mathcal{J}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|$$
$$+\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lceil\frac{k-1}{2}\rceil}\sum_{J_2\subseteq C:|J_2|=\lfloor\frac{k-1}{2}\rfloor}\sum_{C''\neq C'''\in H_u}|\{S : S\in\mathcal{J}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|,$$

and if k is odd, we have

$$\mathbb{E}_{S\leftarrow\mathcal{U}_{u,C,C'}}[\gamma_u(S)] = \frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\frac{k-1}{2}}\sum_{J_2\subseteq C:|J_2|=\frac{k-1}{2}}\sum_{C''\neq C'''\in H_u}|\{S : S\in\mathcal{J}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|.$$

Below, we will bound

$$\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{C''\neq C'''\in H_u}|\{S : S\in\mathcal{J}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|,$$

when k is either even or odd. It will be clear from the calculation that, by symmetry, the bound we show will also apply to the term when k is even.

Consider a fixed choice of $J_1\subseteq C$, $J_2\subseteq C'$ with $|J_1| = \lfloor\frac{k-1}{2}\rfloor$ and $|J_2| = \lceil\frac{k-1}{2}\rceil$. Let us fix $r_1\leq\lfloor\frac{k-1}{2}\rfloor$ and $r_2\leq\lceil\frac{k-1}{2}\rceil$, and let $R_1\subseteq J_1$, $R_2\subseteq J_2$ with $|R_1| = r_1\leq\lfloor\frac{k-1}{2}\rfloor$ and $|R_2| = r_2\leq\lceil\frac{k-1}{2}\rceil$. Let us also consider a fixed choice of $C'', C'''\in H_u$ with $C''\neq C'''$, $C''\cap J_1 = R_1$, and $C'''\cap J_2 = R_2$. We will bound $|\{S : S\in\mathcal{R}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|$.

Observe that $|S| = \ell$ and $J_1^{(1)}\cup J_2^{(2)}\subseteq S$. Thus, to count the number of S, we simply need to count the choices for the remaining $\ell-(k-1)$ elements of S. Because $S\in\mathcal{R}_{(C'',C''')}$, it must contain at least $\lfloor\frac{k-1}{2}\rfloor$ elements of C'' and $\lceil\frac{k-1}{2}\rceil$ elements of C'''. Because $|C''\cap J_1| = r_1$ and $|C'''\cap J_2| = r_2$, this means that the $\ell-(k-1)$ elements of $S\setminus(R_1^{(1)}\cup R_2^{(2)})$ must contain at least $\lfloor\frac{k-1}{2}\rfloor - r_1$ elements of $(C''\setminus R_1)^{(1)}$ and $\lceil\frac{k-1}{2}\rceil - r_2$ elements of $(C'''\setminus R_2)^{(2)}$. Thus, the number of choices for S is

$$|\{S : S\in\mathcal{R}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|\leq\binom{k-1}{\lfloor\frac{k-1}{2}\rfloor-r_1}\cdot\binom{k-1}{\lceil\frac{k-1}{2}\rceil-r_2}\binom{2n}{\ell-2(k-1)+r_1+r_2},$$

where we note that $\ell-(k-1)-(\lfloor\frac{k-1}{2}\rfloor-r_1)-(\lceil\frac{k-1}{2}\rceil-r_2) = \ell-2(k-1)$.

We thus have that

$$\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{C''\neq C'''\in H_u}|\{S : S\in\mathcal{J}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|$$
$$=\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\sum_{C''\neq C'''\in H_u:C''\cap J_1=R_1,C'''\cap J_2=R_2}|\{S : S\in\mathcal{R}_{(C'',C''')}, J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|$$
$$\leq\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\sum_{C''\neq C'''\in H_u:C''\cap J_1=R_1,C'''\cap J_2=R_2}2^{O(k)}\binom{2n}{\ell-2(k-1)+|R_1|+|R_2|}.$$

Now, applying Facts 3.6.1 and 3.6.2, we have that

$$\frac{\binom{2n}{\ell-2(k-1)+r_1+r_2}}{D}\leq 2^{O(k)}\left(\frac{\ell}{n}\right)^{2(k-1)-r_1-r_2-(k-1)} = 2^{O(k)}\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}.$$

Thus, we have the bound

$$\begin{aligned}
&\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{C''\neq C'''\in H_u}|\{S:S\in\mathcal{J}_{(C'',C''')},J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|\\
&\leq\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\sum_{C''\neq C'''\in H_u:C''\cap J_1=R_1,C'''\cap J_2=R_2}2^{O(k)}\left(\frac{\ell}{n}\right)^{(k-1)-r_1-r_2}\\
&\leq\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\deg_u(R_1)\deg_u(R_2)2^{O(k)}\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}.
\end{aligned}$$

Because the H_u's are (ε,ℓ)-regular, we have that for any $b\in\{1,2\}$, $\deg_u(R_b)\leq\frac{1}{\varepsilon^2}\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1-|R_b|}$ if $|R_b|\leq\frac{k-2}{2}$, and $\deg_u(R_b)\leq\frac{1}{\varepsilon^2}$ if $|R_b|=\frac{k-1}{2}$ (if k odd) or $\frac{k}{2}$ (if k even). We have a few cases. If $|R_b|\leq\frac{k}{2}-1$, then

$$\deg_u(R_1)\deg_u(R_2)\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\leq\frac{1}{\varepsilon^4}\cdot\frac{\ell}{n}.$$

If k is odd and $|R_b|=\frac{k-1}{2}$ for one choice of b and $|R_b|\leq\frac{k}{2}-1$ for the other choice of b, then we have

$$\deg_u(R_1)\deg_u(R_2)\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\leq\frac{1}{\varepsilon^4}\cdot\sqrt{\frac{\ell}{n}}.$$

If k is odd and $|R_1|=|R_2|=\frac{k-1}{2}$, then we have

$$\deg_u(R_1)\deg_u(R_2)\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\leq\frac{1}{\varepsilon^4}.$$

Finally, if k is even and $|R_b|=\frac{k}{2}$ for one choice of b, then we must have $|R_b|\leq\frac{k}{2}-1$ for the other choice of b, and we thus have

$$\deg_u(R_1)\deg_u(R_2)\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\leq\frac{1}{\varepsilon^4}.$$

In all cases, we conclude that $\deg_u(R_1)\deg_u(R_2)\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\leq\frac{1}{\varepsilon^4}$, and so we have a bound of

$$\begin{aligned}
&\frac{1}{D}\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{C''\neq C'''\in H_u}|\{S:S\in\mathcal{J}_{(C'',C''')},J_1^{(1)}\cup J_2^{(2)}\subseteq S\}|\\
&\leq\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\deg_u(R_1)\deg_u(R_2)2^{O(k)}\left(\frac{\ell}{n}\right)^{(k-1)-|R_1|-|R_2|}\\
&\leq\sum_{J_1\subseteq C:|J_1|=\lfloor\frac{k-1}{2}\rfloor}\sum_{J_2\subseteq C:|J_2|=\lceil\frac{k-1}{2}\rceil}\sum_{R_1\subseteq J_1,R_2\subseteq J_2}\frac{2^{O(k)}}{\varepsilon^4}\\
&\leq\frac{2^{O(k)}}{\varepsilon^4}.
\end{aligned}$$

□

5.4.4 Bounding the spectral norm of the "reweighted pruned matrix": proof of Lemma 5.4.7

We now prove Lemma 5.4.7. The proof is based on the trace moment method, and will also be important to us in Chapter 9 in Part II of this thesis.

Proof. We observe that $\|W^{-1/2}BW^{-1/2}\|_2^{2r} \le \mathrm{tr}((W^{-1/2}BW^{-1/2})^{2r}) = \mathrm{tr}((W^{-1}B)^{2r})$. We will proceed with the proof in two steps. First, we upper bound $\mathbb{E}[\mathrm{tr}((W^{-1}B)^{2r})]$ by a combinatorial quantity: the total weight of "even walk sequences", which we define below. Then, we bound the total weight of such sequences.

Definition 5.4.13. Let $S \in \binom{2n}{\ell}$. We say that a sequence $(u_1, C_1, C_1'), \dots, (u_{2r}, C_{2r}, C_{2r}')$ with $u_h \in [p]$ and $C_h \neq C_h' \in H_{u_h}$ is a "walk sequence" for S if the sets $S_h := S \oplus \bigoplus_{j \le h}(C_j^{(1)} \oplus C_j'^{(2)})$ each have size *exactly* ℓ and the entries $B_{u_h}(S_h, S_{h+1})$ are nonzero for each $h \in \{0, \dots, 2r\}$, where $S_0 := S$. Moreover, the sequence is *even* if each (u, Q) appears an even number of times in the multiset $\{(u_h, C_h), (u_h, C_h')\}_{h \in [2r]}$.

The weight of the sequence is $\prod_{h=0}^{2r-1} \frac{1}{W_{S_h}}$.

Proposition 5.4.14. *We have*

$$\mathbb{E}[\mathrm{tr}((W^{-1}B)^{2r})] \le \sum_{S \in \binom{2n}{\ell}} \sum_{\substack{\text{even walk sequences} \\ (u_1, C_1, C_1'), \dots, (u_{2r}, C_{2r}, C_{2r}') \text{ for } S}} \mathrm{wt}(S, (u_1, C_1, C_1'), \dots, (u_{2r}, C_{2r}, C_{2r}')) .$$

Lemma 5.4.15 (Sequence counting)**.** *For each S, it holds that*

$$\sum_{\substack{\text{even walk sequences} \\ (u_1, C_1, C_1'), \dots, (u_{2r}, C_{2r}, C_{2r}') \text{ for } S}} \mathrm{wt}(S, (u_1, C_1, C_1'), \dots, (u_{2r}, C_{2r}, C_{2r}')) \le (4r)^r \left(\frac{pN}{m^2 D}\right)^{2r} \left(\frac{2m^2 D}{pN} + r\Delta^2\right)^r .$$

We observe that Proposition 5.4.14 and Lemma 5.4.15 immediately imply Lemma 5.4.7. Indeed, we have that

$$\mathbb{E}[\mathrm{tr}((W^{-1}B)^{2r})] \le N(4r)^r \left(\frac{pN}{m^2 D}\right)^{2r} \left(\frac{2m^2 D}{pN} + r\Delta^2\right)^r ,$$

and hence by Markov's inequality,

$$\Pr[\|W^{-1/2}BW^{-1/2}\|_2 \ge \lambda] \le \frac{\mathbb{E}[\|W^{-1/2}BW^{-1/2}\|_2^{2r}]}{\lambda^{2r}} \le \frac{N(4r)^r \left(\frac{pN}{m^2 D}\right)^{2r} \left(\frac{2m^2 D}{pN} + r\Delta^2\right)^r}{\lambda^{2r}} .$$

Taking $r = \lceil \log_2 N \rceil$ and $\lambda = c\left(\sqrt{\frac{pNr}{m^2 D}} + r\Delta\frac{pN}{m^2 D}\right)$ for a large enough absolute constant c thus implies

$$\Pr\left[\|W^{-1/2}BW^{-1/2}\|_2 \ge c\left(\sqrt{\frac{pNr}{m^2 D}} + r\Delta\frac{pN}{m^2 D}\right)\right] \le \frac{N4^r}{c^{2r}} \le \frac{1}{\mathrm{poly}(N)} ,$$

which finishes the proof of Lemma 5.4.7, as $r \le O(\log N) = O(\ell \log n)$. □

We now prove Proposition 5.4.14 and Lemma 5.4.15.

Proof of Proposition 5.4.14. We compute:

$$\mathbb{E}[\mathrm{tr}((W^{-1}B)^{2r})] = \sum_{(u_1,S_1),\ldots,(u_{2r},S_{2r})} \mathbb{E}[\prod_{h=1}^{2r} \frac{1}{W_{S_{h-1}}} B_{u_h}(S_{h-1},S_h)]\,,$$

where we use the convention that $u_{2r+1} := u_1$ and $S_0 := S_{2r}$. Next, we observe that this is equal to

$$\begin{aligned}
&= \sum_S \sum_{(u_1,C_1,C_1'),\ldots,(u_{2r},C_{2r},C_{2r}') \text{ walk sequence for } S} \mathbb{E}[\prod_{h=1}^{2r} \frac{1}{W_{S_{h-1}}} B_{u_{h-1}}(S_{h-1},S_h)] \\
&= \sum_S \sum_{(u_1,C_1,C_1'),\ldots,(u_{2r},C_{2r},C_{2r}') \text{ walk sequence for } S} \mathbb{E}[\prod_{h=1}^{2r} \frac{1}{W_{S_{h-1}}} b_{u_{h-1},C_{h-1}} b_{u_{h-1},C_{h-1}'}] \\
&\le \sum_S \sum_{\substack{\text{even walk sequences} \\ (u_1,C_1,C_1'),\ldots,(u_{2r},C_{2r},C_{2r}') \text{ for } S}} \mathrm{wt}(S,(u_1,C_1,C_1'),\ldots,(u_{2r},C_{2r},C_{2r}'))\,,
\end{aligned}$$

as the term in the sum is 0 unless the walk sequence is even. □

Proof of Lemma 5.4.15. We shall upper bound the total weight of such sequences for each S via an encoding argument. For a set S and $u \in [p]$, we will say that $C, C' \in H_u$ *extends* S if $B_u(S, S \oplus C^{(1)} \oplus C'^{(2)})$ is well-defined and nonzero. The encoding is as follows:

(1) Choose $z \in [r]$, the number of *distinct* u's that appear in the sequence. Note that z must be at most r because the sequence is even; u_h cannot appear once in $\{u_1,\ldots,u_{2r}\}$, as then we must pair (u_h, C_h) with (u_h, C_h'), but we must have $C_h \neq C_h'$.
(2) Choose $2z$ locations L in $[2r]$. These will denote the first and last occurrence of each distinct u_h for $h \in [z]$.
(3) Choose a perfect matching π for the $2z$ chosen locations. We will think of π as a function $\pi : L \to [z]$, satisfying $t_1 < t_2 < \cdots < t_z$, where t_h is the first preimage of h in L (using the natural ordering on L inherited from $[2r]$). We let t_h' denote the second preimage of h in L.
(4) Proceed in order of steps $t = 1,\ldots,2r$. We thus know the set S_t that we are currently "at". There are three cases.
   (a) Suppose $t = t_h$ for some h. Then, (1) choose $u \in [p]$ (that has not yet been chosen); (2) choose $C, C' \in H_u$ extending S_t; (3) set the t-th element of the sequence to be (u, C, C').
   (b) Suppose that $t \neq t_h, t_h'$ for all $h \in [z]$. Then, pick a previously chosen u (that has not yet reached its last occurrence according to the matching π), and pick $C, C' \in H_u$ that extends S_t. Set the t-th element of the sequence to be (u, C, C').
   (c) Suppose that $t = t_h'$ for some h. Then, choose $u = u_h$ and let $C, C' \in H_u$ be the *unique* pair that extends S_t and keeps the sequence even. Set the t-th element of the sequence to be either (u, C, C') or (u, C', C).

We now count the number of choices. Let us first think of the first 3 steps as fixed. There are 3 cases. If we are choosing a new u, then there are $\sum_u d_u(S_t) = d(S_t)$ ways to pick (u, C, C'), and this is multiplied by a weight of $\frac{1}{W_{S_t}} \le \frac{1}{d(S_t)}$, so this adds a total weight of at most 1.

If we are choosing an old u, then there are $z\Delta$ ways to pick (u, C, C'), as we have z choices for u and then $d_u(S_t) \le \gamma_u(S_t) \le \Delta$ choices for the pair C, C'. This is multiplied by a weight of $\frac{1}{W_{S_t}} \ge \frac{pN}{m^2D}$, for a total contribution of $\frac{z\Delta pN}{m^2D}$.

Finally, if we are at $t = t'_h$ for some h, then we have 2 choices, and thus the total contribution to the weight is at most $\frac{2pN}{m^2D}$. Hence, across all steps, we have $1^z \cdot \left(\frac{2pN}{m^2D}\right)^z \cdot \left(\frac{z\Delta pN}{m^2D}\right)^{2r-2z}$ choices.

Next, we think of z as fixed, and count the choices for Steps (2) and (3). These have $\binom{2r}{2z}$ choices and $\frac{(2z)!}{2^z z!}$ choices, respectively. Combining, we thus have the bound

$$\#\,(u_1, C_1, C'_1), \ldots, (u_{2r}, C_{2r}, C'_{2r}) \text{ even, well-formed for } S \le \sum_{z=1}^{r} \binom{2r}{2z} \frac{(2z)!}{2^z z!} \left(\frac{2pN}{m^2D}\right)^z \cdot \left(\frac{z\Delta pN}{m^2D}\right)^{2r-2z}.$$

We now observe that

$$\begin{aligned}
&\binom{2r}{2z} \frac{(2z)!}{z!} z^{2r-2z} = \frac{(2r)!}{(2r-2z)!z!} \cdot z^{2r-2z} \\
&= \frac{(2r)!}{r!r!} \cdot \frac{(r-z)!(r-z)!}{(2r-2z)!} \cdot \frac{r!}{(r-z)!} \cdot \frac{r!}{z!(r-z)!} \cdot z^{2r-2z} \\
&\le 2^{2r} \cdot 1 \cdot r^z \cdot \binom{r}{z} \cdot r^{2r-2z} \\
&\le (4r)^r \binom{r}{z} r^{r-z}.
\end{aligned}$$

Thus, the total weight is at most

$$\begin{aligned}
&\sum_{z=1}^{r} \binom{2r}{2z} \frac{(2z)!}{2^z z!} \left(\frac{2pN}{m^2D}\right)^z \cdot \left(\frac{z\Delta pN}{m^2D}\right)^{2r-2z} \le (4r)^r \sum_{z=1}^{r} \binom{r}{z} 2^z \left(\frac{pN}{m^2D}\right)^{2r-z} r^{r-z} \Delta^{2r-2z} \\
&= (4r)^r \left(\frac{pN}{m^2D}\right)^{2r} \sum_{z=1}^{r} \binom{r}{z} \left(\frac{2m^2D}{pN}\right)^z (r\Delta^2)^{r-z} \\
&= (4r)^r \left(\frac{pN}{m^2D}\right)^{2r} \left(\frac{2m^2D}{pN} + r\Delta^2\right)^r,
\end{aligned}$$

which finishes the proof. □

5.5 Strong CSP refutation: smoothed via semirandom

In this section, we show how the tight refutation of semirandom sparse polynomials in Section 5.3 can be used in a black-box way to derive nearly optimal algorithms for strongly refuting smoothed CSPs and, as a special case, semirandom CSPs.

Smoothed model. Let us first formally describe the model of smoothed Boolean CSPs.

Definition 5.5.1 (Smoothed CSP Instances [Fei07])**.** Let $k \in \mathbb{N}$. Let ψ be an instance of a CSP with predicate $P : \{-1, 1\}^k \to \{0, 1\}$ specified by a collection of k-tuples H and literal patterns ξ. Let $\vec{p} = \{p_{C,i}\}_{C \in H, i \in [k]}$ with each $p_{C,i} \in [0, 1]$ be smoothing parameters, one for every $C \in H$ and $i \in [k]$. A $\vec{p}$-smoothing of ψ is obtained as follows:

1. For every $C \in H$, let $S_C \subseteq [k]$ be obtained by adding i to S_C with probability $p_{C,i}$ independently for every $i \in C$.
2. For every $i \in S_C$, reset $\xi(C,i)$ to be a uniform and independent random bit in ± 1.

Remark 5.5.2.
1. The notion of smoothing allows using a different probability of "rerandomizing" each of mk literals in a k-CSP instance ψ with m constraints.
2. The two-step random process above is equivalent to flipping the negation pattern $\xi(C,i)$ of the i-th literal in clause $C \in H$ independently of others with probability $p_{C,i}/2$.
3. Setting $p_{C,i} = 1$ for every i, C yields the model where the literal patterns are uniformly random and independent in $\{\pm 1\}$. This is the semirandom model of CSPs.

We now proceed to state and prove our main results concerning refutation of smoothed instances, along the way noting also a better bound for the special semirandom case. We recall the notion of t-wise uniform distributions before presenting the main result.

Definition 5.5.3 (t-wise uniform distribution)**.** A probability distribution μ on $\{-1,1\}^k$ is said to be t-wise uniform if $\mathbb{E}_{z\sim\mu} \prod_{i\in S} z_i = 0$ for every $S \subseteq [k]$ of size $|S| \leq t$.

Theorem 5.5.4 (Smoothed Boolean CSP Refutation)**.** *Let $P : \{-1,1\}^k \to \{0,1\}$ be a k-ary Boolean predicate such that there is no t-wise uniform distribution supported on $P^{-1}(1)$. Let ℓ be an integer with $2(k-1) \leq \ell \leq n$. There is an algorithm that takes as input an instance Θ of CSP(P) and outputs a value* alg-val$(\Theta) \in [0,1]$ *in time $n^{O(\ell)}$ satisfying the following:*

(1) val$(\Theta) \leq$ alg-val$(\Theta) \leq 1$.

(2) Suppose the input instance Θ is a smoothing ψ_s of an arbitrary CSP instance $\psi = (H, \xi)$ with n variables and m constraints w.r.t. a vector of smoothing parameters $\vec{p} = \{p_{C,i}\}$ in $[0,1]$. Suppose that $m \geq \frac{2m_0}{q(\vec{p})}$, where

$$m_0 = \frac{2^{O(k)} \log_2 n}{\varepsilon^5} \cdot \ell \left(\frac{n}{\ell}\right)^{\frac{t}{2}}$$

and

$$q(\vec{p}) = \frac{1}{m} \sum_{C\in H} \prod_{i\in C} p_{C,i}\,. \tag{5.11}$$

Then with probability at least $1 - 1/\mathrm{poly}(n)$ over the randomness of the smoothening process, it holds that alg-val$(\Theta) \leq 1 - \frac{q(\vec{p})}{2} \cdot (\delta_t - \epsilon) + 2^{-n}$. *Here, $\delta_t \geq 2^{-\tilde{O}(k^t)}$ depends only on the predicate P. Furthermore, in the semirandom case (where all $p_{C,i} = 1$), we have* alg-val$(\Theta) \leq 1 - \delta_t + \epsilon + 2^{-n}$ *with probability $1 - 1/\mathrm{poly}(n)$.*

Moreover, the algorithm is captured by the canonical degree 2ℓ sum-of-squares relaxation of the CSP maximization problem over the hypercube.

The following result, proved in [AOW15] using LP duality, plays a crucial role in our proof of the above theorem, by allowing us to bound the value of CSP with predicate P that does not support a t-wise uniform distribution by a degree-t polynomial as proxy.

Fact 5.5.5 (Separating Polynomials, Lemma 3.16 and Theorem 4.10 in [AOW15])**.** *Let $P : \{-1,1\}^k \to \{0,1\}$ be a predicate such that there is no t-wise uniform distribution supported on $P^{-1}(1)$. Then, there is a $\delta_t \geq 2^{-\tilde{O}(k^t)}$ such that for every t-wise uniform distribution ζ, $\mathbb{E}_\zeta[P] \leq 1 - \delta_t$. Furthermore, there is a degree-t polynomial $Q : \{-1,1\}^k \to \mathbb{R}$ such that $Q(z) = \sum_{T\subseteq[k]} \hat{Q}(T) z_T$ and:*

1. *$P(z) \leq 1 - \delta_t + Q(z)$ for every $z \in \{-1,1\}^k$*
2. *$\hat{Q}(\emptyset) = 0$, i.e. Q has no constant coefficient, and,*

3. $\sum_{T\subseteq[k]} |\hat{Q}(T)| \le 2^{2k}$.

We now turn to the task of proving Theorem 5.5.4.

5.5.1 Proof of Theorem 5.5.4

By Fact 3.5.2, there is an algorithm that in $n^{O(\ell)}$-time outputs a value alg-val$(\Theta) \in [0,1]$ such that $\beta \le$ alg-val$(\Theta) \le \beta + 2^{-n}$, where $\beta = \max \tilde{\mathbb{E}}[\Theta]$, $\Theta(x) := \sum_{C\in H} P(\xi(C,1)x_{C_1}, \ldots, \xi(C,k)x_{C_k})$ is a degree $\le 2k$ polynomial, and the maximum is taken over degree-2ℓ pseudo-expectations $\tilde{\mathbb{E}}$ over $\{-1,1\}^n$. Note that Θ is indeed a degree $\le 2k$ polynomial, as P can always be expressed as a degree $\le 2k$ polynomial.

First, we observe that Item (1), i.e., completeness, is completely trivial: simply take $\tilde{\mathbb{E}}$ to be the expectation $\mathbb{E}_\mu$ of a distribution μ supported only on optimal solutions to Θ. Indeed, this implies that val$(\Theta) \le \beta \le$ alg-val(Θ). We thus focus on proving Item (2).

We will analyze the smoothing random process using the two steps that define it. Let us first consider the event that the first step chooses to re-randomize *all* the literals in a given clause $C \in H$; the probability of this event is $\prod_{i=1}^{k} p_{C,i}$. Let $\mathcal{G}$ be the set of clauses for which this occurs. Observe that the 0-1 indicator of "all literals are chosen to be re-randomized in C" is independent across clauses $C \in H$. The expected number of clauses in $\mathcal{G}$ equals $mq(\vec{p}) = \sum_{C\in H} \prod_{i=1}^{k} p_{C,i}$. Thus, by Chernoff bound, $|\mathcal{G}| \ge 0.5mq(\vec{p})$ with probability at least $1 - e^{-mq(\vec{p})/8} \ge 1 - e^{-m_0/4} \ge 1 - 1/\text{poly}(n)$, as $mq(\vec{p}) \ge 2m_0$. Let us proceed assuming that $|\mathcal{G}| \ge 0.5mq(\vec{p})$.

Let ξ denote the literal patterns after re-randomizing. We see that for every $C \in \mathcal{G}$ and $i \in [k]$, $\xi(C,i)$ is drawn uniformly and independently from $\{-1,1\}$. We shall view $\xi(C,i)$ as fixed for all $C \notin \mathcal{G}, i \in [k]$, and think of the $\xi(C,i)$'s for $C \in \mathcal{G}, i \in [k]$ as being random. For $C \in \mathcal{G}$, let $r_{C,i}$ denote the random variable $\xi(C,i)$, which is uniformly random in $\{-1,1\}$.

Let

$$\psi_g = \frac{1}{|\mathcal{G}|} \sum_{C\in\mathcal{G}} P(r_{C_1}x_{C_1}, \ldots, r_{C_k}x_{C_k}),$$

$$\psi_b = \frac{1}{|H| - |\mathcal{G}|} \sum_{C\notin\mathcal{G}} P(\xi(C,1)x_{C_1}, \ldots, \xi(C,k)x_{C_k}),$$

so that $|H|\psi_s = |\mathcal{G}|\psi_g + (|H| - |\mathcal{G}|)\psi_b$. Thus, by linearity of pseudo-expectations, we must have that for any pseudo-expectation $\tilde{\mathbb{E}}$,

$$\tilde{\mathbb{E}}[\psi_s] \le \frac{|\mathcal{G}|}{|H|}|\tilde{\mathbb{E}}[\psi_g]| + (1 - \frac{|\mathcal{G}|}{|H|})|\tilde{\mathbb{E}}[\psi_b]|. \tag{5.12}$$

Note that ψ_g and ψ_b are not known to our algorithm; these quantities appear only in our analysis.

Now, we know that for every x, $P(\xi(C,1)x_{C_1}, \ldots, \xi(C,k)x_{C_k}) \le 1$. As P is a degree k polynomial on k variables, by Fact 3.5.7, for every pseudo-expectation $\tilde{\mathbb{E}}$ of degree $2\ell \ge 2k$, $\tilde{\mathbb{E}}[P(\xi(C,1)x_{C_1}, \ldots, \xi(C,k)x_{C_k})] \le 1$. Using linearity of $\tilde{\mathbb{E}}$ and adding up the inequalities above for $C \notin \mathcal{G}$ yields that:

$$\tilde{\mathbb{E}}[\psi_b] \le 1. \tag{5.13}$$

Let us now analyze $\tilde{\mathbb{E}}[\psi_g]$. First, we invoke Fact 5.5.5 to conclude that for every x, it holds that:

$$P(r_{C,1}x_{C_1}, \ldots, r_{C,k}x_{C_k}) \le 1 - \delta_t + Q(r_{C,1}x_{C_1}, \ldots, r_{C,k}x_{C_k}).$$

As $\deg(Q) = t \le k$, by Fact 3.5.7 and summing up over $C \in \mathcal{G}$, for every pseudo-expectation of degree $2\ell \ge 2k$, we must have that:

$$\tilde{\mathbb{E}}[\psi_g] \le 1 - \delta_t + \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} \tilde{\mathbb{E}}[Q(r_{C,1}x_{C_1}, \dots, r_{C,k}x_{C_k})] \,.$$

Next, let $T \subseteq [k]$ of size $\le t$. For each C, let $x_{C|_T} = \prod_{i \in T} x_{C_i}$ and $b_{C|_T} = \Pi_{i \in T} r_{C,i}$. Observe that $Q(z) = \sum_{0<|T|\le t} \hat{Q}(T) z_T$ from Fact 5.5.5 and that further, $\sum_{0<|T|\le t} |\hat{Q}(T)| \le 2^{2k}$. Thus, we have:

$$\tilde{\mathbb{E}}[\psi_g] \le 1 - \delta_t + \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} \sum_{T \subseteq [k], 0<|T|\le t} |\hat{Q}(T)| b_{C|_T} \tilde{\mathbb{E}}\left[x_{C|_T}\right] \,.$$

Define ϕ_T to be the homogenous degree $|T|$ polynomial described by:

$$\phi_T(x) = \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} b_{C|_T} x_{C|_T}$$

Then, notice that:

$$\tilde{\mathbb{E}}[\psi_g] \le 1 - \delta_t + \sum_{T \subseteq [k], 0<|T|\le t} |\hat{Q}(T)| \tilde{\mathbb{E}}[\phi_T] \,. \tag{5.14}$$

We now observe that each ϕ_T is a polynomial with independent random coefficients in $\{-1, 1\}$. Further, since $|\mathcal{G}| \ge 0.5 q(\vec{p}) m \ge m_0$, by Theorem 5.3.1, with probability at least $1 - 1/\mathrm{poly}(n)$, we must have that for every pseudo-expectation $\tilde{\mathbb{E}}$ of degree at least 2ℓ,

$$\tilde{\mathbb{E}}[\phi_T] \le \frac{\epsilon}{2^{2k}} \,.$$

By a union bound over $\le 2^k$ possible T, this bound holds for every T with probability at least $1 - 1/\mathrm{poly}(n)$. Conditioning on this event, combining with (5.14), and using that $\sum_T |\hat{Q}(T)| \le 2^{2k}$ gives:

$$\tilde{\mathbb{E}}[\psi_g] \le 1 - \delta_t + \epsilon \,. \tag{5.15}$$

Thus, plugging this bound into (5.12) and using (5.13) yields:

$$\tilde{\mathbb{E}}[\psi_s] \le \left(1 - \frac{|\mathcal{G}|}{|H|}\right) \cdot 1 + \frac{|\mathcal{G}|}{|H|} \cdot (1 - \delta_t + \epsilon) \le 1 - \frac{|\mathcal{G}|}{|H|}(\delta_t - \epsilon) \le 1 - (\delta_t - \epsilon) \cdot \frac{q(\vec{p})}{2} \,, \tag{5.16}$$

where we use that $\frac{|\mathcal{G}|}{|H|} \ge q(\vec{p})/2$. Note that here we require $\delta_t \ge \varepsilon$, although the conclusion is trivial if this does not hold. As $\text{alg-val}(\psi_s) \le \beta + 2^{-n} \le 1 - (\delta_t - \epsilon) \cdot \frac{q(\vec{p})}{2} + 2^{-n}$, this completes the proof for the smoothed case.

As the semirandom model is the special case of the smoothed model (where $p_{C,i} = 1$ for every i), the above argument directly yields an upper bound of $\tilde{\mathbb{E}}[\psi] \le 1 - 0.5(\delta_t - \epsilon) + 2^{-n}$ for the case of semirandom instances. However, we incurred the 0.5 factor entirely due to the probabilistic bound on $|\mathcal{G}|$, and in the semirandom setting, $|\mathcal{G}| = |H|$ with probability 1. Hence, for semirandom refutation, we do not lose this extra 0.5 factor.

5.6 Analyzing the [WAM19] approach for random 3-XOR

In this section, we will prove that the approach suggested by [WAM19] (in their Appendix F.1, F.2) for strongly refuting random k-XOR with k odd does not yield the right trade-off for m as a function of n, ℓ. Our proof reduces to showing that a certain matrix defined in [WAM19] does not have small spectral norm. For simplicity, we present the argument for $k = 3$.

First, we give a brief overview of their approach. Let ϕ be a random 3-XOR instance in n variables and m clauses, with hypergraph H and coefficients $\{b_C\}_{C \in H}$. We will assume that each pair $C_1 \neq C_2 \in H$ has $|C_1 \cap C_2| \leq 1$; this "morally" holds with high probability provided that $m \ll n^2$ (and recall that we are working in the regime of $m \sim n^{1.5}$ or smaller, as for $m \gg n^{1.5}$ there is a polynomial-time refutation [AGK21]). More formally, when $m \ll n^2$, then with high probability over H, one can remove $o(m)$ constraints from H so that the remaining hypergraph satisfies this condition.

The construction of [WAM19] is as follows. First, partition the hyperedges H arbitrarily into $H_1, \ldots, H_n$, such that if $C \in H_u$ then $u \in C$. From now on, we shall think of H as $\cup_{u=1}^n H_u$. We note that our lower bound will hold regardless of the choice of the partition here.

Next, let ϕ be the polynomial $\phi(x) := \frac{1}{m} \sum_{C \in H} b_C x_C$, where $x_C := \prod_{i \in C} x_i$. Applying the Cauchy-Schwarz inequality, we have that

$$\phi(x)^2 \leq \frac{1}{m} \sum_{u=1}^{n} x_u^2 + \frac{n}{m^2} \sum_{u=1}^{n} \sum_{C \neq C' \in H_u} b_C b_{C'} x_{C \setminus \{u\}} x_{C' \setminus \{u\}} = \frac{n}{m} + f(x)\,,$$

where $f(x) := \frac{n}{m^2} \sum_{u=1}^{n} \sum_{C \neq C' \in H_u} b_C b_{C'} x_{C \setminus \{u\}} x_{C' \setminus \{u\}}$.

We now recall the following definition from [WAM19].

Definition 5.6.1. Let $\ell \in \mathbb{N}$, and let $H = \cup_{u=1}^n H_u$ be a 3-uniform hypergraph. For $\vec{S}, \vec{T} \in [n]^\ell$ and $C_1 = \{u, v_1, w_1\}, C_2 = \{u, v_2, w_2\} \in H_u$ with $\{v_1, w_1\} \cap \{v_2, w_2\} = \emptyset$, we write $\vec{S} \overset{C_1, C_2}{\leftrightarrow} \vec{T}$ if there exist $i \neq j \in [\ell]$ such that (1) $\vec{S}_t = \vec{T}_t$ for all $t \neq i, j$, and (2) $\{\vec{S}_i, \vec{S}_j\}$ contains exactly one element from each of $\{v_1, w_1\}$ and $\{v_2, w_2\}$, and $\{\vec{T}_i, \vec{T}_j\}$ contains the other two remaining elements. Here, $\vec{S}_i$ denotes the i-th element in the tuple $\vec{S} \in [n]^\ell$. We note that if $\vec{S} \overset{C_1, C_2}{\leftrightarrow} \vec{T}$ for some C_1, C_2, then we cannot have $\vec{S} \overset{C_1', C_2'}{\leftrightarrow} \vec{T}$ for any other pair C_1', C_2'.

Let $A_u \in \mathbb{R}^{n^\ell \times n^\ell}$ be the matrix where $A_u(\vec{S}, \vec{T}) = b_{C_1} b_{C_2}$ if $\vec{S} \overset{C_1, C_2}{\leftrightarrow} \vec{T}$ for some $C_1 \neq C_2 \in H_u$, and 0 otherwise, and let $A := \sum_{u=1}^n A_u$.

It is simple to observe that $\max_{x \in \{-1,1\}^n} f(x) \leq \frac{n}{m^2} \cdot O(\frac{n^2}{\ell^2}) \|A\|_2$, as $\frac{m^2}{n} f(x) = \frac{1}{4\binom{\ell}{2}(n-4)^{\ell-2}} (x^{\otimes \ell})^\top A x^{\otimes \ell}$ for all $x \in \{-1, 1\}^n$ because each pair $C_1 \neq C_2 \in H_u$ "appears" exactly $4\binom{\ell}{2}(n-4)^{\ell-2}$ times in the matrix A. Thus, in order to get the correct $m = n^{1.5}/\sqrt{\ell}$ trade-off, we need to show that $\|A\|_2 \leq O(\ell)$, with high probability over H and the b_C's.

We prove that $\|A\|_2$ is in fact *large* with high probability, and so the above approach of [WAM19] fails. Formally, we prove that with high probability, the matrix A has a spectral norm $\Omega(\min(\ell^2, \frac{m^2}{n^2}))$, which has the following implications. If the minimum is $\frac{m^2}{n^2}$, then the upper bound certified on f is $\Omega(n/\ell^2)$, and thus the upper bound certified on ϕ is $\Omega(\sqrt{n}/\ell)$. This is not very useful, as it is greater than 1 when $\ell \ll \sqrt{n}$. If the minimum is ℓ^2, then we certify a good upper bound on f (and therefore also ϕ) only if $m \geq n^{1.5}$, which is higher than the desired threshold of $n^{1.5}/\sqrt{\ell}$.

Proposition 5.6.2. *Let ϕ be a 3-XOR instance with n variables and m constraints, with constraint hypergraph $H = \cup_{u=1}^{n} H_u$ and coefficients $\{b_C\}_{C \in H}$. Suppose that $2n \leq m$, and that for every pair of constraints $C_1 \neq C_2 \in H$, it holds that $|C_1 \cap C_2| \leq 1$. Let $\ell \leq n$. Then, $\|A\|_2 \geq \binom{\ell'}{2}$, where $\ell' := \min(\lceil \frac{m}{2n} \rceil, \ell)$.*

We note that Proposition 5.6.2 holds regardless of the choice of the partitioning of H into the H_u's, and also for any choice of the b_C's (and so, in particular, for random b_C's). We also note that Proposition 5.6.2 essentially holds for a random H, provided that $m \ll n^2$, for the same reason mentioned earlier: when $m \ll n^2$, with high probability over H, after removing $o(m)$ constraints from H, the resulting hypergraph H' satisfies $|C_1 \cap C_2| \leq 1$ for all $C_1 \neq C_2 \in H'$.

Proof. As $m \geq 2n$, there must exist some variable $u \in [n]$ that appears in at least $\frac{m}{n}$ constraints. Hence, there must exist at least $\lceil \frac{m}{2n} \rceil$ constraints that include u and all have the same sign $b \in \{-1, 1\}$.

Let $\ell' := \min(\lceil \frac{m}{2n} \rceil, \ell)$. By the above, we have ℓ' constraints $\{C_i\}_{i \in [\ell']} = \{\{u, v_i, w_i\}\}_{i \in [\ell']}$ such that $b_{C_i} = b$ for all i. Furthermore, by assumption on H, we have $|C_i \cap C_j| \leq 1$ for all $i \neq j \in [\ell']$. As $u \in C_i \cap C_j$, it thus follows that $\{v_i, w_i\} \cap \{v_j, w_j\} = \emptyset$. Let $z \in [n]$ be arbitrary. Let $\mathcal{R}$ denote the set of tuples $(r_1, \ldots, r_{\ell'}, z, \ldots, z) \in [n]^\ell$ such that $r_i \in \{v_i, w_i\}$ for all $i \in [\ell']$. We note that the element z merely pads each tuple in $\mathcal{R}$ to have length exactly ℓ when $\ell' < \ell$.

Let M be the submatrix of A indexed by the tuples in $\mathcal{R}$. Note that M is a $2^{\ell'} \times 2^{\ell'}$ matrix, as $|R| = 2^{\ell'}$. Let $\vec{S} = (r_1, \ldots, r_{\ell'}, z, \ldots, z)$ be a row in M. We will show that each row of M has exactly $\binom{\ell'}{2}$ nonzero entries, each of which is 1.

First, let us consider the contribution to M from A_u. Fix a row $\vec{S} \in \mathcal{R}$. For each pair of indices $i \neq j \in [\ell']$, we can replace the i-th and j-th elements of $\vec{S}$ with the elements of $\{v_i, w_i\}$ and $\{v_j, w_j\}$ not used in $\vec{S}$, and this will yield some $\vec{T} \in \mathcal{R}$ with $\vec{S} \overset{\{u,v_i,w_i\},\{u,v_j,w_j\}}{\leftrightarrow} \vec{T}$. Hence, $A_u(\vec{S}, \vec{T}) = b^2 = 1$. Any other $\vec{T} \in \mathcal{R}$ will differ from $\vec{S}$ by at least 2 elements, and thus we must have $A_u(\vec{S}, \vec{T}) = 0$ for such $\vec{T}$.

Next, let us consider the contribution to M from $A_{u'}$ for $u' \neq u$. Fix a row $\vec{S} \in \mathcal{R}$. It suffices to only consider $\vec{T}$ obtained by swapping the i-th and j-th entries of $\vec{S}$, for some $i \neq j \in [\ell']$, as above. If $A_{u'}(\vec{S}, \vec{T})$ is nonzero, then we must have $\vec{S} \overset{\{u',v_i,w_i\},\{u',v_j,w_j\}}{\leftrightarrow} \vec{T}$, and thus that $\{u', v_i, w_i\}, \{u', v_j, w_j\} \in H_{u'}$. However, this implies that $|\{u, v_i, w_i\}, \{u', v_i, w_i\}| = 2 > 1$, which contradicts our assumption on H.

We have thus shown that the matrix M is $2^{\ell'} \times 2^{\ell'}$, with each row having exactly $\binom{\ell'}{2}$ nonzero entries, all of which are 1. It thus follows that $\|A\|_2 \geq \|M\|_2 \geq (1^{2^{\ell'}})^\top M 1^{2^{\ell'}} / 2^{\ell'} = \binom{\ell'}{2}$, which finishes the proof. □

Chapter 6

Short Refutation Witnesses for Smoothed CSPs Below the Spectral Threshold

In this chapter, we use our smoothed refutation algorithm along with our proof of Feige's conjecture to show the existence of polynomial size refutation witnesses below the spectral threshold for smoothed instances of Boolean CSPs. Modulo the use of our key new ingredients — Theorem 5.3.1 and Theorem 6 — the rest of the proof plan largely follows the influential work of Feige, Kim and Ofek [FKO06] who proved that *fully random* instances of 3-SAT admit polynomial size refutation witnesses whenever they have at least $\tilde{O}(n^{1.4})$ constraints. Our new ingredients allow us to (1) show a similar result for not just fully random instances, but also semirandom and smoothed ones, and (2) provide an arguably simpler refutation witness even for the fully random instances of 3-SAT studied by [FKO06].

Let us first formalize the idea of a *refutation witness*, or equivalently, a nondeterministic refutation algorithm.

Definition 6.0.1 (Nondeterministic refutation)**.** Fix $k \in \mathbb{N}$, and let $P : \{-1,1\}^k \to \{0,1\}$ be a predicate. We say that a nondeterministic algorithm V is an *nondeterministic efficient weak refutation algorithm* if V takes as input a CSP instance ψ with predicate P in n variables and m clauses and in $\mathrm{poly}(n,m)$-nondeterministic time outputs either "unsatisfiable" or "don't know", such that for every ψ, if $V(\psi)$ outputs "unsatisfiable" then ψ is unsatisfiable. If $V(\psi)$ outputs "unsatisfiable", then we say that V weakly refutes ψ. The string $\pi \in \{0,1\}^{\mathrm{poly}(n,m)}$ of nondeterministic guesses of V is called the weak refutation witness.

We will sketch a proof of the following theorem. We only provide a proof sketch, as the proof merely combines the ideas of [FKO06] with our theorems, Theorem 5.3.1 and Theorem 6.

Theorem 6.0.2. *Let $k \geq 3$, and let $P\colon \{-1,1\}^k \to \{0,1\}$ be a non-trivial predicate. Then there is a nondeterministic efficient weak refutation algorithm V with the following properties. Let ψ be an instance of a CSP with predicate P with n variables and m clauses, specified by a collection of m k-tuples H and literal patterns ξ. Then:*

(1) If ψ is a uniformly random instance with $m \geq \tilde{O}(1) \cdot n^{\frac{k}{2} - \frac{k-2}{2(k+2)}}$ clauses, then V weakly refutes ψ with probability at least $1 - 1/\mathrm{poly}(n)$.

(2) If ψ is a semirandom instance with $m \geq \tilde{O}(1) \cdot n^{\frac{k}{2} - \frac{k-2}{2(k+8)}}$ clauses, then V weakly refutes ψ with probability at least $1 - 1/\mathrm{poly}(n)$.

(3) *If ψ is a smoothed instance obtained using smoothing parameters $\vec{p} = \{p_{C,i}\}_{C\in H, i\in[k]}$ with $m \geq \tilde{O}(1)\cdot n^{\frac{k}{2}-\frac{k-2}{2(k+8)}}/q(\vec{p})$ clauses, where $q(\vec{p}) := \frac{1}{m}\sum_{C\in H}\prod_{i\in C} p_{C,i}$, then V weakly refutes ψ with probability at least $1 - 1/\mathrm{poly}(n)$.*

Finally, if $k = 3$, the threshold of m for the semirandom/smoothed case can be improved to $\tilde{O}(n^{1.4})$ and $\tilde{O}(n^{1.4})/q(\vec{p})$, respectively, matching the random case.

We will first begin by focusing on the case of k-XOR. As in the case of Section 5.5, refuting arbitrary predicates P will reduce to refuting XOR.

In [FKO06], FKO observed that the following type of refutation witnesses, which we shall call *ideal FKO witnesses*, allow for a non-trivial[1] weak refutation of instances of k-XOR whenever the b_C's are chosen uniformly and independently at random. Informally speaking, ideal FKO witnesses are simply a disjoint collection of even covers in H.

Definition 6.0.3 (Ideal FKO witnesses)**.** Let H be k-uniform hypergraph on $[n]$. We say that a collection of even covers $E_1, E_2, \ldots, E_r \subseteq H$ is an *ideal FKO witness of length h* if each $E_i \cap E_j = \emptyset$ for every $i \neq j$ and $|E_i| \leq h$ for every i, where $|E_i|$ denotes the length of the even cover E_i. The size of the witness is $s = \sum_{i=1}^{r}|E_i| \leq hr$.

Ideal FKO witnesses yield non-trivial weak refutation witnesses for semi-random instances of k-XOR.

Lemma 6.0.4 (Ideal FKO witnesses yield refutation witnesses for XOR)**.** *Let $\psi = (H, b)$ be an instance of k-XOR on n variables. Suppose $E_1, E_2, \ldots, E_r \subseteq H$ is an ideal FKO witness in H. Suppose further that each b_C is a uniformly random and independent bit in ± 1. Then, with probability at least $1 - \exp(\Omega(r))$ over the draw of $b = \{b_C\}_{C\in H}$, $\mathrm{val}(\psi) \leq 1 - \frac{r}{3m}$.*

Proof. For each i, consider $Z_i = \prod_{C\in E_i} b_C$. Then, notice that $Z_1, Z_2, \ldots, Z_r$ are independent random variables, each uniformly drawn from $\{-1, 1\}$. Thus, by a Chernoff bound, with probability at least $1 - \exp(\Omega(r))$ there must exist at least $r/3$ E_i's such that $Z_i = -1$. Consider any such E_i where this holds.

Suppose some $x \in \{-1, 1\}^n$ satisfies all the constraints in ψ corresponding to k-tuples $C \in E_i$. Then, $\prod_{C\in E_i} b_C = \prod_{C\in E_i}\prod_{j\leq k} x_{C_j}$. Since E_i is an even cover, every variable occurs an even number of times in the C's in E_i. Since even powers of any x_j evaluate to 1, the RHS above must evaluate to 1. Since we know that $\prod_{C\in E_i} b_C = -1$, this implies that such an x cannot exist: every x must violate at least one constraint in each E_i if $\prod_{C\in E_i} b_C = -1$. Since E_i's are disjoint, this implies that every x violates at least $r/3$ constraints in ψ. The bound on $\mathrm{val}(\psi)$ now follows. □

The key question is whether Ideal FKO witnesses exist in the k-uniform hypergraph specifying the k-XOR instance. In [FKO06], the authors study the question of finding such refutation witnesses in *random* sufficiently dense hypergraphs. They comment that, while they expect Ideal FKO witnesses to exist in the regime they are working in, proving that they exist appears hard. They instead show that a related form of witnesses (these are "almost disjoint" even covers instead of perfectly disjoint) exist by means of a sophisticated second moment method argument.

Here, we show that Ideal FKO witnesses do indeed exist – not only in random dense hypergraphs but in *arbitrary* hypergraphs with the same density. Indeed, this follows almost immediately from Theorem 6.

[1] Note that by running Gaussian elimination, one can decide if a k-XOR instance is unsatisfiable in polynomial time. This is a *trivial* weak refutation.

Lemma 6.0.5. *Fix $k \in \mathbb{N}$ and $\ell = \ell(n)$. Let H be any k-uniform hypergraph with $m \geq 2m_0$ hyperedges, where $m_0 = \Gamma^k \cdot n \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \log n$ is the threshold appearing in Theorem 6. Then, H contains a collection of $m_0/h(n)$ hyperedge-disjoint even covers each of length at most $h(n) = O(\ell \log n)$.*

Proof. The idea is simple. Let m_0 be the number of constraints required in Theorem 6. Choose $m = 2m_0$. Then, by an application of Theorem 6, there is an even cover in H, say, E_1 of size $|E_1| \leq h(n) = O(\ell \log n)$. Let $H_0 = H$. We now repeat the following process for $i = 1, 2, \ldots, r$: apply Theorem 6 to $H_i := H_{i-1} \setminus E_i$ to find an even cover $E_{i+1} \subseteq H_i$ of size $\leq h(n) = O(\ell \log n)$. Notice that the conditions of Theorem 6 are met so long as $|H_i| \geq m - h(n)r \geq m/2$, i.e., if $r \leq 0.5m/h(n)$. Further, each of the even covers $E_1, E_2, \ldots, E_r$ are pairwise disjoint by construction. This completes the proof. □

By combining the above observation with semirandom refutation algorithms, one can show that Ideal FKO witnesses yield weak refutation witnesses for all k-CSPs at densities polynomially below $n^{k/2}$. This is one of the key insights of FKO [FKO06] – to use the non-trivial weak refutation offered by (their variant of) ideal FKO witnesses in order to show the existence of polynomial size weak-refutation witnesses for random 3-SAT with $m = \tilde{\Omega}(n^{1.4})$ constraints: namely, in a regime of m where known spectral algorithms, and more generally those based on the polynomial-time canonical sum-of-squares relaxation, provably fail. Theorem 6 (and its consequence Lemma 6.0.5) implies that the same result holds for *arbitrary* constraint hypergraphs, up to additional polylog(n) factors in the number of constraints.

Lemma 6.0.6 (Ideal FKO witnesses yield weak refutation witnesses for 3-SAT)**.** *Let $\psi = (H, \xi)$ be an instance of 3-SAT described by a 3-uniform hypergraph H on $[n]$ with $m \geq \tilde{O}(n^{1.4})$ arbitrary constraints and uniformly randomly generated literal patterns. Then, with probability at least $1 - 1/\mathrm{poly}(n)$ over the draw of the literal patterns in the instance, there is a polynomial-size refutation witness that certifies $\mathrm{val}(\psi) < 1$.*

Proof Sketch. Let $P : \{-1, 1\}^3 \to \{0, 1\}$ be the 3-SAT predicate. Then, $P(z) = \frac{7}{8} + \frac{1}{8}(z_1 + z_2 + z_3) - \frac{1}{8}(z_1 z_2 + z_2 z_3 + z_1 z_3 - z_1 z_2 z_3)$. We write

$$\begin{aligned}
\psi(x) &= \frac{1}{|H|} \sum_{C \in H} P(x_{C_1} \xi_{C,1}, x_{C_2} \xi_{C,2}, x_{C_3} \xi_{C,3}) \\
&= \frac{7}{8} + \frac{1}{8|H|} \sum_{C \in H} (\xi_{C,1} x_{C_1} + \xi_{C,2} x_{C_2} + \xi_{C,3} x_{C_3} - \xi_{C,1} x_{C_1} \xi_{C,2} x_{C_2} - \xi_{C,2} x_{C_2} \xi_{C,3} x_{C_3} \\
&\quad - \xi_{C,1} x_{C_1} \xi_{C,3} x_{C_3} + \xi_{C,1} \xi_{C,2} \xi_{C,3} x_{C_1} x_{C_2} x_{C_3}) .
\end{aligned}$$

where the $\xi_{C,i}$'s are the literal negation patterns in $\{-1, 1\}$. Note that $\psi(x)$ computes the fraction of constraints satisfied by the assignment $x \in \{-1, 1\}^n$. We refute each of the 7 different XOR instances produced by taking each of the 7 non-constant terms in the expansion of P as a multilinear polynomial above separately.

Our refutation witness helps us efficiently refute each of the instances corresponding to the 7 terms in the expansion above. Specifically, by collecting coefficients together, each the first three terms each produce a linear polynomial of the form $\sum_i B_i x_i$. The next three terms each produce a homogenous quadratic polynomial of the form $\frac{1}{|H|} \sum_{C \in H} B_C x_{C_1} x_{C_2}$, and finally the last term is a cubic polynomial of the form $\frac{1}{|H|} \sum_{C \in H} B_C x_{C_1} x_{C_2} x_{C_3}$. Our refutation witness for each linear

polynomial is simply $\|B\|_1$, where $B = (B_1, \ldots, B_n)$, noting that this is exactly the maximum of the first kind of terms as x varies over the hypercube. For the quadratic case, our refutation witness is the value of SDP relaxation for the $\infty \to 1$-norm that gives a < 2 factor approximation to maximum of bilinear forms over the hypercube. For the homogeneous degree 3 term, our witness is an ideal FKO witness guaranteed by Lemma 6.0.5.

By Chernoff and union bound argument (applied to every assignment in $\{-1,1\}^n$), $\|B\|_1$ for any linear term above is at most $O(\sqrt{n/m})$.

By Chernoff and union bound argument, the $\infty \to 1$-norm of the matrix defining the 2-XOR constraints is at most $O(\sqrt{n/m})$. By Grothendieck's inequality (Fact 3.5.4), we can certify this value efficiently (with an additional loss of at most a factor of < 2) using an SDP.

Thus, we can certify an upper bound of $O(\sqrt{n/m})$ on all but homogeneous degree 3 polynomial produced in the Fourier expansion above. When $m \geq \tilde{\Omega}(n)n^{0.5(1-\delta)}$, i.e., $\ell = n^\delta$, by Lemma 6.0.5, H has a collection of $\frac{m}{\tilde{O}(n^\delta)}$ pairwise disjoint even covers of length at most $\tilde{O}(n^\delta)$. By Chernoff bounds, at least $\frac{1}{3}$ of these even covers must violated and thus, we have obtained a certificate for an upper bound of $1 - \frac{1}{\tilde{O}(n^\delta)}$ on the value of the final term.

Putting these upper bounds together gives an upper bound of $\frac{7}{8} + \frac{1}{8}O(\sqrt{\frac{n}{m}}) + \frac{1}{8}(1 - \frac{1}{\tilde{O}(n^\delta)})$ on the value of the 3-SAT instance. For $\delta = 0.2$, we observe that $\sqrt{\frac{n}{m}} = \tilde{O}(-n^{0.25+\delta/4}) \ll \frac{1}{\tilde{O}(n^\delta)}$. Thus, for $m \geq \tilde{O}(n^{1.4})$, with probability at least $1 - 1/\mathrm{poly}(n)$, we obtain a refutation for the input 3-SAT instance. □

Lemma 6.0.6 generalizes to all k-CSPs with predicate P, provided that P is non-trivial, i.e., P is not identically 1. We only need the following basic fact (and the rest of the proof remains the same as above), as well as known results for spectral refutation of *random* $k-1$ and smaller-arity XOR instances.

Lemma 6.0.7 (Highest Fourier Coefficient of Boolean Functions)**.** *Let $P : \{-1,1\}^k \to \{0,1\}$. Let $\sum_{S\subseteq[k]} \hat{P}(S)x_S$ be the Fourier polynomial representation of P. Then, $\hat{P}(\emptyset) + |\hat{P}([k])| \leq 1$.*

Proof. For each $b \in \{-1,1\}$, consider the distribution that is uniform on all x such that $\prod_i x_i = b$. Then, the expectation of P on this distribution is exactly $\hat{P}(\emptyset) + b\hat{P}([k])$. On the other hand, since P takes values in $\{0,1\}$, this expectation cannot exceed 1. Thus, $1 \geq \hat{P}(\emptyset) + b\hat{P}([k])$ for both values of b and in particular, $1 \geq \hat{P}(\emptyset) + |\hat{P}([k])|$ as desired. □

We now sketch a proof of the generalization of Lemma 6.0.6 to all *fully random* CSPs. This is captured by Item (1) in Theorem 6.0.2. We will assume that the Fourier coefficient $\hat{P}([k])$ is nonzero, as otherwise by Theorem 5.5.4, we have enough constraints to give a polynomial time *deterministic* refutation.[2]

Lemma 6.0.8 (Polynomial Size Refutation Witnesses for all *random* k-CSPs)**.** *Let $P : \{-1,1\}^k \to \{0,1\}$ be an arbitrary k-ary Boolean predicate for $k \geq 3$. Let ψ be a CSP instance with predicate P specified by $H-$ a collection of uniformly at random and independently generated $m \geq m_0 = \tilde{O}(1) \cdot n^{\frac{k}{2} - \frac{k-2}{2(k+2)}}$ k-tuples and uniformly random and independently generated literal patterns $\{\xi(C,i)\}_{C\in H, i\in[k]}$. Then,*

[2]This is because there cannot be a $(k-1)$-uniform distribution μ supported on $P^{-1}(1)$, as otherwise we would have $1 = \mathbb{E}_{x\sim\mu}[P(x)] = \hat{P}(\emptyset) < 1$, where we have $\hat{P}(\emptyset) < 1$ as P is nontrivial. And then we observe that the CSP instance has at least $\tilde{O}(n^{\frac{k}{2} - \frac{k-2}{2(k+2)}})$ constraints, which is at least $\tilde{O}(n^{\frac{k-1}{2}})$.

with probability at least $1 - 1/\mathrm{poly}(n)$ *over the draw of* H *and* $\xi(C,i)$*'s, there exists a polynomial size refutation witness for* ψ*.*

Proof. Observe that the instance ψ has $m = \tilde{O}(1) \cdot \left(\frac{n}{\ell}\right)^{k/2} \ell$ constraints for $\ell \leq \tilde{O}(n^{\frac{1}{k+2}})$. We now use Fourier analysis to decompose $\psi(x) := \frac{1}{|H|} \sum_{C \in H} P(x_{C_1}\xi_{C,1}, \ldots, x_{C_k}\xi_{C,k})$ into 2^k polynomials, each of degree $t \leq k$. We use the same certificate as in Lemma 6.0.6 for the linear polynomials appearing in this decomposition. For quadratic and higher degree ($\leq k-1$) terms, we now use spectral refutation from prior results on refuting fully random CSPs, such as Theorem 1 in [AOW15]. Each degree t polynomial (with $t \leq k-1$) that appears requires at least $\tilde{O}(n^{t/2}/\varepsilon^2)$ constraints to certify an upper bound of ε on its value; we can thus certify an upper bound of $\varepsilon = \sqrt{\frac{n^{(k-1)/2}}{m}}$ on each polynomial. Note that by choice of m, we have $\varepsilon \leq 1$.

Finally, to refute the final and highest degree polynomial obtained by taking the $[k]$-indexed Fourier coefficient of P, we use the Ideal FKO witness from Lemma 6.0.4. Then, as in the argument for 3-SAT above, we arrive at a certificate that (with probability at least $1 - 1/\mathrm{poly}(n)$) certifies an upper bound of $\hat{P}(\emptyset) + \tilde{O}(\sqrt{\frac{n^{(k-1)/2}}{m}}) + |\hat{P}([k])| \cdot (1 - \frac{\tilde{O}(1)}{\ell \log n})$ on the value of ψ, using Lemma 6.0.5. The size of the witness is $s(n) \leq m_0 = \mathrm{poly}(n)$, as the degree $< k$ terms used deterministic refutations. Using Lemma 6.0.7, we thus certify an upper bound of $1 + \tilde{O}(\sqrt{\frac{n^{(k-1)/2}}{m}}) - \frac{\tilde{O}(1)}{\ell \log n} = 1 - o(1)$ on $\psi(x)$, which finishes the proof. Note that this is indeed $1 - o(1)$ as $\tilde{O}(1)\sqrt{\frac{n^{(k-1)/2}}{m}} = \tilde{O}(1) \cdot \ell^{\frac{k}{4} - \frac{1}{2}} / n^{\frac{1}{4}} \ll \tilde{O}(1/\ell)$, since $\ell \leq \tilde{O}(1) n^{\frac{1}{k+2}}$. □

By switching the CSP refutation algorithms in [AOW15] with the semirandom refutation algorithm from Theorem 5.3.1 in this work, we arrive at Item (2) of Theorem 6.0.2, a version of the above result that shows the existence of polynomial size refutation witnesses below the $n^{k/2}$-threshold for *semirandom* instances. As the proof is very similar, we omit the details of the proof; the final bound is stated in Item (2). Note that the precise value of m at which this refutation succeeds is strictly larger (though still polynomially smaller than $n^{k/2}$) than the one in Lemma 6.0.8, i.e., Item (1). The difference comes from the fact that the dependence on ϵ (the strength of the refutation) in our semirandom refutation algorithms grows as $1/\epsilon^5$ instead of the $1/\epsilon^2$ dependence of algorithms for fully random instances; we thus have to take $\varepsilon = \left(n^{(k-1)/2}/m\right)^{1/5}$ instead of $\left(n^{(k-1)/2}/m\right)^{1/2}$, which in turn makes $\ell = n^{1/(k+8)}$ and then $m \geq \tilde{O}(1) n^{\frac{k}{2} - \frac{k-2}{2(k+8)}}$. Our belief is that the $1/\varepsilon^5$ dependence is sub-optimal in the semirandom setting but inherent to our current proof techniques.

We note that for large k, the density required for the polynomial size refutation witnesses to exist in both Item (1) and Item (2) is $\sim n^{\frac{k}{2} - 0.5 + o_k(1)}$, effectively giving a $\sqrt{n}$ factor "win" over the threshold at which spectral (and sum-of-squares based methods more generally) succeed.

In the specific case of $k = 3$, we can improve the bound in the semirandom case to match the $\tilde{O}(n^{1.4})$ achieved in the random case. This is because the instances appearing in the decomposition are all semirandom 2-XOR instances, and we can refute these instances with the correct $1/\varepsilon^2$ dependence: see Proposition 5.2.2 and Theorem 5.2.3 in [Wit17], combined with the fact that the value of a semirandom 2-XOR instance is at most $\frac{1}{2} + \varepsilon$ when $m \gg n/\varepsilon^2$.

Finally, to handle Item (3), we observe that by Chernoff bound, if $m \geq O(1) m_0 / q(\vec{p})$, where $m_0 = \tilde{O}(1) \cdot n^{\frac{k}{2} - \frac{k-2}{2(k+8)}}$, then with high probability there are at least m_0 clauses in ψ where all literals

in the clause are re-randomized by the smoothing process. Call this subinstance ψ'. As ψ' is semirandom, by Item (2) there is a weak refutation for ψ'. As we can nondeterministically guess ψ', it follows that the smoothed instance ψ also has a weak refutation.

We note that technically speaking, the smoothed nondeterministic refutation algorithm V is different than the V for the random/semirandom settings, as it has the additional step of guessing ψ'. However, we can use the V for the smoothed case also in the random/semirandom settings, by simply guessing $\psi' = \psi$.

Chapter 7

Efficient Algorithms for Semirandom Planted CSPs at the Refutation Threshold

In this chapter, we will prove Theorem 4. First, in Section 7.1, we give intuition and an overview for the proof. Then, in Section 7.2, we prove Theorem 4 from Theorem 5 by reducing semirandom planted CSPs to noisy XOR. In Sections 7.3 and 7.4, we prove Theorem 5, following the blueprint that we will explain in Section 7.1.

7.1 Technical overview

In this section, we give an overview of the proof of Theorem 5 and our algorithm for noisy planted k-XOR. We defer discussion of the reduction from general k-CSPs to k-XOR used to obtain Theorem 4 to Section 7.2. There, we explain the additional challenges encountered in the semirandom case as compared to the random case [FPV15, Section 4]. Somewhat surprisingly, the reduction is complicated and quite different from the random planted case or even the semirandom refutation setting, where the reduction to XOR is straightforward.

We now explain Theorem 5. As is typical in algorithm design for k-XOR, the case when k is even is considerably simpler than when k is odd. For the purpose of this overview, we will focus mostly on the even case, and only briefly discuss the additional techniques for odd k in Section 7.1.5.

Notation. We will use the following notation in this chapter. Given a k-XOR instance ψ on hypergraph $H \subseteq \binom{[n]}{k}$ with $m = |H|$ and right-hand sides $\{b_C\}_{C \in H}$, we define $\psi(x) \coloneqq \sum_{C \in H} b_C \prod_{i \in C} x_i$ to be a degree-k polynomial mapping $\{-1,1\}^n \to [-m, m]$. We note that $\mathrm{val}_\psi(x) = \frac{1}{2} + \frac{1}{2m}\psi(x) \in [0,1]$ is the fraction of constraints in ψ satisfied by x. Moreover, we will write $x_C \coloneqq \prod_{i \in C} x_i$.

Unless otherwise stated, we will use ϕ to denote a 2-XOR instance and ψ to denote a k-XOR instance for any $k \geq 2$.

We note that for even arity k-XOR, we have $\mathrm{val}_\psi(x) = \mathrm{val}_\psi(-x)$, and so it is only possible for the optimal solution to be unique *up to a global sign*. We will abuse terminology and say that x^* is the unique optimal assignment if $\pm x^*$ are the only optimal assignments, and we will say that we have recovered x^* exactly if we obtain one of $\pm x^*$.

7.1.1 Approximate recovery for 2-XOR from refutation

First, let us focus on the case of $k = 2$, the simplest case, and let us furthermore suppose that we only want to achieve the weaker goal of recovering an assignment of value $1 - \eta - o(1)$. (Note that we do need the stronger guarantee of Theorem 5 to solve general planted CSPs in Theorem 4.)

For 2-XOR, this goal is actually quite straightforward to achieve using 2-XOR refutation as a blackbox. Let us represent the 2-XOR instance ϕ as a graph G on n vertices, along with right-hand sides b_{ij} for each edge $(i, j) \in E$. Recall that we have $b_{ij} = x^*_i x^*_j$ with probability $1 - \eta$, and $b_{ij} = -x^*_i x^*_j$ otherwise. Note that by concentration, $\mathrm{val}_\phi(x^*) = 1 - \eta \pm o(1)$ with high probability.

We now make the following observation. Let us suppose that we sample the noise in two steps: first, we add each $(i, j) \in E$ to a set E' with probability 2η independently; then for each $(i, j) \in E'$ we set b_{ij} to be uniformly random from $\{-1, 1\}$. Using known results for semirandom 2-XOR refutation, it is possible to certify, via an SDP relaxation, that no assignment x can satisfy (or violate) more than $\frac{1}{2} + o(1)$ fraction of the constraints in E'.

Thus, we can simply solve the SDP relaxation for ϕ and obtain a degree-2 pseudo-expectation $\tilde{\mathbb{E}}$ in the variables $x_1, \dots, x_n$ over $\{-1, 1\}^n$ that maximizes $\phi(x)$. Let $\phi_{E'}$ be the subinstance containing only the constraints in E', and let $\phi_{E \setminus E'}$ be the subinstance containing only the constraints in $E \setminus E'$, which are uncorrupted. We have $\tilde{\mathbb{E}}[\mathrm{val}_\phi(x)] \geq 1 - \eta - o(1)$, and the guarantee of refutation implies that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E'}}(x)] \leq \frac{1}{2} + o(1)$. As $\mathrm{val}_\phi(x) = (1 - 2\eta) \cdot \mathrm{val}_{\phi_{E \setminus E'}}(x) + 2\eta \cdot \mathrm{val}_{\phi_{E'}}(x)$, we therefore have that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E \setminus E'}}(x)] \geq 1 - o(1)$, i.e., $\tilde{\mathbb{E}}$ satisfies $1 - o(1)$ fraction of the constraints in $E \setminus E'$. Then, applying the standard Gaussian rounding, we obtain an x that satisfies $1 - \sqrt{o(1)}$ fraction of the constraints in $E \setminus E'$ and thus has value $\mathrm{val}_\phi(x) \geq 1 - \eta - o(1)$ (as any x must satisfy at least $\frac{1}{2} - o(1)$ fraction of the constraints in E', with high probability over the noise).

One interesting observation is that in the above discussion, we can additionally allow E' to be an *arbitrary* subset of E of size $2\eta m$. Indeed, this is because the rounding only "remembers" that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E \setminus E'}}(x)]$ has value $1 - o(1)$. As we shall see shortly, this is the key reason that the reduction breaks down for k-XOR.

7.1.2 The challenges for k-XOR and our strategy

Unfortunately, the natural blackbox reduction to refutation given in Section 7.1.1 does not generalize to k-XOR for $k \geq 3$. Following the approach described in the previous section, given a k-XOR instance ψ, one can solve a sum-of-squares SDP and obtain a pseudo-expectation $\tilde{\mathbb{E}}$ where $\tilde{\mathbb{E}}[\mathrm{val}_\psi(x)] \geq 1 - \eta - \delta$ and $\tilde{\mathbb{E}}[\mathrm{val}_{\psi_{E \setminus E'}}(x)] \geq 1 - \delta$ as before, where $\delta \sim 1/\mathrm{polylog}(n)$ when $m \gtrsim n^{k/2}$, due to the guarantees of refutation algorithms [AGK21]. However, unlike 2-XOR where we have Gaussian rounding, for k-XOR there is no known rounding algorithm that takes a pseudo-expectation $\tilde{\mathbb{E}}$ with $\tilde{\mathbb{E}}[\mathrm{val}_{\psi_{E \setminus E'}}(x)] \geq 1 - \delta$ and outputs an assignment x such that $\mathrm{val}_{\psi_{E \setminus E'}}(x) \geq 1 - f(\delta)$, for some $f(\cdot)$ such that $f(\delta) \to 0$ as $\delta \to 0$. In fact, if we only "remember" that $\psi_{E \setminus E'}$ has value $1 - \delta$, then it is NP-hard to find an x with value $> 1/2 + \delta$ even when $\delta = n^{-c}$ for some constant $c > 0$, assuming a variant of the Sliding Scale Conjecture [BGLR93][1] (see e.g. [MR10, Mos15] for more details).

As we have seen, while semirandom k-XOR refutation allows us to efficiently approximate

[1] Note that we do need the Sliding Scale Conjecture, as the hardness shown in [MR10] is not strong enough; it only proves hardness for $\delta \geq (\log \log n)^{-c}$, whereas we have $\delta \sim 1/\mathrm{polylog}(n)$.

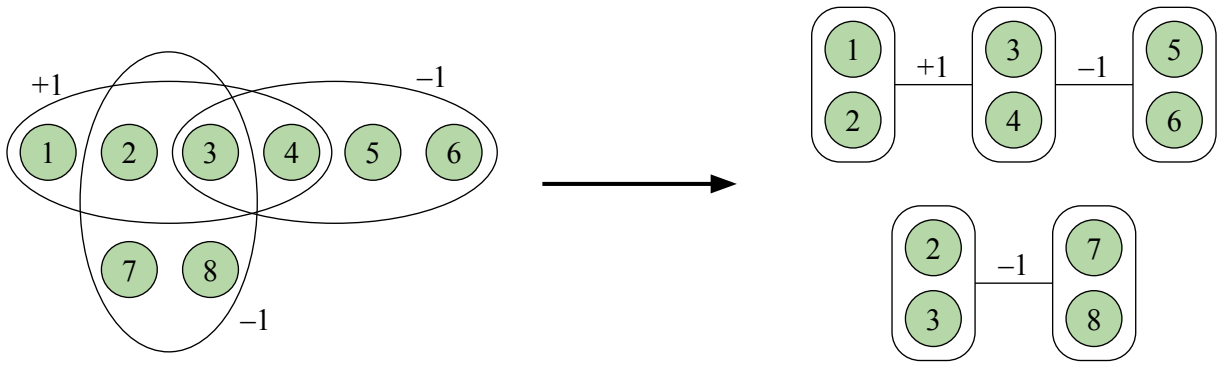

Figure 7.1: An example of the 2-XOR instance ϕ from a 4-XOR instance ψ.

and certify the *value* of the planted instance, the challenge lies in the *rounding* of the SDP, where the goal is to recover an assignment x. This is a technical challenge that does not arise in the context of CSP refutation, as there we are merely trying to bound the value of the instance. As a result, new ideas are required to address this challenge.

Reduction from k-XOR to 2-XOR for even k. One could still consider the following natural approach. For simplicity, let $k = 4$. Given a 4-XOR instance ψ, we can write down a natural and related 2-XOR instance ϕ, as follows.

Definition 7.1.1 (Reduction to 2-XOR)**.** Let ψ be a 4-XOR instance, and let ϕ be the 2-XOR defined as follows. The variables of ϕ are $y_{\{i,j\}}$ and correspond to *pairs* of variables $\{x_i, x_j\}$, and for each constraint $x_i x_j x_{i'} x_{j'} = b_{i,j,i',j'}$ in ψ, we split $\{i, j, i', j'\}$ into $\{i, j\}$ and $\{i', j'\}$ arbitrarily and add a constraint $y_{\{i,j\}} y_{\{i',j'\}} = b_{i,j,i',j'}$ to ϕ. See Fig. 7.1 for an example. This reduction easily generalizes to k-XOR for any even k.

By following the approach for 2-XOR described in Section 7.1.1, we can recover an assignment y that satisfies $1 - \eta - o(1)$ fraction of the constraints in ϕ. However, we need to recover an assignment x to the original k-XOR ψ, and it is quite possible that while y is a good assignment to ϕ, *it is not* close to $x^{\otimes 2}$ for *any* $x \in \{-1, 1\}^n$. If this happens, we will be unable to recover a good assignment to the 4-XOR instance ψ.

The key reason that this simple idea fails is because, unlike for random noisy XOR, the assignment y recovered is *not* necessarily unique, and we cannot hope for it to be in the semirandom setting! For random noisy XOR, one can argue that with high probability, y will be equal to $x^{*\otimes 2}$, and then we can immediately decode and recover x^* up to a global sign, i.e., we recover $\pm x^*$. But for semirandom instances, the situation can be far more complex.

Approximate 2-XOR recovery does not suffice for 4-XOR. When constructing the 2-XOR instance ϕ from the 4-XOR ψ (Definition 7.1.1), it may be the case that ϕ can be partitioned into multiple disconnected clusters (or have very few edges across different clusters), even when the hypergraph H of ψ is connected; see Fig. 7.1 for example. By the algorithm described in Section 7.1.1, we can get an assignment y that satisfies $1 - \eta - o(1)$ fraction of the constraints within each cluster.

The main challenge is to combine the information gathered from each cluster to recover an assignment x for the original 4-XOR ψ. Unfortunately, we do not know of a way to obtain a good assignment x based solely on the guarantee that y satisfies $1 - \eta - o(1)$ fraction of constraints in

each cluster. The issue occurs because the same variable $i \in [n]$ can appear in different clusters, e.g., $y_{\{1,2\}}$ and $y_{\{2,3\}}$ lie in different clusters in Fig. 7.1, and the recovered assignments in each cluster may implicitly choose different values for x_i because of the noise. Indeed, even if the local optimum is consistent with x^*, there can still be multiple "good" assignments that achieve $1 - \eta - o(1)$ value on the subinstance restricted to a cluster. So, unless the SDP can certify unique optimality of x^*, standard rounding techniques such as Gaussian rounding will merely output a "good" y, which may be inconsistent with x^* and thus can choose inconsistent values of x_i across the different clusters.

Exact 2-XOR recovery implies exact 4-XOR recovery. This leads to our main insight: if the subinstance of ϕ admits a *unique* local optimal assignment y^* (restricted to the cluster) that matches the planted assignment up to a sign, i.e., $y^*_{\{i,j\}} = \pm x^*_i x^*_j$, then for each edge in the cluster we know $y^*_{\{i,j\}} y^*_{\{i',j'\}} = x^*_i x^*_j x^*_{i'} x^*_{j'}$, and so the local constraints that are violated must be exactly the corrupted ones. Moreover, if the SDP can certify the uniqueness of the local optimal assignment for a cluster, then the SDP solution will be a *rank* 1 *matrix* $y^* y^{*\top}$, and so we can precisely identify which constraints in ϕ are corrupted. By repeating this for every cluster, we can identify all corrupted constraints in the original 4-XOR ψ (except for the small number of "cross cluster" edges), and thus achieve the guarantee stated in Theorem 5.

The general algorithmic strategy. The above discussion suggests that given a k-XOR instance ψ, we should first construct the 2-XOR ϕ, and then decompose the constraint graph G of ϕ into pieces in some particular way so that the induced local instances have unique solutions. Namely, the examples suggest the following algorithmic strategy.

Strategy 1 (Algorithm Blueprint for even k)**.** Given a noisy k-XOR instance ψ with planted assignment x^* and m constraints, we do the following:

(1) Construct the 2-XOR instance ϕ described in Definition 7.1.1, which is a noisy 2-XOR on $n^{k/2}$ variables with planted assignment y^*. Moreover, there is a one-to-one mapping between constraints in ϕ and ψ.

(2) Let G be the constraint graph of ϕ. Decompose G into subgraphs $G_1, \ldots, G_T$ while only discarding a $o(1)$-fraction of edges such that each subgraph G_i satisfies "some property". For each subgraph G_i, we define ϕ_i to be the subinstance of ϕ corresponding to the constraints in G_i. The goal is to identify a local property that the G_i's satisfy so that (1) we can perform the decomposition efficiently, and (2) for each subinstance ϕ_i, we can "recover y^* locally", i.e., we can find an assignment $y^{(i)}$ to the 2-XOR instance ϕ_i that is consistent with the planted assignment y^*.

(3) As each $y^{(i)}$ is consistent with y^*, the constraints in ϕ_i violated by $y^{(i)}$ must be precisely the corrupted constraints in ϕ_i. Hence, for the constraints that appear in one of the ϕ_i's, we have determined exactly which ones are corrupted.

(4) We have thus determined, for all but $o(m)$ constraints, precisely which ones are corrupted in the original k-XOR instance ψ. (Note that this is the *stronger* guarantee that we achieve in Theorem 5.) By discarding the corrupted constraints along with the $o(m)$ constraints where we "give up", we thus obtain a system of k-sparse linear equations with $m(1 - \eta - o(1))$ equations that has at least one solution (namely x^*), and so by solving it we obtain an x with $\mathrm{val}_\psi(x) \geq 1 - \eta - o(1)$.

7.1.3 Information-theoretic exact recovery from relative cut approximation

Following Strategy 1, the first technical question to now ask is: given a noisy 2-XOR instance ϕ with n variables, $m \gg n$ constraints, and planted assignment x^*, what conditions do we need to impose on the constraint graph G so that we can recover x^* (up to a sign) exactly? As a natural first step, we investigate what conditions are required so that we can accomplish this *information-theoretically*.

Fact 7.1.2. *Let $G = (V, E_G)$ be an n-vertex graph, and let $H = (V, E_H)$ be a subgraph of G where $E_H \subseteq E_G$. Let L_G, L_H be the* unnormalized *Laplacians of G and H. Consider a noisy planted 2-XOR instance ϕ on G with planted assignment $x^* \in \{-1,1\}^n$ (Definition 4.2.2), and suppose E_H is the set of corrupted edges. Suppose that for every $x \in \{-1,1\}^n \setminus \{\vec{1}, -\vec{1}\}$, it holds that $x^\top L_H x < \frac{1}{2} x^\top L_G x$. Then, x^* and $-x^*$ are the only two optimal assignments to ϕ.*

Note that the condition $x^\top L_H x < \frac{1}{2} x^\top L_G x$ for $x \notin \{\vec{1}, -\vec{1}\}$ implies that G is connected, as otherwise L_G has a kernel of dimension ≥ 2, which would contradict this assumption.

Proof. Let $x \in \{-1,1\}^n$ be any assignment. We wish to show that $\phi(x)$ is uniquely maximized when $x = x^*, -x^*$. We observe that

$$\phi(x) = \sum_{(i,j)\in E_G} x_i x_j b_{ij} = \sum_{(i,j)\in E_G} x_i x_j x_i^* x_j^* - 2 \sum_{(i,j)\in E_H} x_i x_j x_i^* x_j^* \ .$$

Hence, by replacing x with $x \odot x^*$, without loss of generality we can assume that $x^* = \vec{1}$. Now, let D_G, D_H and A_G, A_H be the degree and adjacency matrices of G and H, so that $L_G = D_G - A_G$ and $L_H = D_H - A_H$. We thus have that

$$\begin{aligned} 2\phi(x) &= x^\top A_G x - 2x^\top A_H x = x^\top (D_G - 2D_H)x - x^\top (L_G - 2L_H)x \\ &= 2(|E_G| - 2|E_H|) - x^\top (L_G - 2L_H)x \ . \end{aligned}$$

By assumption, if $x \in \{-1,1\}^n$ and $x \neq \vec{1}, -\vec{1}$, then we have that $x^\top (L_G - 2L_H)x > 0$, which implies that $\phi(x) < \phi(\vec{1})$, and finishes the proof. □

Fact 7.1.2 shows that if we can argue that $x^\top L_H x < \frac{1}{2} x^\top L_G x$ for every $x \in \{-1,1\}^n \setminus \{\vec{1}, -\vec{1}\}$, then at least information-theoretically we can uniquely determine x^*. Observe that if we view x as the signed indicator vector of a subset $S \subseteq [n]$, then $x^\top L_G x = E_G(S, \bar{S})$, the number of edges in G crossing the cut defined by S, and similarly for $x^\top L_H x$. So, one can view the condition in Fact 7.1.2 as saying that the subgraph H needs to be a (one-sided) cut sparsifier of G, i.e., it needs to roughly preserve the size of all cuts in G. The following relative cut approximation result of Karger [Kar94] shows that this will hold with high probability when H is a randomly chosen subset of G, provided that the minimum cut in G is not too small.

Lemma 7.1.3 (Relative cut approximation [Kar94])**.** *Let $\eta \in (0,1)$. Suppose an n-vertex graph G has min-cut $c_{\min} \geq \frac{12 \log n}{\eta}$, and suppose H is a subgraph of G by selecting each edge with probability η. Then, with probability $1 - o(1)$,*

$$(1-\delta) x^\top L_G x \leq \frac{1}{\eta} \cdot x^\top L_H x \leq (1+\delta) x^\top L_G x \,, \quad \text{for all } x \in \{-1,1\}^n$$

for $\delta = \sqrt{\frac{12 \log n}{\eta c_{\min}}}$.

With Lemma 7.1.3 and Fact 7.1.2 in hand, we now have at least an information-theoretic algorithm with the same guarantees as in Theorem 5. We follow the strategy highlighted in Strategy 1. To decompose the graph G, we recursively find a min cut and split if it is below the threshold in Lemma 7.1.3. Notice that this discards at most $O(n \log n) = o(m)$ constraints (for $m \gg n \log n$), and these are precisely the constraints that we "give up" on and do not determine which ones are corrupted. Then, with high probability the local optimal assignment is consistent with x^*, and so locally we have learned *exactly* which constraints are corrupted. Hence, we have produced two sets of constraints: E_1, the $o(1)$-fraction of edges discarded during the decomposition, and $E_2 = (G \setminus E_1) \cap \mathcal{E}_\phi$, which is exactly the set of corrupted constraints after discarding E_1. We note that it is a priori not obvious that this is achievable even for an *exponential-time* algorithm, as even though the 2^n-time brute force algorithm will find the best assignment x to ϕ, it may not necessarily be x^*, and so the set of constraints violated by the globally optimal assignment might not be $\mathcal{E}_\phi$.

7.1.4 Efficient exact recovery from relative spectral approximation

Information-theoretic uniqueness implies that the planted assignment x^* is the unique optimal assignment. But can we efficiently recover x^*? One natural approach is to simply solve the basic SDP relaxation of ϕ: for $X \in \mathbb{R}^{n \times n}$, maximize $\phi(X) := \sum_{(i,j) \in G} X_{ij} b_{ij}$ subject to $X \succeq 0$, $X = X^\top$, and $\mathrm{diag}(X) = \mathbb{I}$. If the optimal SDP solution is simply $X = x^* x^{*\top}$, then we trivially recover x^* from the SDP solution. We thus ask: does the min cut condition of Fact 7.1.2 and Lemma 7.1.3 imply that $x^* x^{*\top}$ is the unique optimal solution to the SDP? Namely, is the min cut condition sufficient for the SDP to certify that x^* is the unique optimal assignment?

Unfortunately, it turns out that this is not the case, and we give a counterexample in Section 7.5. We thus require a stronger condition than the min cut one in order to obtain efficient algorithms. Nonetheless, an analogue of Fact 7.1.2 continues to hold, although now we require a stronger version that holds for all SDP solutions X, not just $x \in \{-1, 1\}^n$. This stronger statement shows the SDP can *certify* that x^* is the unique optimal assignment if and only if a certain relative spectral approximation guarantee holds for the corrupted edges.

Lemma 7.1.4 (SDP-certified uniqueness from relative spectral approximation)**.** *Let $G = (V, E_G)$ be an n-vertex connected graph, and let $H = (V, E_H)$ be a subgraph of G where $E_H \subseteq E_G$. Let L_G, L_H be the unnormalized Laplacians of G and H. Consider a noisy planted 2-XOR instance ϕ on G with planted assignment $x^* \in \{-1, 1\}^n$ (Definition 4.2.2), and suppose E_H is the set of corrupted edges.*

The SDP relaxation of ϕ satisfies

$$\max_{X \succeq 0,\ X = X^\top,\ \mathrm{diag}(X) = \mathbb{I}} \phi(X) = \phi(x^*) = |E_G| - 2|E_H|\,,$$

where $X = x^* x^{*\top}$ is the unique *optimum if and only if G and H satisfy*

$$\langle X, L_H \rangle < \frac{1}{2} \langle X, L_G \rangle\,, \quad \forall X \succeq 0,\ X = X^\top,\ \mathrm{diag}(X) = \mathbb{I},\ X \neq \vec{1}\vec{1}^\top\,.$$

Proof. Recall that each $e = \{i, j\} \in E$ corresponds to a constraint $x_i x_j = b_e$ where $b_e = x^*_i x^*_j$ if $e \in E_G \setminus E_H$ and $b_e = -x^*_i x^*_j$ if $e \in E_H$, meaning that $\phi(X) = \sum_{\{i,j\} \in G \setminus E} X_{ij} x^*_i x^*_j - \sum_{\{i,j\} \in E} X_{ij} x^*_i x^*_j$. Without loss of generality, we can assume that $x^* = \vec{1}$ and that $\phi(X) = \frac{1}{2}\langle X, A_G - 2A_H \rangle$, where A_G, A_H are the adjacency matrices of G and H.

Note that $L_G = D_G - A_G$ and $L_H = D_H - A_H$, and $\mathrm{tr}(D_G) = 2|E_G|$, $\mathrm{tr}(D_H) = 2|E_H|$. For any $X \succeq 0$ with $\mathrm{diag}(X) = \mathbb{I}$,

$$\langle X, A_G - 2A_H \rangle = \langle X, (D_G - L_G) - 2(D_H - L_H) \rangle = 2(|E_G| - 2|E_H|) + \langle X, 2L_H - L_G \rangle \ .$$

Suppose $\langle X, L_H \rangle < \frac{1}{2}\langle X, L_G \rangle$ for all $X \neq \vec{1}\vec{1}^\top$. Since $\langle \vec{1}\vec{1}^\top, L_G \rangle = \langle \vec{1}\vec{1}^\top, L_H \rangle = 0$, we have that the maximum of $\frac{1}{2}\langle X, A_G - 2A_H \rangle$ is $|E_G| - 2|E_H|$ and $X = \vec{1}\vec{1}^\top$ is the unique maximum.

For the other direction, suppose there is an $X \neq \vec{1}\vec{1}^\top$ such that $\langle X, L_H \rangle \geq \frac{1}{2}\langle X, L_G \rangle$. Then, $\phi(X) \geq |E_G| - 2|E_H| = \phi(\vec{1}\vec{1}^\top)$, meaning that $\vec{1}\vec{1}^\top$ is not the unique optimum. □

Relative spectral approximation from uniform subsamples. We now come to a key technical observation. Suppose that H is a *spectral sparsifier* of G, so that $v^\top(\frac{1}{\eta}L_H)v$ is $(1 \pm \delta)v^\top L_G v$ for any $v \in \mathbb{R}^n$. Then clearly $\langle X, L_H \rangle < \frac{1}{2}\langle X, L_G \rangle$ if $\eta < 1/2$ and $\delta = o(1)$, as we can write $X = \sum_{i=1}^n \lambda_i v_i v_i^\top$, and

$$\langle X, L_H \rangle = \sum_{i=1}^n \lambda_i v_i^\top L_H v_i \leq \eta(1+\delta) \sum_{i=1}^n \lambda_i v_i^\top L_G v_i = \eta(1+\delta) \cdot \langle X, L_G \rangle < \frac{1}{2}\langle X, L_G \rangle \, .$$

Furthermore, note that above we only required that $L_H \preceq \eta(1+\delta)L_G$, i.e., we only use the upper part of the spectral approximation.

We are now ready to state the key relative spectral approximation lemma. We observe that when H is a uniformly random subsample of G and G has a *spectral gap* and minimum degree polylog(n), then with high probability $L_H \preceq \eta(1+\delta)L_G$. We note that, while we do not provide a formal proof, the same argument using the lower tail of Matrix Chernoff can also establish a lower bound on L_H, which proves that H is indeed a spectral sparsifier of G.

Lemma 7.1.5 (Relative spectral approximation from uniform subsamples)**.** *Let $\eta \in (0,1)$. Suppose $G = (V, E)$ is an n-vertex graph with minimum degree $d_{\min}$ (self-loops allowed) and spectral gap $\lambda_2(\widetilde{L}_G) = \lambda$ such that $d_{\min}\lambda > \frac{18}{\eta}\log n$, where $\widetilde{L}_G \coloneqq D_G^{-1/2} L_G D_G^{-1/2}$ is the* normalized *Laplacian. Let H be a subgraph of G obtained by selecting each edge with probability η. Then, with probability at least $1 - O(n^{-2})$,*

$$L_H \preceq \eta(1+\delta) \cdot L_G$$

for $\delta = \sqrt{\frac{18 \log n}{\eta d_{\min}\lambda}}$.

Proof. First, note that $\vec{1}$ lies in the kernel of both L_G and L_H, and because of the spectral gap of G, $\dim(\ker(L_G)) = 1$. Therefore, recalling that $L_G = D_G^{1/2}\widetilde{L}_G D_G^{1/2}$, it suffices to prove that

$$\left\| (\widetilde{L}_G^\dagger)^{1/2} D_G^{-1/2} L_H D_G^{-1/2} (\widetilde{L}_G^\dagger)^{1/2} \right\|_2 \leq \eta(1+\delta) \, .$$

Here $\widetilde{L}_G^\dagger$ is the pseudo-inverse of $\widetilde{L}_G$, and $\|\widetilde{L}_G^\dagger\|_2 \leq 1/\lambda$ because G has spectral gap λ. We will write $X \coloneqq (\widetilde{L}_G^\dagger)^{1/2} D_G^{-1/2} L_H D_G^{-1/2} (\widetilde{L}_G^\dagger)^{1/2}$ for convenience.

Note that $L_G = \sum_{e \in E} L_e$, where $L_e \succeq 0$ is the Laplacian of a single edge e and $\|L_e\|_2 = 2$. Let $X_e = (\widetilde{L}_G^\dagger)^{1/2} D_G^{-1/2} L_e D_G^{-1/2} (\widetilde{L}_G^\dagger)^{1/2}$ if e is chosen in H and 0 otherwise. Then, $X = \sum_{e \in E} X_e$ and

$\|\mathbb{E}[X]\|_2 = \eta$. Moreover, each X_e satisfies $X_e \succeq 0$ and $\|X_e\|_2 \le \|\widetilde{L}_G^{\dagger}\|_2 \cdot \|D_G^{-1}\|_2 \cdot \|L_e\|_2 \le \frac{2}{d_{\min}\lambda}$. Thus, by Matrix Chernoff (Fact 3.4.5),

$$\Pr\{\|X\|_2 \ge \eta(1+\delta)\} \le n \cdot \exp\left(-\frac{\delta^2\eta}{3} \cdot \frac{d_{\min}\lambda}{2}\right) \le O(n^{-2})$$

as long as $\frac{18\log n}{\eta d_{\min}\lambda} \le \delta^2 \le 1$. □

Finishing the algorithm. By Lemmas 7.1.4 and 7.1.5, we can thus recover x^* exactly if the constraint graph G of ϕ has a nontrivial spectral gap and minimum degree $d_{\min} \ge \text{polylog}(n)$. To finish the implementation of Strategy 1, we thus need to explain how to algorithmically decompose any graph G into subgraphs $G_1, \dots, G_T$, each with reasonable min degree and nontrivial spectral gap, while only discarding a $o(1)$-fraction of the edges in G. This is the well-studied task of expander decomposition, for which we appeal to known results [KVV04, ST11, Wul17, SW19].

This completes the high-level description of the algorithm in the even k case. Below, we summarize the steps of the final algorithm.

Algorithm 7.1.6 (Algorithm for k-XOR for even k)**.**
Input: k-XOR instance ψ on n variables with m constraints and constraint hypergraph H.
Output: Disjoint sets of constraints $\mathcal{A}_1, \mathcal{A}_2 \subseteq H$ such that $|\mathcal{A}_1| \le o(m)$ and only depends on H, and $\mathcal{A}_2 = (H \setminus \mathcal{A}_1) \cap \mathcal{E}_\psi$.
Operation:

1. Construct the 2-XOR instance ϕ with constraint graph G, as described in Definition 7.1.1.
2. Remove small-degree vertices and run expander decomposition on G to produce expanders $G_1, \dots, G_T$. Set $\mathcal{A}_1$ to be the set of discarded constraints of size $o(m)$.
3. For each $i \in [T]$, solve the basic SDP on the subinstance ϕ_i defined by the constraints G_i. Let $\mathcal{A}_2^{(i)}$ denote the set of constraints violated by the optimal local SDP solution.
4. Output $\mathcal{A}_1$ and $\mathcal{A}_2 = \bigcup_{i=1}^{T} \mathcal{A}_2^{(i)}$.

7.1.5 The case of odd k

We are now ready to briefly explain the differences in the case when k is odd. For the purposes of this overview, we will focus only on the case of $k = 3$. Recall that we are given a 3-XOR instance ψ, specified by a 3-uniform hypergraph $H \subseteq \binom{[n]}{3}$, as well as the right-hand sides $b_C \in \{-1, 1\}$ for $C \in H$, where $b_C = x^*_C$ with probability $1 - \eta$ and $b_C = -x^*_C$ otherwise and $x^* \in \{-1, 1\}^n$ is the planted assignment.

We now produce a 4-XOR instance using the well-known "Cauchy-Schwarz trick" from CSP refutation [CGL04]. The general idea is to, for any pair of clauses (C, C') that intersect, add the "derived constraint" $x_C x_{C'} = b_C b_{C'}$ to the 4-XOR instance. Notice that if, e.g., $C = \{u, i, j\}$ and $C' = \{u, i', j'\}$, then x_u appears twice on the left-hand side, and thus the constraint is $x_i x_j x_{i'} x_{j'} = b_C b_{C'}$. Given this 4-XOR, we produce a 2-XOR following a similar strategy as in Definition 7.1.1. The above description omits many technical details, which we handle in

Sections 7.3 and 7.4; we remark here that these are the same issues that arise in the CSP refutation case, and we handle them using the techniques in [GKM22].

We have thus produced a 2-XOR instance ϕ that is noisy but not in the sense of Definition 4.2.2. Indeed, each edge e in ϕ is "labeled" by a pair (C, C') of constraints in ψ, and e is noisy if and only if *exactly* one of (C, C') is, and so the noise is not independent across constraints. Nonetheless, we can still follow the general strategy as in Algorithm 7.1.6. The main technical challenge is to argue that the relative spectral approximation guarantee of Lemma 7.1.5 holds even when the noise has the aforementioned correlations, and we do this in Lemma 7.4.7. This allows us to recover, for most intersecting pairs (C, C'), the quantity $\xi(C)\xi(C')$, where $\xi(C) = -1$ if C is corrupted, and is 1 otherwise, i.e., $b_C = x^*_C \xi(C)$; we do not determine $\xi(C)\xi(C')$ if and only if the pair (C, C') corresponds to an edge e that was discarded during the expander decomposition.

However, we are not quite done, as we would like to recover $\xi(C)$ for most C, but we only know $\xi(C)\xi(C')$ for most intersecting pairs (C, C'). Let us proceed by assuming that we know $\xi(C)\xi(C')$ for all intersecting pairs (C, C'), and then we will explain how to do a similar decoding process when we only know most pairs. Let us fix a vertex u, and let H_u denote the set of $C \in H$ containing u. Now, we know $\xi(C)\xi(C')$ for all $C, C' \in H_u$, and so by Gaussian elimination we can determine $\xi(C)$ for all $C \in H_u$ up to a global sign. Now, we know that the vector $\{\xi(C)\}_{C \in H_u}$ should have roughly $\eta|H_u|$ entries that are -1. So, choosing the global sign that results in fewer -1's, we thus correctly determine $\xi(C)$ for all $C \in H_u$. We can then repeat this process for each choice of u to decode $\xi(C)$ for all C.

Of course, we only actually know $\xi(C)\xi(C')$ for most intersecting pairs (C, C'). This implies that for most choices of u, the graph G_u with vertices H_u and edges (C, C') if we know $\xi(C)\xi(C')$ is obtained from the complete graph on vertices H_u and deleting some $o(1)$-fraction of edges. This implies that G_u has a connected component of size $(1 - o(1))|H_u|$, and again via Gaussian elimination and picking the proper global sign, we can determine $\xi(C)$ on this large connected component. By repeating this process for each choice of u, we thus recover $\xi(C)$ for most u.

7.2 From planted CSPs to noisy XOR

In this section, we show how to use Theorem 5 to prove Theorem 4. Before we delve into the formal proof, we will first explain the reduction given in [FPV15]. We begin with some definitions.

Setup. Let Ψ be sampled from $\Psi(\vec{H}, x^*, Q)$, where $x^* \in \{-1, 1\}^n$, $\vec{H} \subseteq [n]^k$, and Q is a planting distribution for the predicate P. Let $Q(y) = \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} y_i$ be the Fourier decomposition of Q, where $\hat{Q}(S) = \frac{1}{2^k} \sum_{y \in \{-1,1\}^k} Q(y) \prod_{i \in S} y_i \in [-2^{-k}, 2^{-k}]$. Recall (Definition 4.2.1) that Ψ is specified by a collection $\vec{H} \subseteq [n]^k$ of scopes, along with a vector $\xi(\vec{C}) \in \{-1, 1\}^k$ for each $\vec{C} \in \vec{H}$ of literal negations.

Definition 7.2.1. Let $S \subseteq [k]$ be nonempty. Let $\psi^{(S,+)}$ be the $|S|$-XOR instance obtained by, for each constraint $\vec{C}$ in Ψ, adding the constraint $\prod_{i \in S} x_{\vec{C}_i} = \prod_{i \in S} \xi(\vec{C})_i$. Similarly, let $\psi^{(S,-)}$ have constraints $\prod_{i \in S} x_{\vec{C}_i} = -\prod_{i \in S} \xi(\vec{C})_i$.

We make use of the following simple claim.

Claim 7.2.2. For each nonempty $S \subseteq [k]$, $\psi^{(S,+)}$ is a noisy $|S|$-XOR instance (Definition 4.2.2) with planted assignment x^* and noise $\eta = \frac{1}{2}(1 - 2^k \hat{Q}(S))$. Similarly, $\psi^{(S,-)}$ is a noisy $|S|$-XOR instance with planted assignment x^* and noise $\eta = \frac{1}{2}(1 + 2^k \hat{Q}(S))$.

Proof. For each $\vec{C}$, the literal negation $\xi(\vec{C})$ is sampled such that $\Pr[\xi(\vec{C}) = \xi] = Q(\xi \odot x^*_{\vec{C}})$, where $\odot$ denotes the element-wise product. This is equivalent to sampling $y \leftarrow Q$ and setting $\xi(\vec{C}) = y \odot x^*_{\vec{C}}$. It thus follows that the probability that the constraint $\vec{C}$ produces a corrupted constraint in $\psi^{(S,+)}$ is

$$\Pr_{y \leftarrow Q}\left\{\prod_{i \in S} y_i = -1\right\} = \frac{1}{2}\left(1 - \mathbb{E}_{y \leftarrow Q}\left\{\prod_{i \in S} y_i\right\}\right) = \frac{1}{2}(1 - 2^k \hat{Q}(S)) \ ,$$

and is independent for each $\vec{C}$. A similar calculation handles the case of $\psi^{(S,-)}$. □

With the above observations in hand, we can now easily describe the reduction in [FPV15]. First, their reduction requires the algorithm to have a description of the distribution Q. Given Q, the algorithm then finds the smallest S such that $\hat{Q}(S)$ is nonzero. Since they know the exact value of $\hat{Q}(S)$, they can determine its sign correctly. Suppose that $\hat{Q}(S) > 0$ (the other case is similar). Then, by solving the $|S|$-XOR instance $\psi^{(S,+)}$, they recover the planted assignment of $\psi^{(S,+)}$ *exactly*.[2] But this planted assignment is precisely x^*, and so they have also succeeded in recovering the planted assignment of ψ.

The aforementioned reduction clearly does not generalize to the semirandom setting, as in general the subinstances $\psi^{(S,\pm)}$ will not uniquely determine x^*. Furthermore, their reduction additionally requires knowing Q, and while it is not too unreasonable to assume this for random planted CSPs (as it is perhaps natural for the algorithm to know the distribution), in the semirandom setting this assumption is a bit strange because we want to view semirandom CSPs as "moving towards" worst-case ones.

We now prove Theorem 4 from Theorem 5.

Proof of Theorem 4 from Theorem 5. We will present the proof in three steps. First, like [FPV15], we will assume that the algorithm is given a description of Q and we will assume that each $|\hat{Q}(S)|$ is either 0 or at least $2^{-k}\varepsilon > 0$.[3] Then, we will remove this assumption provided that $Q(y) > 2\varepsilon$ for all y with $Q(y) > 0$, i.e., the every y in the support of Q has some minimum probability. Finally, we will remove the last assumption.

Step 1: the proof when we are given Q. For each S where $\hat{Q}(S) \neq 0$, we construct the instance $\psi^{(S,+)}$ (if $\hat{Q}(S) > 0$) or $\psi^{(S,-)}$ (if $\hat{Q}(S) < 0$). We then apply[4] Theorem 5 to each such instance. Note that by Claim 7.2.2, the instance has noise $\eta = \frac{1}{2}(1 - 2^k|\hat{Q}(S)|) \leq \frac{1}{2}(1 - \varepsilon)$ (because we picked the correct sign when choosing between $\psi^{(S,+)}$ and $\psi^{(S,-)}$, and we assume $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$). Then, since $m \geq c^k n^{k/2} \cdot \frac{\log^3 n}{\varepsilon^9}$ and $|S| \leq k$, by applying Theorem 5 with noise η and parameter $\varepsilon' := 2^{-k}\varepsilon$, we obtain sets $\vec{H}^{(S,1)}$ (the discarded set) and $\vec{H}^{(S,2)}$ (the corrupted constraints) where $|\vec{H}^{(S,1)}| \leq \varepsilon' m$ and $\vec{H}^{(S,2)} = (\vec{H} \setminus \vec{H}^{(S,1)}) \cap \mathcal{E}_{\psi^{(S)}}$. Hence, for every constraint $\vec{C} \in \vec{H} \setminus \vec{H}^{(S,1)}$, it follows that we have learned $\prod_{i \in S} x^*_{\vec{C}_i}$, where x^* is the planted assignment for Ψ. By setting

[2] Here, they also treat $|\hat{Q}(S)|$ as constant, as if $|\hat{Q}(S)| \ll 1/n$, say, then their algorithm would not succeed in recovering the planted assignment on the XOR instance.

[3] This assumption is implicit in [FPV15]; see the previous footnote.

[4] Note that Theorem 5 only applies when $|S| \geq 2$. When $|S| = 1$, there is a trivial algorithm; see Section 7.7 for details.

$\vec{H}' := \vec{H} \setminus \cup_{S:\hat{Q}(S)\neq 0} \vec{H}^{(S,1)}$, it follows that we know $\prod_{i\in S} x^*_{\vec{C}_i}$ for all $\vec{C} \in \vec{H}'$ and S with $\hat{Q}(S) \neq 0$, where $|\vec{H}'| \geq (1-2^k\varepsilon')m = (1-\varepsilon)m$.

We now solve the system of linear equations given by $\prod_{i\in S} x^*_{\vec{C}_i}$ for all $\vec{C} \in \vec{H}'$ and S with $\hat{Q}(S) \neq 0$ to obtain some assignment $x \in \{-1,1\}^n$. As x^* is a valid solution to these equations, such an x exists, although it may not be x^*.

The final step is to argue that for every $\vec{C} \in \vec{H}'$, x satisfies the constraint $\vec{C}$, namely that $P(\xi(\vec{C})_1 x_{\vec{C}_1}, \xi(\vec{C})_2 x_{\vec{C}_2}, \ldots, \xi(\vec{C})_k x_{\vec{C}_k}) = 1$. Indeed, if this is true then we are done, as x satisfies at least $(1-\varepsilon)m$ constraints in Ψ, and so we have obtained the desired assignment.

Let $\vec{C} \in \vec{H}'$. We know that for every S with $\hat{Q}(S) \neq 0$, we have that $\prod_{i\in S} x_{\vec{C}_i} = \prod_{i\in S} x^*_{\vec{C}_i}$. Hence, it follows that

$$Q(\xi(\vec{C}) \odot x) = \sum_{S\subseteq[k]} \hat{Q}(S) \prod_{i\in S} \xi(\vec{C})_i x_{\vec{C}_i} = \sum_{S\subseteq[k]} \hat{Q}(S) \prod_{i\in S} \xi(\vec{C})_i x^*_{\vec{C}_i} = Q(\xi(\vec{C}) \odot x^*) > 0\,,$$

where the last inequality is because $\xi(\vec{C})$ was sampled from the distribution $Q(\xi(\vec{C}) \odot x^*)$, and so it must be sampled with nonzero probability. As Q is supported only on satisfying assignments to the predicate P, it thus follows that $\xi(\vec{C}) \odot x^*$ must also satisfy P.

Step 2: removing the dependence on Q assuming a lower bound on $Q(y)$. First, we observe that because k is constant, we can, for each S, guess a symbol $\{0,+,-\}$, where 0 denotes, informally, the belief that $|\hat{Q}(S)| < 2^{-k}\varepsilon$, $+$ denotes that $\hat{Q}(S) \geq 2^{-k}\varepsilon$, and $-$ denotes that $\hat{Q}(S) \leq -2^{-k}\varepsilon$. For each of the 3^{2^k} choices of guesses, i.e., functions $f\colon \{S \subseteq [k]\} \to \{0,+,-\}$, we run algorithm mentioned in the previous step. Namely, for each S: (1) if $f(S) = 0$, then we ignore S, (2) if $f(S) = +$, then we run Theorem 5 on $\psi^{(S,+)}$ to obtain $\vec{H}^{(S,1)}$ and $\vec{H}^{(S,2)}$, and (3) if $f(S) = -$, then we run Theorem 5 on $\psi^{(S,+)}$ to obtain $\vec{H}^{(S,1)}$ and $\vec{H}^{(S,2)}$. As before, we solve the system of linear equations to obtain some assignment $x^{(f)} \in \{-1,1\}^n$. By enumerating over all possible choices of f, we obtain a list of at most $3^{2^k} = O(1)$ assignments. We then try all of them and output the best one.

It thus remains to show that at least one of the assignments in the list has high value. As one may expect, this will be the assignment $x^{(f^*)}$, where f^* is the correct label function. Indeed, when $f = f^*$, then we are precisely running the algorithm in Step 1, and as observed, after solving the linear system of equations we obtain an assignment $x := x^{(f^*)}$ with the following property. For every $\vec{C} \in \vec{H}'$ and every S with $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$, we have that $\prod_{i\in S} x_{\vec{C}_i} = \prod_{i\in S} x^*_{\vec{C}_i}$, where $\vec{H}' \subseteq \vec{H}$ has size $\geq (1-\varepsilon)m$.

Finally, we show that for every $\vec{C} \in \vec{H}'$, x satisfies the constraint $\vec{C}$. Namely, we have $P(\xi(\vec{C})_1 x_{\vec{C}_1}, \xi(\vec{C})_2 x_{\vec{C}_2}, \ldots, \xi(\vec{C})_k x_{\vec{C}_k}) = 1$. Let $\vec{C} \in \vec{H}'$. We know that for every S with $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$, we have that $\prod_{i\in S} x_{\vec{C}_i} = \prod_{i\in S} x^*_{\vec{C}_i}$. Hence, it follows that

$$\begin{aligned}\left|Q(\xi(\vec{C}) \odot x) - Q(\xi(\vec{C}) \odot x^*)\right| &= \left|\sum_{S\subseteq[k]} \hat{Q}(S) \prod_{i\in S} \xi(\vec{C})_i x_{\vec{C}_i} - \sum_{S\subseteq[k]} \hat{Q}(S) \prod_{i\in S} \xi(\vec{C})_i x^*_{\vec{C}_i}\right| \\ &= \left|\sum_{S\subseteq[k]:|\hat{Q}(S)|<2^{-k}\varepsilon} \hat{Q}(S) \left(\prod_{i\in S} \xi(\vec{C})_i x_{\vec{C}_i} - \prod_{i\in S} \xi(\vec{C})_i x^*_{\vec{C}_i}\right)\right| \leq 2^k \cdot 2^{-k+1}\varepsilon\,.\end{aligned}$$

Now, if we assume that $Q(y) > 2\varepsilon$ for every $y \in \{-1,1\}^k$ with $Q(y) > 0$, then it follows that $Q(\xi(\vec{C}) \odot x) > 0$, and so x satisfies the constraint $P(\xi(\vec{C})_1 x_{\vec{C}_1}, \xi(\vec{C})_2 x_{\vec{C}_2}, \ldots, \xi(\vec{C})_k x_{\vec{C}_k}) = 1$.

Step 3: removing the lower bound on $Q(y)$. In Step 2, we assumed that $Q(y) > 2\varepsilon$ for all $y \in \{-1,1\}^k$ with $Q(y) > 0$. However, we only used this fact in the final step, when we argue that $Q(\xi(\vec{C}) \odot x) > 0$ by observing that $Q(\xi(\vec{C}) \odot x) \geq Q(\xi(\vec{C}) \odot x^*) - 2\varepsilon > 0$. To remove the assumption, we will show that for at most $2^{k+2}\varepsilon$ constraints $\vec{C} \in \vec{H}$, it holds that $Q(\xi(\vec{C}) \odot x^*) \leq 2\varepsilon$. This then implies that x satisfies at least $(1 - \varepsilon - 2^{k+2}\varepsilon)m = (1 - O(\varepsilon))m$ constraints, which finishes the proof.

Let $\mathcal{S}$ denote the set of $\vec{C} \in \vec{H}$ where $Q(\xi(\vec{C}) \odot x^*) \leq 2\varepsilon$. Observe that the probability, over the choice of $\xi(\vec{C})$, that $\vec{C} \in \mathcal{S}$ is at most $2^k \cdot 2\varepsilon = 2^{k+1}\varepsilon$, and moreover this is independent for each $\vec{C} \in \vec{H}$. Thus, by a Chernoff bound, it follows that with probability $\geq 1 - \exp(-O(\varepsilon m)) \geq 1 - 1/\mathrm{poly}(n)$, it holds that $|\mathcal{S}| \leq 2 \cdot 2^{k+1}\varepsilon$, and so we are done. □

Remark 7.2.3 (Tolerating fewer constraints for structured Q's). We have shown that the above algorithm succeeds in finding an assignment x that satisfies at least $(1 - O(\varepsilon))m$ constraints when $m \geq n^{k/2} \cdot \mathrm{poly}(\log n, 1/\varepsilon)$. However, if the distribution Q has $|\hat{Q}(S)| < 2^{-k}\varepsilon$ for all S with $|S| > r$, then we only need $n^{r/2} \cdot \mathrm{poly}(\log n, 1/\varepsilon)$ constraints. (If $r = 0$, then for small enough constant ε, Q will be supported on all of $\{-1,1\}^k$, and so any assignment satisfies all constraints. If $r = 1$, we require $O(n \cdot \frac{\log n}{\varepsilon})$ constraints; see Lemma 7.7.1.) Indeed, this follows because for such Q, the true label function f^* will have $f^*(S) = 0$ for any S with $|S| > r$. Hence, for this choice of f^*, we only call Theorem 5 on noisy t-XOR instances for $t \leq r$, and so we have enough constraints. It therefore follows that the assignment $x^{(f^*)}$ that we obtain for the label function f^* will be, with high probability an assignment that satisfies at least $(1 - O(\varepsilon))m$ constraints.

An example where this gives an improvement is the well-studied NAE-3-SAT (not-all-equal-3SAT) predicate [AE98, ACIM01, DSS14]. Suppose Q is the uniform distribution over satisfying assignments to NAE-3-SAT: $Q(x_1, x_2, x_3) = \frac{1}{6} \cdot \frac{1}{4}(3 - x_1x_2 - x_2x_3 - x_1x_3)$. Then, we only need $m \geq \tilde{O}(n)$ constraints, even though it is a 3-CSP ($k = 3$).

7.3 From k-XOR to spread bipartite k-XOR

In this section, we begin the proof of Theorem 5. See Definition 4.2.2 for a reminder of our semirandom planted k-XOR model $\psi(H, x^*, \eta)$ given a k-uniform hypergraph H, assignment $x^* \in \{-1,1\}^n$, and noise parameter $\eta \in (0, 1/2)$. Recall also that $\mathcal{E}_\psi$ denotes the set of corrupted hyperedges.

We think of $\mathcal{A}_1(H)$ as the small set of edges that we discard (or give up on), and this will only depend on the hypergraph H. For the rest of the graph, the algorithm will correctly identify which edges are corrupted.

Our proof of Theorem 5 goes via a reduction to *spread bipartite t-XOR* instances for $t = 2, \ldots, k$, which are t-XOR instances with some additional desired structure. Such instances were introduced in [GKM22] to study the refutation of semirandom k-XOR instances. The reduction here is nearly identical to the corresponding reduction in [GKM22, Section 4].

Definition 7.3.1 (Spread bipartite k-XOR)**.** A p-bipartite k-XOR instance ψ on n variables with m constraints is defined by a collection of $(k-1)$-uniform hypergraphs $H = \{H_u\}_{u \in [p]}$ on the

vertex set $[n]$, as well as "right-hand sides" $b_{u,C}$ for each $u \in [p]$ and $C \in H_u$. There are two sets of variables of ψ: the "normal" variables $x_1, \dots, x_n$, and the "special" variables $y_1, \dots, y_p$. The constraints of ψ are $y_u \prod_{i \in C} x_i = b_{u,C}$ for each $u \in [p]$, $C \in H_u$.

We furthermore say that ψ is τ-*spread* if it has the following additional properties:

(1) $|H_u| = \frac{m}{p} \geq 2\lfloor \frac{1}{2\tau^2} \rfloor$ and $\frac{m}{p}$ is even for each $u \in [p]$,

(2) For each $u \in [p]$ and set $Q \subseteq [n]$, $\deg_u(Q) \leq \frac{1}{\tau^2} \max(1, n^{\frac{k}{2}-1-|Q|})$.

Analogously to Definition 4.2.2, we call ψ a *semirandom planted* instance with planted assignment (x^*, y^*) and noise parameter η if the right-hand sides $b_{u,C}$ are generated by setting $b_{u,C} = y^*_u \prod_{i \in C} x^*_i$ with probability $1 - \eta$ and $b_{u,C} = -y^*_u \prod_{i \in C} x^*_i$ otherwise, independently for each choice of u, C. For a choice of x^*, y^*, $H = \{H_u\}_{u \in [p]}$, and η, we call this distribution $\psi(\{H_u\}_{u \in [p]}, x^*, y^*, \eta)$. As before, if an edge (u, C) has $b_{u,C} = -y^*_u \prod_{i \in C} x^*_i$, we call (u, C) a *corrupted* hyperedge, and we denote the set of corrupted hyperedges in ψ by $\mathcal{E}_\psi$.

The main technical result of this chapter is the following lemma, which gives an algorithm to find the noisy constraints in a semirandom planted τ-spread bipartite k-XOR instance.

Lemma 7.3.2 (Algorithm for τ-spread bipartite k-XOR)**.** *Let $k \geq 2$, $n, p \in \mathbb{N}$, $\varepsilon \in (0,1)$, $\eta \in [0, 1/2)$, and let $\gamma := 1 - 2\eta > 0$. Let $\tau \leq \frac{c\gamma}{\sqrt{k \log n}}$, and let $m \geq C n^{\frac{k-1}{2}} \sqrt{p} \cdot \frac{(k \log n)^{3/2}}{\tau \gamma^2 \varepsilon^{3/2}}$ for some universal constants c, C. There is a polynomial-time algorithm $\mathcal{A}$ that takes as input an τ-spread p-bipartite k-XOR instance ψ with constraint hypergraph $H = \{H_u\}_{u \in [p]}$ and outputs two disjoint sets $\mathcal{A}_1(H), \mathcal{A}_2(\psi) \subseteq H$ with the following guarantee: (1) for any instance ψ with m constraints, $|\mathcal{A}_1(H)| \leq \varepsilon m$ and $\mathcal{A}_1(H)$ only depends on H, and (2) for any $x^* \in \{-1,1\}^n, y^* \in \{-1,1\}^p$ and any $H = \{H_u\}_{u \in [p]}$ with $|H| := \sum_{u \in [p]} |H_u| \geq m$, with probability $1 - \frac{1}{\mathrm{poly}(n)}$ over $\psi \leftarrow \psi(\{H_u\}_{u \in [p]}, x^*, y^*, \eta)$, it holds that $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (H \setminus \mathcal{A}_1(H))$.*

Note that as $\eta \to \frac{1}{2}$, $\gamma = 1 - 2\eta \to 0$ and $\tau \to 0$, which blows up m. This is the expected behavior since when $\eta = \frac{1}{2}$, it is impossible to recover the planted assignment since the signs of the constraints are uniformly random.

7.3.1 Proof of Theorem 5 from Lemma 7.3.2

With Lemma 7.3.2, we can finish the proof of Theorem 5. The high-level idea of this proof is very simple. First, we decompose the k-XOR instance ψ into subinstances $\psi^{(t)}$ for each $t = 2, \dots, k$, using a hypergraph decomposition algorithm very similar to the one used in [GKM22, HKM23]. The algorithm and its guarantees are shown in Section 7.6. Then, we run the algorithm in Lemma 7.3.2 to identify a set of corrupted constraints and a small set of discarded constraints within each subinstance $\psi^{(t)}$. We then take the union of these outputs to be the final output of the algorithm.

Proof of Theorem 5. We begin with the decomposition of ψ into $\psi^{(2)}, \dots, \psi^{(k)}$ along with a set of "discarded" hyperedges $H^{(1)}$, which is done using Algorithm 7.6.1 with spread parameter $\tau := \frac{c(1-2\eta)}{\sqrt{k \log n}}$ where c is the constant in Lemma 7.3.2. For each $t = 2, \dots, k$, $\psi^{(t)}$ is a semirandom (with noise η) planted τ-spread $p^{(t)}$-bipartite t-XOR instance specified by $(t-1)$-uniform hypergraphs $\{H^{(t)}_u\}_{u \in [p^{(t)}]}$.

Let $m^{(t)} := \sum_{u \in [p^{(t)}]} |H^{(t)}_u|$. Algorithm 7.6.1 has the following guarantees:

(1) The runtime is $n^{O(k)}$,

(2) For each $t \in \{2, \dots, k\}$ and $u \in [p^{(t)}]$, $|H_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}} = 2\lfloor \frac{1}{2\tau^2} \max(1, n^{t-\frac{k}{2}-1}) \rfloor$; in particular, $|H_u^{(t)}|$ is even and is at least $2\lfloor \frac{1}{2\tau^2} \rfloor$,

(3) For each $t = 2, \dots, k$, the instance $\psi^{(t)}$ is τ-spread,

(4) The number of "discarded" hyperedges is $m^{(1)} := |H^{(1)}| \le \frac{1}{k\tau^2} n^{\frac{k}{2}}$,

(5) For $t \in \{2, \dots, k\}$, each $C \in H_u^{(t)}$ is obtained by removing $k - (t-1)$ vertices from an edge in the original hypergraph H. Thus, there is a one-to-one map $\mathsf{Decomp}\colon H \to H^{(1)} \cup \bigcup_{t=2}^{k} \{H_u^{(t)}\}_{u \in [p^{(t)}]}$, such that an edge $C \in H$ is corrupted if and only if the edge $\mathsf{Decomp}(C)$ is corrupted in the instance $\psi^{(t)}$ that it lies in.

For convenience, we denote $\gamma := 1 - 2\eta$ and $\beta := 4C \cdot \frac{(k \log n)^{3/2}}{\tau \gamma^2 \varepsilon^{3/2}} = \frac{4C}{c} \cdot \frac{k^2 \log^2 n}{\gamma^3 \varepsilon^{3/2}}$ where C, c are the constants in Lemma 7.3.2. The algorithm in Theorem 5 works as follows. First, it runs Algorithm 7.6.1 to produce the instances $\psi^{(2)}, \dots, \psi^{(k)}$. Then, for each $t = 2, \dots, k$, if $m^{(t)} \ge n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta$, we run Lemma 7.3.2 on $\psi^{(t)}$ and obtain, with probability $1 - 1/\mathrm{poly}(n)$, a set $A_1^{(t)}$ where $|A_1^{(t)}| \le \frac{\varepsilon}{2} m^{(t)}$ and $A_2^{(t)} = \mathcal{E}_{\psi^{(t)}} \setminus A_1^{(t)}$. Otherwise, if $m^{(t)} < n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta$, we set $A_1^{(t)} = H^{(t)}$ and $A_2^{(t)} = \emptyset$. Finally, we output $\mathcal{A}_1 := H^{(1)} \cup \bigcup_{t=2}^{k} \mathsf{Decomp}^{-1}(A_1^{(t)})$ and $\mathcal{A}_2 := \bigcup_{t=2}^{k} \mathsf{Decomp}^{-1}(A_2^{(t)})$, where Decomp is the mapping in property (5) of Algorithm 7.6.1.

Note that $m^{(t)} = p^{(t)} |H_u^{(t)}| \ge p^{(t)} \cdot \frac{1}{2\tau^2} n^{t-\frac{k}{2}-1}$, which means $p^{(t)} \le 2\tau^2 n^{\frac{k}{2}-t+1} m^{(t)}$, and since $\sum_t \sqrt{m^{(t)}} \le \sqrt{k \sum_t m^{(t)}} \le \sqrt{km}$ by Cauchy-Schwarz, we have

$$\sum_{t=2}^{k} n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta \le O(\tau) \cdot n^{\frac{k}{4}} \sqrt{km} \cdot \beta \le o(\varepsilon) m$$

as long as $m \gg n^{\frac{k}{2}} \cdot k\tau^2 \beta^2 / \varepsilon^2$. Moreover, $m^{(1)} \le \frac{1}{k\tau^2} n^{\frac{k}{2}} = \frac{\log n}{c^2 \gamma^2} n^{\frac{k}{2}} \le o(\varepsilon) m$. One can verify, by plugging in β, that the lower bound on m in Theorem 5 suffices.

By union bound over t, it thus follows that

$$|\mathcal{A}_1| \le m^{(1)} + \sum_{t=2}^{k} \frac{\varepsilon}{2} m^{(t)} + \sum_{t=2}^{k} n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \beta \le \varepsilon m \,,$$

and $\mathcal{A}_2 = \mathcal{E}_\psi \setminus \mathcal{A}_1$. Moreover, by Lemma 7.3.2, $\mathcal{A}_1$ only depends on the hypergraph H. This completes the proof. □

7.4 Identifying noisy constraints in spread bipartite k-XOR

In this section, we prove Lemma 7.3.2. The proof will be decomposed into the following steps. First, we take the semirandom planted bipartite k-XOR instance ψ and transform it into a 2-XOR instance ϕ. Second, we decompose the constraint graph of ϕ into expanders. For each expander in the decomposition, we argue that the SDP solution to this subinstance is rank 1, and moreover agrees *exactly* with the planted assignment. This allows us to identify, for each expanding subinstance, *exactly* which edges in ϕ are errors. Finally, we use this information to identify the set of corrupted constraints in the original instance ψ, which finishes the proof.

7.4.1 Setup and key notation

We now introduce the key notation that shall be used throughout this section. Let ψ be the semirandom τ-spread p-bipartite k-XOR instance (recall Definition 7.3.1) with m constraints given as the input to the algorithm. Recall that the instance ψ is specified by a collection of p hypergraphs $\{H_u\}_{u\in[p]}$, where each H_u is a $(k-1)$-uniform hypergraph on n vertices and $|H_u| = m/p$. Each constraint in ψ is specified by a pair (u, C) where $u \in [p]$, $C \in H_u$, and has a right-hand side $b_{u,C} \in \{-1, 1\}$, and the constraints are $y_u \prod_{i\in C} x_i = b_{u,C}$, where $\{y_u\}_{u\in[p]}$ and $\{x_i\}_{i\in[n]}$ are variables. Because the instance ψ is semirandom with noise parameter η and planted assignment (x^*, y^*), for each constraint (u, C) we have, with probability $1-\eta$ independently, $b_{u,C} = y^*_u \prod_{i\in C} x^*_i$, and otherwise $b_{u,C} = -y^*_u \prod_{i\in C} x^*_i$. Our goal is to output, in $n^{O(k)}$-time, a set $\mathcal{A}_1(H)$ of size $\leq \tau m$ to discard, and then for the rest of the instance, identify exactly the corrupted constraints, i.e., those for which $b_{u,C} = -y^*_u \prod_{i\in C} x^*_i$.

We now define the 2-XOR instance ϕ from ψ. An example is shown in Fig. 7.2.

Definition 7.4.1 (2-XOR instance ϕ from bipartite k-XOR ψ)**.** For every $u \in [p]$ and H_u, we partition H_u arbitrarily into two sets $H^{(L)}_u$ and $H^{(R)}_u$ of equal size.

- If k is odd, then there are $\binom{n}{\frac{k-1}{2}}^2$ variables in ϕ, one variable $z_{(S_1,S_2)}$ for each pair of sets $S_1, S_2 \subseteq [n]$ where $|S_1| = |S_2| = \frac{k-1}{2}$.
- If k is even, then there are $2\binom{n}{\lceil\frac{k-1}{2}\rceil}\binom{n}{\lfloor\frac{k-1}{2}\rfloor}$ variables in ϕ, one variable $z_{(S_1,S_2)}$ for each pair of sets $S_1, S_2 \subseteq [n]$ where either $|S_1| = \lceil\frac{k-1}{2}\rceil$ and $|S_2| = \lfloor\frac{k-1}{2}\rfloor$ or $|S_1| = \lfloor\frac{k-1}{2}\rfloor$ and $|S_2| = \lceil\frac{k-1}{2}\rceil$.

For each $u \in [p]$, $C \in H^{(L)}_u$ and $C' \in H^{(R)}_u$, we arbitrarily partition C into sets $S_1 \cup S_2$ and C' into sets $S'_1 \cup S'_2$, where $|S_1| = |S'_1| = \lceil\frac{k-1}{2}\rceil$ and $|S_2| = |S'_2| = \lfloor\frac{k-1}{2}\rfloor$. We then add the constraint $z_{(S_1,S'_2)} z_{(S_2,S'_1)} = b_{u,C} b_{u,C'}$ to ϕ.

It is intuitive to think of clauses from $H^{(L)}_u$ and $H^{(R)}_u$ as having different colors, and each variable $z_{(S_1,S'_2)}$ contains roughly $k/2$ of each color. See Fig. 7.2 for an example of a 2-XOR ϕ constructed from a bipartite k-XOR ψ.

Observation 7.4.2 (Size of ϕ). The number of variables in ϕ is at most n^{k-1} (for both even and odd k). Since each $|H_u| = m/p$, $|H^{(L)}_u| = |H^{(R)}_u| = \frac{m}{2p}$, and the number of constraints in ϕ is exactly $p \cdot (\frac{m}{2p})^2 = \frac{m^2}{4p}$. In particular, when $m \geq n^{\frac{k-1}{2}}\sqrt{p} \cdot \beta$ for $\beta = \mathrm{poly}(\log n)$ as assumed in Lemma 7.3.2, the average degree of ϕ is at least $\frac{1}{4}\beta^2$.

Remark 7.4.3 (Corrupted constraints in ϕ). A constraint $z_{(S_1,S'_2)} z_{(S_2,S'_1)} = b_{u,C} b_{u,C'}$ in ϕ is *corrupted* if exactly one of $b_{u,C}$ and $b_{u,C'}$ is corrupted in ψ. Thus, if each constraint in ψ is corrupted with probability $\eta \in (0, 1/2)$, then each constraint in ϕ is corrupted with probability $2\eta(1-\eta) < 1/2$. Note, however, that the constraints in ϕ are not corrupted independently.

We need some more definitions about the constraint graph of ϕ.

Definition 7.4.4 (Constraint graph of ϕ)**.** Let $G(\phi) = (V, E)$ be the constraint graph of ϕ. Notice that each edge $e \in E$ uniquely identifies $u(e) \in [p]$ and $C_L(e) \in H^{(L)}_{u(e)}$, $C_R(e) \in H^{(R)}_{u(e)}$. For each $u \in [p]$, $C \in H^{(L)}_u$, define $G^{(L)}_{u,C}(\phi)$ to be the subgraph of G that C participates in, i.e., with edge set $\{e \in E : u(e) = u,\ C_L(e) = C\}$. We similarly define $G^{(R)}_{u,C'}(\phi)$ for $C' \in H^{(R)}_u$.

We next make the important observation that the degree of a vertex in $G^{(L)}_{u,C}(\phi)$ is upper bounded by the number of $C' \in H^{(R)}_u$ sharing at least $\lfloor\frac{k-1}{2}\rfloor$ vertices. See Fig. 7.2 also for an

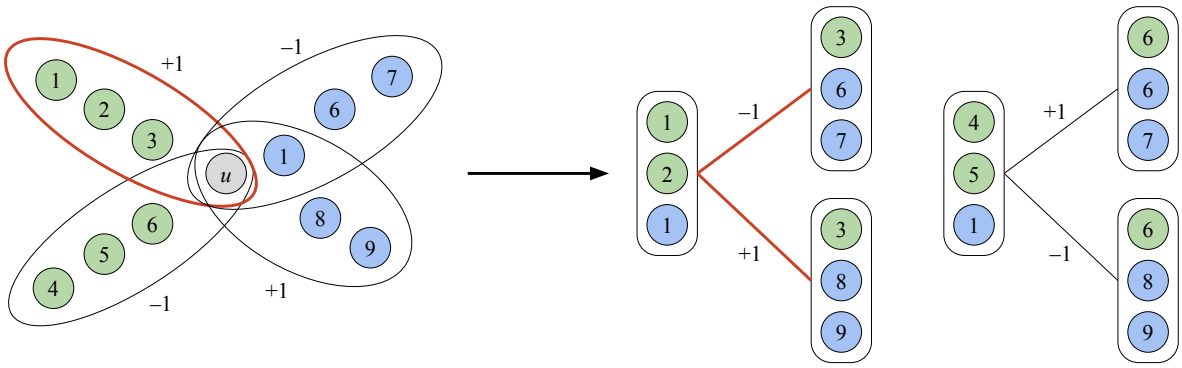

Figure 7.2: An example of the 2-XOR instance ϕ from a bipartite 4-XOR ψ (Definition 7.4.1). On the left, $H_u^{(L)}$ consists of $C_1 = \{1,2,3\}$ and $C_2 = \{4,5,6\}$ (with green vertices), and $H_u^{(R)}$ consists of $C_1' = \{1,6,7\}$ and $C_2' = \{1,8,9\}$ (with blue vertices). On the right, the constraint graph $G(\phi)$ has vertices z_{S_1,S_2} where either $|S_1| = 2$, $|S_2| = 1$ or $|S_1| = 1$, $|S_2| = 2$ (we can view S_1, S_2 as having green, blue vertices). Each edge corresponds to two clauses in ψ; for example, the edge $\{z_{\{1,2\},\{1\}}, z_{\{3\},\{6,7\}}\}$ comes from the clauses C_1 and C_1'.
Corruptions. In the figure, we label a clause -1 if it is corrupted and $+1$ otherwise. An edge in G is corrupted if exactly one of the two corresponding clauses in ψ is corrupted.
Degree of $G_{u,C}^{(L)}(\phi)$. For $C_1 \in H_u^{(L)}$, the subgraph $G_{u,C_1}^{(L)}(\phi)$ corresponds to the edges colored red, i.e., all edges that C_1 participates in. The vertex $z_{\{1,2\},\{1\}}$ has degree 2 in $G_{u,C_1}^{(L)}(\phi)$ because $|C_1' \cap C_2'| = 1$.

illustration. Therefore, assuming that ψ is τ-spread, we have a maximum degree bound on $G_{u,C}^{(L)}(\phi)$ and $G_{u,C'}^{(R)}(\phi)$ for all $u \in [p]$, $C \in H_u^{(L)}$ and $C' \in H_u^{(R)}$.

Lemma 7.4.5 (Degree bounds for $G_{u,C}^{(L)}$, $G_{u,C'}^{(R)}$)**.** *Let ψ be an τ-spread p-bipartite k-XOR instance. Then, for any $u \in [p]$, $C \in H_u^{(L)}$ and $C' \in H_u^{(R)}$, the maximum degree of $G_{u,C}^{(L)}(\phi)$, $G_{u,C'}^{(R)}(\phi)$ is at most $1/\tau^2$.*

Proof. Consider any $C \in H_u^{(L)}$ and two adjacent edges $\{z_{(S_1,S_2')}, z_{(S_2,S_1')}\}$ and $\{z_{(S_1,S_2'')}, z_{(S_2,S_1'')}\}$ in $G_{u,C}^{(L)}(\phi)$ formed by joining $C = S_1 \cup S_2$ with $C' = S_1' \cup S_2'$ and $C'' = S_1'' \cup S_2'' \in H_u^{(R)}$. As the edges are adjacent, it must be the case that either $S_1' = S_1''$ or $S_2' = S_2''$, which means that $|C' \cap C''| \geq \lfloor \frac{k-1}{2} \rfloor$. Thus, the degree of a vertex $z_{(S_1,S_2')}$ in G is upper bounded by the maximum number of $C' \in H_u^{(R)}$ that all share the same $\lfloor \frac{k-1}{2} \rfloor$ variables.

Suppose ψ is τ-spread, meaning that $\deg_u(Q) \leq \frac{1}{\tau^2} \max(1, n^{\frac{k}{2}-1-|Q|})$ for $Q \subseteq [n]$. Since $\frac{k}{2} - 1 - \lfloor \frac{k-1}{2} \rfloor \leq 0$, we have that $G_{u,c}^{(L)}(\phi)$ has maximum degree $\leq 1/\tau^2$. □

7.4.2 Proof outline

With the setup in Section 7.4.1 in hand, our proof now proceeds in three conceptual steps.

Step 1: graph pruning and expander decomposition. Suppose the instance ϕ has average degree d. We first prune the instance using Lemma 3.1.1 such that the resulting constraint graph has minimum degree $\geq \varepsilon d$ while only removing ε fraction of the constraints, where $\varepsilon = o(1)$. We

further apply expander decomposition (Fact 3.1.2) to the pruned instance to obtain subinstances $\phi_1, \dots, \phi_T$ while discarding only a ε fraction of the constraints of ϕ such that the constraint graph of each ϕ_i has spectral gap $\widetilde{\Omega}(\varepsilon^2)$.

Step 2: relative spectral approximation and recovery of corrupted pairs. We show that for each expanding subinstance ϕ_i, the basic SDP for the 2-XOR instance ϕ_i is equal to $x^*(x^*)^\top$, where x^* is the planted assignment for ϕ. That is, the SDP solution is *rank* 1 and agrees with the *planted assignment* for ϕ. We show this by arguing that, for each ϕ_i, the Laplacian of the corrupted constraints in ϕ_i is a *spectral sparsifier* of the Laplacian of the constraint graph of ϕ_i (see Lemma 7.1.4). Here, we crucially use that each such constraint graph has large minimum degree and spectral gap.

From this, it is trivial to identify the corrupted edges in each ϕ_i, as they are the ones violated by the SDP solution. We are not quite done yet, however, because each constraint in ϕ corresponds to a *pair* of constraints in the original instance ψ.

Step 3: recovery of corrupted constraints from corrupted pairs. The previous step shows that for all but a ε fraction of tuples (u, C, C') where $u \in [p]$, $C \in H_u^{(L)}$, and $C' \in H_u^{(R)}$, we can recover the product $\xi_u(C)\xi_u(C')$, where $\xi_u(C) = -1$ if (u, C) is noisy in ψ, and is +1 otherwise. Because ε is small, it must be the case that for most $u \in [p]$, we know the product $\xi_u(C)\xi_u(C')$ (from Step 2) for *most* pairs (C, C') with $C \in H_u^{(L)}$ and $C' \in H_u^{(R)}$.

Suppose we knew $\xi_u(C)\xi_u(C')$ for all $(C, C') \in H_u^{(L)} \times H_u^{(R)}$. Then, it is trivial to decode $\xi_u(C)$ *up to a global sign*. Formally, we could obtain $z \in \{-1, 1\}^{H_u}$ where $z_C = \alpha\xi_u(C)$ for some $\alpha \in \{-1, 1\}$. From this, it is easy to obtain $\xi_u(C)$, as the fraction of $C \in H_u$ for which $\xi_u(C) = -1$ should be roughly $\eta < \frac{1}{2}$; so, if z has $< \frac{1}{2}$-fraction of -1's, then $z = \xi_u(C)$, and otherwise $-z = \xi_u(C)$. This, however, requires $|H_u| \geq \Omega\left(\frac{\log n}{(1-2\eta)^2}\right)$ for a high-probability result.

Additionally, we do not quite know $\xi_u(C)\xi_u(C')$ for all $(C, C') \in H_u^{(L)} \times H_u^{(R)}$: we only know this for all but a ε_u-fraction of the pairs. By forming a graph G_u where we have an edge (C, C') if (C, C') is a pair where we know $\xi_u(C)\xi_u(C')$, we can thus obtain such a z for all C in the largest connected component of G_u. Because G_u is obtained by taking a *complete biclique* and deleting only a ε_u-fraction of all edges, the largest connected component has size $(1 - \varepsilon_u)|H_u|$, and so we can recover $\xi_u(C)$ for all but a ε_u-fraction of constraints in H_u. We do this for each partition u, which finishes the proof.

7.4.3 Graph pruning and expander decomposition

This step is a simple combination of graph pruning and expander decomposition.

Lemma 7.4.6. *Fix $\varepsilon \in (0, 1)$. There is a polynomial-time algorithm such that, given a 2-XOR instance ϕ whose constraint graph has m edges and average degree d, outputs subinstances $\phi_1, \dots, \phi_T$ on disjoint variables with the following guarantees: $\phi_1, \dots, \phi_T$ contain at least $1 - \varepsilon$ fraction of the constraints in ϕ, and for each $i \in [T]$, the constraint graph G_i of ϕ_i, after adding some self-loops, has minimum degree at least $\frac{1}{3}\varepsilon d$ and $\lambda_2(\widetilde{L}_{G_i}) \geq \Omega(\varepsilon^2/\log^2 m)$.*

The self-loops in Lemma 7.4.6 are only for the analysis of $\widetilde{L}_{G_i}$ and do not correspond to actual constraints in ϕ_i. Observe that adding self-loops to a graph G does not change the *unnormalized* Laplacian L_G, but as D_G (the degree matrix) increases, the spectral gap of the *normalized* Laplacian, i.e. $\lambda_2(\widetilde{L}_G) = \lambda_2(D_G^{-1/2} L_G D_G^{-1/2})$, may decrease. The expander decomposition algorithm (Fact 3.1.2)

guarantees that each piece, even after adding self-loops to preserve degrees, has large spectral gap. This does not change the subinstances $\phi_1, \dots, \phi_T$, but in the next section, it is crucial that we use this stronger guarantee to ensure a lower bound on the minimum degree.

Proof of Lemma 7.4.6. We first apply the graph pruning algorithm (Lemma 3.1.1) such that the resulting instance has minimum degree $\geq \frac{\varepsilon}{3}d$ and at least $(1-\frac{2}{3}\varepsilon)m$ constraints. Then, we apply expander decomposition (Fact 3.1.2) that partitions the vertices of the pruned graph G' into $V_1, \dots, V_T$ such that the number of edges across partitions is at most $\frac{\varepsilon}{3}m$, and for each $i \in [T]$, the normalized Laplacian satisfies $\lambda_2(\widetilde{L}_{G'\{V_i\}}) \geq \Omega(\varepsilon^2/\log^2 m)$. Here we recall that $G'\{V_i\}$ is the induced subgraph of G' with self-loops such that the vertices in $G'\{V_i\}$ have the same degrees as in G'.

In total, we have removed at most εm edges. This completes the proof. □

7.4.4 Rank-1 SDP solution from expansion and relative spectral approximation

We next show that for each subinstance ϕ_i obtained from Lemma 7.4.6, its constraint graph G and the subgraph of corrupted edges H satisfy $L_H \prec \frac{1}{2}L_G$. Recall from Lemmas 7.1.4 and 7.1.5 that this implies the basic SDP for the 2-XOR ϕ_i is rank 1 and agrees with the planted assignment of ϕ.

The next lemma is analogous to Lemma 7.1.5 but differs in an important way: a constraint in ϕ is corrupted if and only if exactly one of the two corresponding constraints in ψ is corrupted; thus, the corruptions in ϕ are *correlated*. This is why each constraint in ϕ is obtained from one clause in $H_u^{(L)}$ and one clause in $H_u^{(R)}$ (recall Definition 7.4.1), so that in the proof below we have independent randomness to perform a "2-step sparsification" proof. It is also worth noting that the following lemma requires not just a lower bound on the minimum degree and spectral gap of G but also that the original bipartite k-XOR instance ψ is *well-spread*, which allows us to apply Lemma 7.4.5.

Same as Lemma 7.1.5, the following lemma is a purely graph-theoretic statement.

Lemma 7.4.7 (Relative spectral approximation with correlated subsamples)**.** *Suppose $G = (V, E)$ is an n-vertex graph with minimum degree $d_{\min}$ (self-loops and parallel edges allowed) and spectral gap $\lambda_2(\widetilde{L}_G) = \lambda > 0$. Let $m_1, m_2 \in \mathbb{N}$, $\eta \in [0, 1/2)$, and let $\xi_1^{(1)}, \dots, \xi_{m_1}^{(1)}, \xi_1^{(2)}, \dots, \xi_{m_2}^{(2)}$ be i.i.d. random variables that take value -1 with probability η and $+1$ otherwise. Suppose there is an injective map that maps each edge $e \mapsto (c_1(e), c_2(e)) \in [m_1] \times [m_2]$, and for each $i \in [m_1]$ (resp. $j \in [m_2]$) define $G_i^{(1)}$ (resp. $G_j^{(2)}$) be the subgraph of G with edge set $\{e \in E : c_1(e) = i\}$ (resp. $\{e \in E : c_2(e) = j\}$). Moreover, suppose $G_i^{(1)}$ and $G_j^{(2)}$ have maximum degree $\leq \Delta$ for all $i \in [m_1]$, $j \in [m_2]$.*

Let H be the subgraph of G with edge set $\left\{e \in E : \xi_{c_1(e)}^{(1)} \xi_{c_2(e)}^{(2)} = -1\right\}$. There is a universal constant $B > 0$ such that if $d_{\min}\lambda \geq B\Delta \log n$, then with probability $1 - O(n^{-2})$,

$$L_H \leq \max\left((1+\delta)\cdot 2\eta(1-\eta),\ \frac{1}{3}\right)\cdot L_G$$

for $\delta = \sqrt{\frac{B\Delta \log n}{d_{\min}\lambda}}$.

Let $\gamma := 1 - 2\eta > 0$ since $\eta < \frac{1}{2}$. Notice that $2\eta(1-\eta) = \frac{1}{2}(1-\gamma^2)$, which approaches $\frac{1}{2}$ as $\eta \to \frac{1}{2}$. Thus, if $\delta \leq \gamma^2$, then $(1+\delta)\cdot 2\eta(1-\eta) \leq (1+\gamma^2)\cdot\frac{1}{2}(1-\gamma^2) < \frac{1}{2}$, and $L_H \prec \frac{1}{2}L_G$ suffices to

conclude via Lemma 7.1.4 that the SDP relaxation on the expanding subinstance is rank 1 and recovers the planted assignment, which also gives us the set of corrupted constraints.

Proof of Lemma 7.4.7. First, note that by the definition of Laplacian and the spectral gap of L_G, $\mathrm{span}(\vec{1})$ is exactly the null space of L_G and is contained in the null space of L_H. Therefore, recalling that $L_G = D_G^{1/2}\widetilde{L}_G D_G^{1/2}$, it suffices to prove that

$$\left\|(\widetilde{L}_G^{\dagger})^{1/2} D_G^{-1/2} L_H D_G^{-1/2} (\widetilde{L}_G^{\dagger})^{1/2}\right\|_2 \le \max\left((1+\delta)\cdot 2\eta(1-\eta),\ \frac{1}{3}\right). \tag{7.1}$$

Here $\widetilde{L}_G^{\dagger}$ is the pseudo-inverse of $\widetilde{L}_G$, and $\|\widetilde{L}_G^{\dagger}\|_2 \le 1/\lambda$. For simplicity, for any graph G', we will write $\widehat{L}_{G'} \coloneqq (\widetilde{L}_G^{\dagger})^{1/2} D_G^{-1/2} L_{G'} D_G^{-1/2} (\widetilde{L}_G^{\dagger})^{1/2}$. Thus,

$$\widehat{L}_H = \sum_{e\in E} \mathbf{1}\left(\xi^{(1)}_{c_1(e)}\xi^{(2)}_{c_2(e)} = -1\right)\cdot \widehat{L}_e\,, \text{ and } \mathbb{E}[\widehat{L}_H] = 2\eta(1-\eta)\sum_{e\in E}\widehat{L}_e\,.$$

Note that $\sum_{e\in E}\widehat{L}_e = \widehat{L}_G$, a projection matrix, thus $\left\|\sum_{e\in E}\widehat{L}_e\right\|_2 = 1$.

For each $i \in [m_1]$, we further define $G^{(1)}_{i,+}$ and $G^{(1)}_{i,-}$ to be (random) edge-disjoint subgraphs of $G^{(1)}_i$ where $G^{(1)}_{i,+}$ has edge set $\left\{e \in E : c_1(e) = i, \xi^{(2)}_{c_2(e)} = +1\right\}$ and $G^{(1)}_{i,-}$ has edge set $\left\{e \in E : c_1(e) = i, \xi^{(2)}_{c_2(e)} = -1\right\}$. Note that $G^{(1)}_{i,+}$, $G^{(1)}_{i,-}$ are independent of $\xi^{(1)} = (\xi^{(1)}_1, \ldots, \xi^{(1)}_{m_1})$. By the maximum degree bound on $G^{(1)}_i$, we have that $\left\|L_{G^{(1)}_{i,+}}\right\|_2$ and $\left\|L_{G^{(1)}_{i,-}}\right\|_2 \le \left\|L_{G^{(1)}_i}\right\|_2 \le 2\Delta$. Thus,

$$\left\|\widehat{L}_{G^{(1)}_{i,+}}\right\|_2,\ \left\|\widehat{L}_{G^{(1)}_{i,-}}\right\|_2 \le \left\|\widehat{L}_{G^{(1)}_i}\right\|_2 \le 2\Delta\cdot\left\|\widetilde{L}_G^{\dagger}\right\|_2\cdot\left\|D_G^{-1}\right\|_2 \le \frac{2\Delta}{d_{\min}\lambda}\,. \tag{7.2}$$

Similarly, for $j \in [m_2]$, $G^{(2)}_{j,+}$ and $G^{(2)}_{j,-}$ are (random) edge-disjoint subgraphs of $G^{(2)}_j$ independent of $\xi^{(2)} = (\xi^{(2)}_1, \ldots, \xi^{(2)}_{m_2})$ such that $\left\|\widehat{L}_{G^{(2)}_{j,+}}\right\|_2$ and $\left\|\widehat{L}_{G^{(2)}_{j,-}}\right\|_2 \le \frac{2\Delta}{d_{\min}\lambda}$.

Now, we first fix $\xi^{(2)} \in \{-1,1\}^{m_2}$. Observe that we can write $\widehat{L}_H$ as

$$\widehat{L}_H = \sum_{i\in[m_1]} \mathbf{1}(\xi^{(1)}_i = +1)\cdot\widehat{L}_{G^{(1)}_{i,-}} + \mathbf{1}(\xi^{(1)}_i = -1)\cdot\widehat{L}_{G^{(1)}_{i,+}}\,, \tag{7.3}$$

and

$$\begin{aligned}\mathbb{E}[\widehat{L}_H|\xi^{(2)}] &= (1-\eta)\sum_{i\in[m_1]}\widehat{L}_{G^{(1)}_{i,-}} + \eta\sum_{i\in[m_1]}\widehat{L}_{G^{(1)}_{i,+}}\\ &= \sum_{e\in E}\left((1-\eta)\cdot\mathbf{1}(\xi^{(2)}_{c_2(e)} = -1) + \eta\cdot\mathbf{1}(\xi^{(2)}_{c_2(e)} = +1)\right)\cdot\widehat{L}_e\\ &\coloneqq \sum_{e\in E} w_{c_2(e)}\cdot\widehat{L}_e\,.\end{aligned} \tag{7.4}$$

Here $w_{c_2(e)} \in \{\eta, 1-\eta\}$, thus $\left\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\right\|_2 \ge \eta\left\|\sum_{e\in E}\widehat{L}_e\right\|_2 = \eta$.

We now split the analysis into two cases. Let $\eta_0 \coloneqq 1/12$.

Case 1: $\eta \geq \eta_0$. In light of Eq. (7.3), we define $X_i := \mathbf{1}(\xi_i^{(1)} = +1) \cdot \widehat{L}_{G_{i,-}^{(1)}} + \mathbf{1}(\xi_i^{(1)} = -1) \cdot \widehat{L}_{G_{i,+}^{(1)}}$ such that $\widehat{L}_H = \sum_{i \in [m_1]} X_i$. Moreover, we have that $X_i \succeq 0$ and $\|X\|_2 \leq \frac{2\Delta}{d_{\min}\lambda}$ almost surely from Equation (7.2). Thus, applying matrix Chernoff (Fact 3.4.5), we get

$$\begin{aligned} \Pr_{\xi^{(1)}} \left\{ \left\| \widehat{L}_H \right\|_2 \geq (1+\delta) \left\| \mathbb{E}[\widehat{L}_H | \xi^{(2)}] \right\|_2 \right\} &\leq n \cdot \exp\left(-\frac{1}{3}\delta^2 \left\| \mathbb{E}[\widehat{L}_H | \xi^{(2)}] \right\|_2 \cdot \frac{d_{\min}\lambda}{2\Delta} \right) \\ &\leq n \cdot \exp\left(-\frac{\delta^2 \eta d_{\min}\lambda}{6\Delta} \right), \end{aligned} \tag{7.5}$$

which is at most $O(n^{-2})$ as long as $\delta^2 \geq \frac{B_1 \Delta \log n}{d_{\min}\lambda}$ for a large enough constant B_1.

Next, we similarly prove concentration for $\left\| \mathbb{E}[\widehat{L}_H | \xi^{(2)}] \right\|_2$ over $\xi^{(2)}$. Recalling Equation (7.4),

$$\mathbb{E}[\widehat{L}_H | \xi^{(2)}] = \sum_{e \in E} w_{c_2(e)} \cdot \widehat{L}_e = \sum_{j \in [m_2]} w_j \sum_{e \in G_j^{(2)}} \widehat{L}_e = \sum_{j \in [m_2]} w_j \cdot \widehat{L}_{G_j^{(2)}}.$$

$\mathbb{E}[w_j] = 2\eta(1-\eta)$, and $\left\| \mathbb{E}_{\xi^{(2)}} \mathbb{E}[\widehat{L}_H | \xi^{(2)}] \right\|_2 = 2\eta(1-\eta) \left\| \sum_{e \in E} \widehat{L}_e \right\|_2 = 2\eta(1-\eta)$. Since $\left\| w_j \widehat{L}_{G_j^{(2)}} \right\|_2 \leq \frac{2(1-\eta)\Delta}{d_{\min}\lambda}$, we can apply matrix Chernoff again:

$$\Pr_{\xi^{(2)}} \left\{ \left\| \mathbb{E}[\widehat{L}_H | \xi^{(2)}] \right\|_2 \geq (1+\delta') \cdot 2\eta(1-\eta) \right\} \leq n \cdot \exp\left(-\frac{1}{3}\delta'^2 \cdot 2\eta(1-\eta) \cdot \frac{d_{\min}\lambda}{2(1-\eta)\Delta} \right) \tag{7.6}$$

which is at most $O(n^{-2})$ as long as $\delta'^2 \geq \frac{B_2 \Delta \log n}{d_{\min}\lambda}$ for a large enough constant B_2. Combining both tail bounds, by the union bound, we have that with probability at least $1 - O(n^{-2})$, $\left\| \widehat{L}_H \right\|_2 \leq (1+\delta) \cdot 2\eta(1-\eta)$ as long as $\delta^2 \geq \frac{B\Delta \log n}{d_{\min}\lambda}$ for a large enough B. This establishes Equation (7.1), proving the lemma for this case.

Case 2: $\eta < \eta_0$. To handle this case, observe that the exact same analysis goes through for $\widetilde{H} = \{e \in E : \xi_{c_1(e)}^{(1)} = -1 \text{ or } \xi_{c_2(e)}^{(2)} = -1\} \supseteq H$. Indeed, similar to Eqs. (7.3) and (7.4), we have $\widehat{L}_{\widetilde{H}} = \sum_{i \in [m_1]} \widetilde{X}_i$ where $\widetilde{X}_i = \mathbf{1}(\xi_i^{(1)} = +1) \cdot \widehat{L}_{G_{i,-}^{(1)}} + \mathbf{1}(\xi_i^{(1)} = -1) \cdot \widehat{L}_{G_i^{(1)}}$ (notice the 2nd term is $G_i^{(1)}$ instead of $G_{i,+}^{(1)}$), and

$$\mathbb{E}[\widehat{L}_{\widetilde{H}} | \xi^{(2)}] = (1-\eta) \sum_{i \in [m_1]} \widehat{L}_{G_{i,-}^{(1)}} + \eta \sum_{i \in [m_1]} \widehat{L}_{G_i^{(1)}} = \sum_{e \in E} \widetilde{w}_{c_2(e)} \cdot \widehat{L}_e = \sum_{j \in [m_2]} \widetilde{w}_j \cdot \widehat{L}_{G_j^{(2)}},$$

where $\widetilde{w}_j = 1$ if $\xi_j^{(2)} = -1$ and η if $\xi_j^{(2)} = +1$, hence $\mathbb{E}[\widetilde{w}_j] = \eta + \eta(1-\eta) = \eta(2-\eta)$. Moreover, $\left\| \mathbb{E}_{\xi^{(2)}} \mathbb{E}[\widehat{L}_{\widetilde{H}} | \xi^{(2)}] \right\|_2 = \eta(2-\eta) \left\| \sum_{e \in E} \widehat{L}_e \right\|_2 = \eta(2-\eta)$.

First, set $\eta = \eta_0$, and let $\widetilde{H}_0$ be the random subgraph as defined above. Similar to Eqs. (7.5) and (7.6), we apply matrix Chernoff (Fact 3.4.5) and get that with probability $1 - O(n^{-2})$, $\left\| \widehat{L}_{\widetilde{H}_0} \right\|_2 \leq (1+\delta) \cdot \eta_0(2-\eta_0)$ for $\delta = \sqrt{\frac{B\Delta \log n}{d_{\min}\lambda}} \leq 1$. In particular, this means that $L_{\widetilde{H}_0} \preceq 2\eta_0(2-\eta_0) L_G \preceq \frac{1}{3} L_G$ when $\eta_0 = 1/12$.

Now, fix any $\eta < \eta_0$. We can obtain a coupling between this case and the case when $\eta = \eta_0$ by randomly changing $\xi_i^{(1)}$ and $\xi_j^{(2)}$ from $+1$ to -1 (while not flipping the ones with -1). Notice

that $\widetilde{H}$ is monotone increasing as we change any $+1$ to -1 (whereas H is not!), thus we must have $\widetilde{H} \subseteq \widetilde{H}_0$ in this coupling. Then, as $H \subseteq \widetilde{H}$, we have

$$L_H \preceq L_{\widetilde{H}} \preceq L_{\widetilde{H}_0} \preceq \frac{1}{3} L_G$$

with probability $1 - O(n^{-2})$. This finishes the proof of Lemma 7.4.7. □

7.4.5 Recovery of corrupted constraints from corrupted pairs

We have thus shown that, with probability $\geq 1 - 1/\mathrm{poly}(n)$, we can *exactly* recover the set of corrupted constraints within each expanding subinstance $\phi_1, \ldots, \phi_T$. Recall that after pruning and expander decomposition (Lemma 7.4.6), the expanding subinstances contain a $(1-\varepsilon)$-fraction of all edges in the instance ϕ, and the set of edges removed only depends on the constraint graph and not the right-hand sides of ϕ. As stated in Observation 7.4.2, the instance ϕ has exactly $m^2/4p$ edges, and they correspond exactly to the set $\{(u, C, C') : u \in [p], C \in H_u^{(L)}, C' \in H_u^{(R)}\}$, and moreover an edge e in ϕ is corrupted if and only if exactly one of the two constraints $(u, C), (u, C')$ is corrupted in the original instance ψ, where e corresponds to (u, C, C'). For each $u \in [p]$ and $C \in H_u = H_u^{(L)} \cup H_u^{(R)}$, let $\xi_u(C) = -1$ if (u, C) is corrupted in ψ, and 1 otherwise. It thus follows that we have learned, for $1 - \varepsilon$ fraction of all $\{(u, C, C') : u \in [p], C \in H_u^{(L)}, C' \in H_u^{(R)}\}$, the product $\xi_u(C) \cdot \xi_u(C')$.

It now remains to show how to recover $\xi_u(C)$ for most $u \in [p]$, $C \in H_u$. For each $u \in [p]$, let $P_u \subseteq \{(C, C') : C \in H_u^{(L)}, C' \in H_u^{(R)}\}$ such that we have determined $\xi_u(C) \cdot \xi_u(C')$, and let $P = \cup_{u \in [p]} P_u$. We know that $|P| \geq (1-\varepsilon)\frac{m^2}{4p}$. Let ε_u be chosen so that $|P_u| = (1-\varepsilon_u)\frac{m^2}{4p^2}$, i.e., ε_u is the fraction of pairs in $H_u^{(L)} \times H_u^{(R)}$ that were deleted in Lemma 7.4.6. Notice that we have

$$\begin{aligned} (1-\varepsilon)\frac{m^2}{4p} \leq |P| = \sum_{u \in [p]} |P_u| = \frac{m^2}{4p^2} \sum_{u \in [p]} (1 - \varepsilon_u) \\ \implies \frac{1}{p} \sum_{u \in [p]} \varepsilon_u \leq \varepsilon \,. \end{aligned} \tag{7.7}$$

One can think of this problem as a collection of disjoint *satisfiable* (noiseless) 2-XOR instances on P_u, where each P_u is a biclique ($\frac{m}{2p}$ vertices on each side) with ε_u fraction of edges are removed.

Algorithm 7.4.8 (Recover corrupted constraints from corrupted pairs)**.**

Given: For each $u \in [p]$, a set $P_u \subseteq H_u^{(L)} \times H_u^{(R)}$ such that $|P_u| = (1-\varepsilon_u)\frac{m^2}{4p^2}$ for $\varepsilon_u \in [0, 1]$, along with "right-hand sides" $\xi_u(C) \cdot \xi_u(C')$ for each $(C, C') \in P_u$.

Output: For each $u \in [p]$, disjoint subsets $\mathcal{A}_u^{(1)}, \mathcal{A}_u^{(2)} \subseteq H_u$.

Operation:

1. **Initialize:** $\mathcal{A}_u^{(1)}, \mathcal{A}_u^{(2)} = \emptyset$ for each $u \in [p]$.
2. **For each** $u \in [p]$:
   (a) If $\varepsilon_u \geq 1/3$, set $\mathcal{A}_u^{(1)} = H_u$ and $\mathcal{A}_u^{(2)} = \emptyset$.
   (b) Else if $\varepsilon_u < 1/3$, let G_u be the graph with vertex set $H_u = H_u^{(L)} \cup H_u^{(R)}$ with edges given by P_u, and let S_u be the size of the largest connected component

in G_u.

(c) As S_u is connected in G_u, and we know $\xi_u(C)\xi_u(C')$ for each edge (C, C') in G_u, by solving a linear system of equations we obtain $z \in \{-1,1\}^{H_u}$ such that either $z_C = \xi_u(C)$ for all $C \in S_u$, or $z_C = -\xi_u(C)$ for all $C \in S_u$. That is, $z_C = \xi_u(C)$ up to a global sign.

(d) Pick the global sign to minimize the number of $C \in S_u$ for which $z_C = -1$. Set $\mathcal{A}_u^{(1)} = H_u \setminus S_u$ and $\mathcal{A}_u^{(2)} = \{C \in S_u : z_C = -1\}$.

3. Output $\{\mathcal{A}_u^{(1)}\}_{u\in[p]}, \{\mathcal{A}_u^{(2)}\}_{u\in[p]}$.

We now analyze Algorithm 7.4.8 via the following lemma.

Lemma 7.4.9. *Let $\eta \in [0,1/2)$, and let $|H_u| = \frac{m}{p} \geq \frac{24k}{(1-2\eta)^2}\log n$ and $|P_u| = (1-\varepsilon_u)\frac{m^2}{4p^2}$ with $\varepsilon_u \in [0,1]$ for each $u \in [p]$, and $\frac{1}{p}\sum_{u\in[p]}\varepsilon_u \leq \varepsilon$. The outputs of Algorithm 7.4.8 satisfy the following: (1) $\sum_{u\in[p]}|\mathcal{A}_u^{(1)}| \leq 4\varepsilon m$, and (2) with probability $1-n^{-k}$ over the noise $\{\xi_u(C)\}_{u\in[p],C\in H_u}$, for every $u \in [p]$ we have that $\mathcal{A}_u^{(2)} = \{C \in H_u : \xi_u(C) = -1\} \setminus \mathcal{A}_u^{(1)}$.*

Proof. Suppose that $\varepsilon_u < 1/3$. Observe that G_u is a graph obtained by taking a biclique with left vertices $H_u^{(L)}$ and right vertices $H_u^{(R)}$, i.e., with $m/2p$ left vertices and $m/2p$ right vertices. The following lemma shows that the largest connected component S_u in G_u has size at least $\frac{m}{p}(1-\varepsilon_u)$.

Claim 7.4.10. Let $K_{n,n}$ be the complete bipartite graph with n left vertices L and n right vertices R. Let G be a graph obtained by deleting εn^2 edges from $K_{n,n}$. Then, the largest connected component in G has size $\geq 2n(1-\varepsilon)$.

We postpone the proof of Claim 7.4.10 to the end of the section, and continue with the proof of Lemma 7.4.9.

We now argue that we can efficiently obtain the vector z in Step (2c) of Algorithm 7.4.8. Indeed, this is done as follows. First, pick one $C_0 \in S_u$ arbitrarily, and set $z_{C_0} = 1$. Then, we propagate in a breadth-first search manner: for any edge (C, C') in S_u where z_C is determined, set $z_{C'} = z_C \cdot \xi_u(C)\xi_u(C')$. We repeat this process until we have labeled all of S_u. Notice that as S_u is a connected component, fixing z_{C_0} for any $C_0 \in S_u$ uniquely determines the assignment of all S_u, thus we have obtained $z_C = \xi_u(C)$ up to a global sign.

Now, we observe that S_u does not depend on the noise in ψ. Indeed, this is because the pruning and expander decomposition (and thus the graph G_u) depends solely on the constraint graph G of the instance ϕ, and not on the right-hand sides of the constraints. The following lemma thus shows that with high probability over the noise, the number of $C \in S_u$ where $\xi_u(C) = -1$ is strictly less than $1/2|S_u|$. Hence, in Step (2d), by picking the assignment $\pm z$ that minimizes the number of $C \in S_u$ with $\xi_u(C) = -1$, we see that $\mathcal{A}_u^{(2)} = \{C \in S_u : z_C = -1\} = \{C \in S_u : \xi_u(C) = -1\}$.

Claim 7.4.11. Let $\eta \in (0,1/2)$ be the corruption probability, and assume that $p \leq n^k$ and $\frac{m}{p} \geq \frac{24k}{(1-2\eta)^2}\log n$. With probability $1-n^{-k}$ over the noise in ψ, it holds that for each $u \in [p]$ with $\varepsilon_u < 1/3$, $|\{C \in S_u : \xi_u(C) = -1\}| < \frac{1}{2}|S_u|$.

We postpone the proof of Claim 7.4.11, and finish the proof of Lemma 7.4.9. We next bound

$\sum_{u\in[p]} |\mathcal{A}_u^{(1)}|$. By Eq. (7.7) we have that $\frac{1}{p}\sum_u \varepsilon_u \le \varepsilon$. Thus,

$$\sum_{u:\varepsilon_u \ge 1/3} |H_u| \le \frac{m}{p} \sum_{u:\varepsilon_u \ge 1/3} 3\varepsilon_u \le 3\varepsilon m\,.$$

Moreover, by Claim 7.4.10 we have $|S_u| \ge (1-\varepsilon_u)|H_u| = (1-\varepsilon_u)\frac{m}{p}$. Thus,

$$\sum_{u:\varepsilon_u < 1/3} |H_u \setminus S_u| \le \sum_{u:\varepsilon_u<1/3} \varepsilon_u \cdot \frac{m}{p} \le \varepsilon m\,.$$

Therefore, combining the two,

$$\sum_{u\in[p]} |\mathcal{A}_u^{(1)}| = \sum_{u:\varepsilon_u<1/3} |H_u \setminus S_u| + \sum_{u:\varepsilon_u\ge 1/3} |H_u| \le 4\varepsilon m\,,$$

which finishes the proof of Lemma 7.4.9. □

In the following, we prove Claims 7.4.10 and 7.4.11.

Proof of Claim 7.4.10. Let $S_1,\dots,S_t$ be the connected components of G. Let $\ell_i = |S_i \cap L|$ and $r_i = |S_i \cap R|$. The number of edges in G is at most $\sum_{i=1}^t \ell_i r_i$.

Now, suppose that the largest connected component of G has size at most M. Then, we have that $\ell_i + r_i \le M$ for all $i \in [t]$. Notice that the number of edges deleted from $K_{n,n}$ to produce G must be at least $n^2 - \sum_{i=1}^t \ell_i r_i$, and this is at most εn^2. Hence, by maximizing the quantity $\sum_{i=1}^t \ell_i r_i$ subject to $\ell_i + r_i \le M$ for all $i \in [t]$ and $\sum_{i=1}^t \ell_i + r_i = 2n$, we can obtain a lower bound on the number of edges deleted from $K_{n,n}$ in order for the largest connected component of G to have size at most M. We have that

$$\sum_{i=1}^t \ell_i r_i \le \sum_{i=1}^t \left(\frac{\ell_i + r_i}{2}\right)^2 \le \frac{M}{2}\cdot \sum_{i=1}^t \frac{\ell_i + r_i}{2} = \frac{nM}{2}\,,$$

where the first inequality is by the AM-GM inequality. Thus,

$$\varepsilon n^2 \ge n^2 - \frac{nM}{2} \implies M \ge 2n(1-\varepsilon)\,,$$

which finishes the proof. □

Proof of Claim 7.4.11. Let u be such that $\varepsilon_u < 1/3$, and let S_u be the largest connected component in G_u. Observe that S_u is determined solely by the constraint graph of ϕ, and in particular does not depend on the noise in ϕ (and hence on the noise in ψ). As $p \le n^k$ by assumption, it thus suffices to show that for each $u \in [p]$, with probability $1 - n^{-2k}$ it holds that $|\{C \in S_u : \xi_u(C) = -1\}| < \frac{1}{2}|S_u|$. Notice that $|\{C \in S_u : \xi_u(C) = -1\}|$ is simply the sum of $|S_u|$ Bernoulli(η) random variables. By Hoeffding's inequality, with probability $\ge 1 - \exp(-2\delta^2|S_u|)$ it holds that $|\{C \in S_u : \xi_u(C) = -1\}| \le (\eta+\delta)|S_u|$. We choose $\delta = \frac{1}{2}(\frac{1}{2} - \eta)$ such that $\eta + \delta < \frac{1}{2}$ for $\eta \in (0, \frac{1}{2})$. Then, by noting that $2\delta^2|S_u| \ge 2\delta^2(1-\varepsilon_u)|H_u| \ge \frac{1}{2}(\frac{1}{2}-\eta)^2 \cdot \frac{2}{3}\cdot\frac{m}{p} \ge 2k\log n$ since $\frac{m}{p} \ge \frac{24k}{(1-2\eta)^2}\log n$, Claim 7.4.11 follows. □

7.4.6 Finishing the proof of Lemma 7.3.2

Proof of Lemma 7.3.2. We are given an τ-spread p-bipartite k-XOR instance ψ with constraint graph $H = \{H_u\}_{u\in[p]}$, where we recall from Definition 7.3.1 that (1) $m = |H|$ and each $|H_u| = \frac{m}{p} \geq 2\lfloor \frac{1}{2\tau^2} \rfloor$ and $\frac{m}{p}$ is even, and (2) for any $Q \subseteq [n]$, $\deg_u(Q) \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1-|Q|})$. For convenience, let $m \geq n^{\frac{k-1}{2}}\sqrt{p}\cdot\beta$ where $\beta := C \cdot \frac{(k\log n)^{3/2}}{\tau\gamma^2\varepsilon^{3/2}}$ and $\gamma := 1-2\eta \in (0,1]$ since $\eta \in [0, \frac{1}{2})$.

First, we construct the 2-XOR instance ϕ defined in Definition 7.4.1. As stated in Observation 7.4.2, the average degree is at least $d := \frac{1}{4}\beta^2$, and furthermore, by Lemma 7.4.5, the maximum degree of $G^{(L)}_{u,C}(\phi)$ and $G^{(R)}_{u,C'}(\phi)$ for any $u \in [p]$, $C \in H^{(L)}_u$ and $C' \in H^{(R)}_u$ is bounded by $\Delta := 1/\tau^2$. The algorithm then follows the steps outlined in Section 7.4.2.

Step 1. We apply graph pruning and expander decomposition (Lemma 7.4.6) with parameter $\varepsilon' := \frac{1}{4}\varepsilon$, which decomposes ϕ into $\phi_1, \ldots, \phi_T$ such that they contain $1-\varepsilon'$ fraction of the constraints in ϕ, and their constraint graphs (after adding some self-loops due to expander decomposition) have minimum degree $d_{\min} \geq \frac{1}{3}\varepsilon' d = \frac{1}{48}\varepsilon\beta^2$ and spectral gap $\lambda \geq \Omega(\varepsilon'^2/\log^2 m) = \Omega(\varepsilon^2/(k^2\log^2 n))$.

Step 2. We solve the SDP relaxation for each subinstance ϕ_i. Let G be the constraint graph of ϕ_i (with at most $N \leq n^{k-1}$ vertices) and H be the corrupted edges of G. We apply the relative spectral approximation result (Lemma 7.4.7) with $\xi^{(1)}_1, \ldots, \xi^{(1)}_{m/2p}$ (resp. $\xi^{(2)}_1, \ldots, \xi^{(2)}_{m/2p}$) being $\{-1,1\}$ random variables indicating whether each $C \in H^{(L)}_u$ (resp. $C' \in H^{(R)}_u$) is corrupted. Moreover, the subgraphs $G^{(1)}_i$ and $G^{(2)}_j$ in Lemma 7.4.7 (which are simply subgraphs of $G^{(L)}_{u,C}(\phi)$ and $G^{(R)}_{u,C'}(\phi)$) have maximum degree $\leq \Delta = 1/\tau^2$. Thus, we have that with probability $1 - O(N^{-2})$,

$$L_H \leq \max\left((1+\delta)\cdot 2\eta(1-\eta),\ \frac{1}{3}\right)\cdot L_G$$

where $\delta = \sqrt{\frac{B\Delta\log N}{d_{\min}\lambda}} \leq O\left(\sqrt{\frac{k^3\log^3 n}{\tau^2\varepsilon^3\beta^2}}\right)$. Plugging in β (for large enough C), we get that $\delta \leq \gamma^2 = 1 - 4\eta(1-\eta)$. Therefore, we have $(1+\delta)\cdot 2\eta(1-\eta) \leq (1+\gamma^2)\cdot\frac{1}{2}(1-\gamma^2) < \frac{1}{2}$, hence $L_H \prec \frac{1}{2}L_G$. By union bound over all $T \leq N$ subinstances, this holds for all subinstances ϕ_i with probability $1 - \frac{1}{\mathrm{poly}(n)}$ over the randomness of the noise.

Then, by Lemma 7.1.4, the SDP relaxation has a unique optimum which is the planted assignment. Thus, we can identify the set of corrupted edges in each ϕ_i.

Step 3. So far we have identified, for $\geq 1-\varepsilon'$ fraction of all $\{(u, C, C') : u \in [p], C \in H^{(L)}_u, C' \in H^{(R)}_u\}$, the product $\xi_u(C)\cdot\xi_u(C')$, where $\xi_u(C) = -1$ if (u, C) is corrupted in ψ, and $+1$ otherwise. Let $P_u \subseteq \{(C, C') : C \in H^{(L)}_u, C' \in H^{(R)}_u\}$ be such pairs for each $u \in [p]$, and let $P = \cup_{u\in[p]} P_u$. Note that $|P| \geq (1-\varepsilon')\frac{m^2}{4p}$ and P depends only on H and not on the noise.

We then run Algorithm 7.4.8. By the assumption that $\tau \leq \frac{c\gamma}{\sqrt{k\log n}}$ for a small enough c, we have $|H_u| = \frac{m}{p} \geq 2\lfloor\frac{1}{2\tau^2}\rfloor \geq \frac{24k}{(1-2\eta)^2}$, which is the condition we need in Lemma 7.4.9. Thus, with probability $1 - n^{-k}$, Algorithm 7.4.8 outputs (1) $\mathcal{A}_1 \subseteq H$ which only depends on H and such that $|\mathcal{A}_1| \leq 4\varepsilon' m = \varepsilon m$, and (2) $\mathcal{A}_2 \subseteq H$, the set of corrupted constraints in $H \setminus \mathcal{A}_1$. This completes the proof of Lemma 7.3.2. □

7.5 Notions of relative approximation

In this chapter, we have encountered several notions of relative graph approximations. Let G be an n-vertex graph, and let H be a random subgraph of G by selecting each edge with a fixed probability $\eta \in (0,1)$. We are interested in the sufficient conditions on G for each of the following to hold with probability $1 - o(1)$ (for some $\delta = o(1)$):

(1) **Relative cut approximation**: $x^\top L_H x \le (1+\delta)\eta \cdot x^\top L_G x$ for all $x \in \{-1,1\}^n$.

(2) **Relative SDP approximation**: $\langle X, L_H \rangle \le (1+\delta)\eta \cdot \langle X, L_G \rangle$ for all symmetric matrices $X \succeq 0$ with $\mathrm{diag}(X) = \mathbb{I}$.

(3) **Relative spectral approximation:** $L_H \preceq (1+\delta)\eta \cdot L_G$.

Here, we only state one-sided inequalities, as solving noisy XOR requires only an upper bound on L_H. Note also that the above is in increasing order: relative spectral approximation implies relative SDP approximation, which in turn implies relative cut approximation.

Recall from Lemma 7.1.3 that a lower bound on the min-cut of G suffices for cut approximation to hold, while Lemma 7.1.5 shows that lower bounds on the minimum degree and spectral gap of G suffice for spectral approximation to hold. It is natural to wonder whether a min-cut lower bound is sufficient for SDP approximation as well, since it allows us to efficiently recover the planted assignment in a noisy planted 2-XOR via solving an SDP relaxation (see Lemma 7.1.4). Unfortunately, there is a counterexample.

Separation of cut and SDP approximation. The example is the same graph that separates cut and spectral approximation described in [ST11]. Let n be even and $k = k(n)$. Define $G = (V, E)$ be a graph on $N = nk$ vertices where $V = \{0, 1, \ldots, n-1\} \times \{1, \ldots, k\}$ and (u, i), $(v, j) \in V$ are connected if $v = u \pm 1 \mod n$. Moreover, there is one additional edge e^* between $(0,1)$ and $(n/2, 1)$. In other words, G consists of n clusters of vertices of size k, where the clusters form a ring with a complete bipartite graph between adjacent clusters, along with a special edge e^* in the middle.

Clearly, the minimum cut of G is $2k$, which means that cut approximation holds. Essentially, the special edge e^* does not play a role here.

However, we will show that e^* breaks SDP approximation. Define vector $x_0 \in \mathbb{R}^V$ such that the (u, i) entry is

$$x_0(u, i) = \min(u, n - u)\,,$$

and vectors $x_1, \ldots, x_{n-1}$ to be cyclic shifts of x_0: for $w \in \{0, 1, \ldots, n-1\}$,

$$x_w(u, i) = x_0(u - w \pmod n,\ i)\,.$$

We note that x_0 is the vector shown in [ST11] that breaks spectral approximation. We now show that $X = \sum_{w=0}^{n-1} x_w x_w^\top$ (scaled so that X has all 1s on the diagonal) breaks SDP approximation.

First, it is easy to see that the diagonal entries of X are all equal due to symmetry. Thus, for some scaling c, $cX \succeq 0$ and $\mathrm{diag}(cX) = \mathbb{I}$.

Observe that for $w \le \frac{n}{2} - 1$, $x_w(0,1) = w$ and $x_w(\frac{n}{2}, 1) = \frac{n}{2} - w$. For $w \ge \frac{n}{2}$, $x_w(0,1) = n - w$

and $x_w(\frac{n}{2},1) = w - \frac{n}{2}$. Thus, as $x_w^\top L_{e^*} x_w = \big(x_w(0,1) - x_w(\frac{n}{2},1)\big)^2$,

$$\langle X, L_{e^*}\rangle = \sum_{w=0}^{n-1} x_w^\top L_{e^*} x_w = \sum_{w=0}^{\frac{n}{2}-1}\left(\frac{n}{2}-2w\right)^2 + \sum_{w=\frac{n}{2}}^{n-1}\left(\frac{3n}{2}-2w\right)^2 = \Theta(n^3)\,.$$

On the other hand, $x_w^\top L_{G\setminus e^*} x_w = nk^2$ for any w, thus $\langle X, L_{G\setminus e^*}\rangle = n^2k^2$. This is $o(n^3)$, i.e. dominated by $\langle X, L_{e^*}\rangle$, when $k = o(\sqrt{n})$. Since e^* is selected in H with probability η, we have that with probability η,

$$\langle X, L_H\rangle \geq \langle X, L_{e^*}\rangle \geq (1-o(1))\cdot\langle X, L_G\rangle\,,$$

which violates the desired SDP approximation.

7.6 Hypergraph decomposition

In this section, we describe the hypergraph decomposition algorithm used in Section 7.3 (for the proof of Theorem 5). This algorithm is nearly identical to the hypergraph decomposition step of Section 5.2.

Algorithm 7.6.1.

Given: A semirandom (with noise η) k-XOR instance ψ with constraint hypergraph H over n vertices, and a spread parameter $\tau \in (0,1)$.

Output: For each $t = 2,\dots,k$, a semirandom (with noise η) planted τ-spread $p^{(t)}$-bipartite t-XOR instance $\psi^{(t)}$ with constraint hypergraph $\{H_u^{(t)}\}_{u\in[p^{(t)}]}$, along with "discarded" hyperedges $H^{(1)}$.

Operation:

1. **Initialize:** $\psi^{(t)}$ to the empty instance, and $p^{(t)} = 0$ for $t = 2,\dots,k$.
2. **Fix violations greedily**:
   (a) Find a maximal nonempty violating Q. That is, find $Q \subseteq [n]$ of size $1 \leq |Q| \leq k-1$ such that $\deg(Q) = |\{C \in H : Q \subseteq C\}| > \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q|})$, and $\deg(Q') \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q'|})$ for all $Q' \supsetneq Q$.
   (b) Let $q = |Q|$. Let $u = 1 + p^{(k+1-q)}$ be a new "label", and define $H_u^{(k+1-q)}$ to be an arbitrary subset of $\{C \setminus Q : C \in H, Q \subseteq C\}$ of size exactly $2\cdot\lfloor\frac{1}{2\tau^2}\max(1, n^{\frac{k}{2}-q})\rfloor$.
   (c) Set $p^{(k+1-q)} \leftarrow 1 + p^{(k+1-q)}$, and $H \leftarrow H \setminus H_u^{(k+1-q)}$.
3. If no such Q exists, then put the remaining hyperedges in $H^{(1)}$.

Lemma 7.6.2. *Algorithm 7.6.1 has the following guarantees:*

(1) The runtime is $n^{O(k)}$,

(2) The number of "discarded" hyperedges is $m^{(1)} := |H^{(1)}| \leq \frac{1}{k\tau^2}n^{\frac{k}{2}}$,

(3) For each $t \in \{2,\dots,k\}$ and $u \in [p^{(t)}]$, $|H_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}} = 2\lfloor\frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$,

(4) For each $t = 2,\dots,k$, the instance $\psi^{(t)}$ is τ-spread.

Proof. The runtime of Algorithm 7.6.1 is obvious. We now argue that $m^{(1)}$ is small. By construction, $H^{(1)}$ is the set of remaining hyperedges when the inner loop terminates, and so we must have $\deg(\{i\}) \le \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1}) = \frac{1}{\tau^2} n^{\frac{k}{2}-1}$ for every $i \in [n]$; here, deg only counts hyperedges remaining in H. We then have $\sum_{i\in[n]} \deg(\{i\}) = k|H^{(1)}|$, as every $C \in H^{(1)}$ is counted exactly k times in the sum. Hence, $m^{(1)} \le \frac{1}{k\tau^2} n^{\frac{k}{2}}$.

Next, for each $t \in \{2, \dots, k\}$, by construction (Step (2b)) each $H_u^{(t)}$ has the same size, namely $2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$. It then follows that $m^{(t)} := \sum_{u\in[p^{(t)}]} |H_u^{(t)}| = p^{(t)} \cdot 2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$, and so $|H_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}}$. We also note that $m^{(t)}/p^{(t)}$ is clearly even.

We now argue that for each t, the instance $\psi^{(t)}$ is τ-spread. From Definition 7.3.1, we need to prove that for each $u \in [p^{(t)}]$ and $Q \subseteq [n]$, $\deg_u(Q) \le \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1-|Q|})$. To see this, let $u \in [p^{(t)}]$, and let Q_u be the set "associated" with the label u, i.e., the set picked in Step (2a) of Algorithm 7.6.1 when the label u is added in Step (2b). Note that we must have $|Q_u| = k+1-t$. Let H' denote the set of constraints in H at the time when u and $H_u^{(t)}$ is added to $\psi^{(t)}$. Namely, we have that for every $C \in H_u^{(t)}$, $Q_u \cup C \in H'$, and Q_u, C are disjoint. Now, let $R \subseteq [n]$ be a nonempty set of size at most $t-1$. First, observe that if $R \cap Q_u$ is nonempty, then we must have $\deg_u(R) = 0$ (this degree is in the hypergraph $H_u^{(t)}$). Indeed, this is because $C \cap Q_u = \emptyset$ for all $C \in H_u^{(t)}$. So, we can assume that $R \cap Q_u = \emptyset$. Next, we see that $\deg_u(R) \le \deg_{H'}(Q_u \cup R)$ (where $\deg_{H'}$ is the degree in H'), as $Q_u \cup C \in H'$ for every $C \in H_u^{(t)}$. Because Q_u was maximal whenever it was processed in our decomposition algorithm and $Q_u \subsetneq Q_u \cup R$ as R is nonempty and $R \cap Q_u = \emptyset$, it follows that

$$\begin{aligned}\deg_{H'}(Q_u \cup R) &\le \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q_u\cup R|}) = \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q_u|-|R|}) \\ &= \frac{1}{\tau^2}\max(1, n^{t-\frac{k}{2}-1-|R|}) \le \frac{1}{\tau^2}\max(1, n^{\frac{t}{2}-1-|R|}),\end{aligned}$$

where the last inequality follows because $t - \frac{k}{2} - 1 - |R| \le \frac{t}{2} - 1 - |R|$ always holds, as $t \le k$.

Finally, when $R = \emptyset$, we trivially have

$$\deg_u(\emptyset) = \left|H_u^{(t)}\right| = 2\left\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\right\rfloor \le \frac{1}{\tau^2}\max(1, n^{t-\frac{k}{2}-1}) \le \frac{1}{\tau^2}\max(1, n^{\frac{t}{2}-1}),$$

where we use again that $t - \frac{k}{2} \le \frac{t}{2}$ as $t \le k$. This finishes the proof. □

7.7 Theorem 5 when $k = 1$

In this section, we state and prove a variant of Theorem 5 for the degenerate case of $k = 1$. The algorithm here is straightforward, and we include it only for completeness.

Lemma 7.7.1 (Algorithm for noisy 1-XOR)**.** *Let $\eta \in (0, 1/2)$ be a constant. Let $n \in \mathbb{N}$ and $\varepsilon \in (0,1)$, and let $m \ge O(n \log n/\varepsilon)$. There is a polynomial-time algorithm $\mathcal{A}$ that takes as input a 1-XOR instance ψ with constraint hypergraph H and outputs two disjoint sets $\mathcal{A}_1(H), \mathcal{A}_2(\psi) \subseteq H$ with the following guarantees: (1) for any instance ψ with m constraints, $|\mathcal{A}_1(H)| \le \varepsilon m$ and $\mathcal{A}_1(H)$ only depends on H, and (2) for any $x^* \in \{-1,1\}^n$ and any k-uniform hypergraph H with at least m hyperedges, with high probability over $\psi \leftarrow \psi(H, x^*, \eta)$, it holds that $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (H \setminus \mathcal{A}_1(H))$.*

Proof. First, observe that a 1-XOR instance is a degenerate case where H is a multiset of $[n]$ of size m. Let $S \subseteq [n]$ denote the set of $i \in [n]$ where i appears in H with multiplicity $\leq c \log n$, where c is a constant to be determined later. Let $\mathcal{A}_1(H)$ denote $H \cap S$, i.e., the set of elements in H that are in S. We clearly have that $|\mathcal{A}_1(H)| \leq cn \log n \leq \varepsilon m$.

Now, let $i \notin S$. Observe that for each occurrence of i in H, we have a corresponding *independent* right-hand side $b \in \{-1, 1\}$ where $b = x_i^*$ with probability $1 - \eta$ and $-x_i^*$ with probability η. Thus, by taking the majority, we can with high probability decode x_i^* and thus determine the corrupted constraints. It thus remains to show that with probability $\geq 1 - 1/\text{poly}(n)$, the fraction of corrupted right-hand sides for i is $< \frac{1}{2}$. Indeed, by a Chernoff bound, with probability $\geq 1 - \exp(-2\delta^2 c \log n)$, it holds that the fraction of corrupted right-hand sides is at most $(\eta + \delta)$. By choosing $\delta = \frac{1}{2}(\frac{1}{2} - \eta)$ and c to be a sufficiently large constant, Lemma 7.7.1 follows. □

Part II

Extremal Girth vs. Density Trade-Offs for Hypergraphs

Chapter 8

Background and Results

A very basic, well-studied problem in extremal combinatorics is to understand the length of the shortest cycle in d-regular graph G. If $d \geq 3$, by computing the size of the ball of some radius r around a vertex v in G, one can show that G must have a cycle of length $2\log_{d-1} n + 2$; this is the well-known Moore bound for graphs. One can ask the same question more generally for irregular graphs with average degree d, i.e., what is the extremal girth vs. density trade-off for graphs? This question was resolved in the work of [AHL02], which extended the classical Moore bound to the setting of irregular graphs.

Feige's conjectured Moore bound for hypergraphs. In [Fei08], motivated by the refutation witnesses established in [FKO06] (which is covered in detail in Section 4.1.2 in Part I of this thesis), Feige made an elegant conjecture on the existence of even covers (hypergraph cycles) in sufficiently dense hypergraphs. This conjecture can be interpreted as generalizing the classical Moore bound to hypergraphs. Let us explain this conjecture below.

Definition 8.0.1 (Even covers and girth)**.** For a k-uniform hypergraph H on $[n]$, an *even cover* (hypergraph cycle) of length r is a collection of r distinct hyperedges $C_1, C_2, \ldots, C_r$ in H such that every vertex in $[n]$ appears in an even number of C_i's. The *girth* of H is the length of the smallest even cover in H.

Conjecture 8.0.2 (Feige's conjecture, Conjecture 1.2 in [Fei08])**.** *Every k-uniform hypergraph H on $[n]$ with $m \geq m_0 = O(n)\left(\frac{n}{\ell}\right)^{\frac{k}{2}-1}$ hyperedges has an even cover of length $O(\ell \log n)$.*

Conjecture 8.0.2 has implications beyond finding FKO certificates. For example, one can identify the k-uniform hypergraph H with its incidence matrix $A \in \mathbb{F}_2^{n\times m}$ that has k-sparse columns; the girth of H is the size of the smallest set of linearly dependent columns of A. By viewing A as the parity check matrix of a low-density parity check (LDPC) code, Conjecture 8.0.2 conjectures an extremal rate vs. distance trade-off for LDPC codes.

In this thesis, we prove Feige's conjecture using Kikuchi matrices. This argument is a *spectral double counting* argument that relates subexponential-time smoothed refutation algorithms and the existence of even covers in hypergraphs. As explained in Section 4.1.2, as a corollary of our proof of Conjecture 8.0.2, we show that there are efficiently verifiable witnesses of unsatisfiability for smoothed instances of all k-CSPs with $m \sim n^{k/2-\delta_k}$ constraints, for some constant δ_k, which is polynomially smaller than the threshold at which efficient refutation algorithms exist even for random k-CSPs.

A brief history of the conjecture. For $k = 2$, an even cover is a 2-regular subgraph (and thus a

union of cycles) in a graph, and thus the conjecture above reduces to the question of determining the maximum girth (the length of the smallest cycle) in a graph with n vertices and $nd/2$ edges for parameter d. As mentioned earlier, the best-known bound is due to [AHL02], which proved that for every graph on n vertices with $nd/2$ edges for $d > 2$, there is a cycle of length at most $c \log_{d-1} n$ for $c \leq 2$. The best-known lower bound on the girth is $c \log_{d-1} n$ for $c \geq 4/3$ by Margulis [Mar88] and Lubotzky, Philips and Sarnak [LPS88] via explicit constructions of Ramanujan graphs. Obtaining a tight bound on c has been an outstanding open problem for the last 3 decades.

As is typically the case for hypergraph Turán problems, much less is known for hypergraphs. When k even and $\ell = O(1)$, Naor and Verstraete [NV08] proved the conjecture. They were motivated by the connection to the rate vs. distance trade-off for LDPC codes explained above. In the more challenging case when k is odd, the bounds for $\ell = O(1)$ case in [NV08] were improved to essentially optimal ones in [Fei08]. For $\ell \gg 1$, the best previous bound for 3-uniform hypergraphs is due to a simple argument of Alon and Feige [AF09] (Lemma 3.3), who proved that every 3-uniform hypergraph with $\tilde{O}(n^2/\ell)$ hyperedges has an even cover of size ℓ (this is off by $\sim \sqrt{n}$ factor in m). For 3-uniform hypergraphs with $m \gg n^{1.5+\epsilon}$ (and the case when $m \gg n^{k/2}$ in general), [JHL+12] proved that there are even covers of size $O(1/\epsilon)$. Finally, Feige and Wagner [FW16] proved some variants ("generalized girth problems") in order to build tools to approach this conjecture.

To summarize, prior to this thesis, the conjecture was known to be true only for $\ell = O(1)$. For larger ℓ, the only approach was the combinatorial strategy introduced in [FW16]. In this thesis, we prove Feige's conjecture (up to $\log n$ slack in m) via a new *spectral double counting argument*.

Theorem 6 (Feige's conjecture is true)**.** *For every $k \in \mathbb{N}$ and $\ell = \ell(n)$, every k-uniform hypergraph H with $m \geq m_0 = n \cdot 2^{O(k)} (\frac{n}{\ell})^{\frac{k}{2}-1} \cdot \log_2 n$ hyperedges has an even cover of size $O(\ell \log n)$.*

We note that the original version of this theorem proven in [GKM22] had a larger polylog(n) factor, which was improved in a follow-up work of [HKM23]. In this thesis, we include the improvements of [HKM23] to the original proof in order to show the stronger result.

Our spectral double counting argument is heavily derived from our analysis for smoothed refutation using our Kikuchi matrices. Indeed, our proof of Theorem 6 mirrors our steps in the analysis of our refutation algorithm. In fact, in a precise sense (as we explain in Chapter 9), our approach gives a tight connection between even covers in hypergraphs and simple cycles (and in turn, the spectral norm of the corresponding adjacency matrix) in the "Kikuchi graph" built from the hypergraph.

Chapter 9

A Proof of the Hypergraph Moore Bound

In this chapter, we prove Feige's conjecture (Theorem 6), that every k-uniform hypergraph with a sufficient number of hyperedges has a short even cover.

We begin by defining even (multi)covers.

Definition 9.0.1 (Even (multi)covers)**.** Let H be a k-uniform hypergraph on $[n]$. A set of *distinct* hyperedges $C_1, C_2, \ldots, C_r \in H$ is said to be an *even cover of length* r in H if every element $j \in [n]$ belongs to an even number of C_i's; equivalently, $\oplus_{i=1}^{r} C_i = \emptyset$. An even *multicover* in H is exactly the same except $C_1, C_2, \ldots, C_r \in H$ need not be distinct. Even (multi)covers are defined similarly for bipartite hypergraphs, using the hyperedges (u, C).

We note that if H is not simple, i.e., H is a multi-set, then H trivially has an even cover of length 2. Indeed, H must contain distinct elements C_1 and C_2 that are equal as sets, and so $C_1 \oplus C_2 = \emptyset$.

Analogous to the proof of Theorem 4.1.6 presented in Part I, we will first give a simple proof of Theorem 6 when k is even (Section 9.1), which will serve as a warmup to the full proof. Then, we will prove the full theorem in Section 9.2, which has a substantially more technical proof.

9.1 Proof of Theorem 6 for even k

Let H be a k-uniform hypergraph, and suppose that H has $m \geq \Gamma^k \cdot n \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \log_2 n$ hyperedges, where Γ is an absolute constant. In Section 2.2, we gave an algorithm to certify that, for *random* b_C's chosen in $\{-1, 1\}$ independently for each C, the polynomial $\phi(x) = \frac{1}{m} \sum_{C \in H} b_C x_C$ satisfies $\mathrm{val}(\phi) \leq 0.5$ (we can set 0.5 to be any constant < 1).

We will now observe that if we assume that H has no length $O(\ell \log n)$ even cover, then a near-identical proof implies that the same conclusion holds, namely that $\mathrm{val}(\phi) \leq 0.5$, *regardless of the choice of* b_C*'s*! This conclusion is absurd, as by setting $b_C = 1$ for all $C \in H$, we clearly $\mathrm{val}(\phi) = 1 > 0.5$. Hence, we conclude that H has a length $O(\ell \log n)$ even cover.

We will present the argument below by choosing $b_C = 1$ for all C, which suffices to prove Theorem 6. We can also view this argument as *spectral double counting* argument: the choice of $b_C = 1$ for all C yields a lower bound on $\|\tilde{A}\|_2$, where $\tilde{A}$ is the matrix from Section 2.2. We then upper bound $\|\tilde{A}\|_2$ using the fact that H has no length $O(\ell \log n)$ even cover.

Let us now present the full proof. We will assume familiarity with the notation and definitions from Section 2.2. Recall that by Eq. (2.2), we have that $\tilde{A}_2 \geq \frac{1}{2}\,\mathrm{val}(\phi) = \frac{1}{2}$. Hence, it suffices to

show that when H has no even cover, $\|\tilde{A}\|_2 < \frac{1}{2}$. To show this, we will use the *trace moment method*. Let $r = O(\ell \log n)$. Because $\tilde{A}$ is symmetric, we have that

$$\|\tilde{A}\|_2^{2r} \le \operatorname{tr}(A^{2r}) = \operatorname{tr}((W^{-1/2}AW^{-1/2})^{2r}) = \operatorname{tr}((W^{-1}A)^{2r}) .$$

We can view $W^{-1}A$ as the (weighted) adjacency matrix of a graph, and so the quantity $\operatorname{tr}((W^{-1}A)^{2r})$ counts weighted closed walks of length $2r$ in the graph.

The graph $W^{-1}A$ has an edge (S,T) with weight $\frac{1}{\Upsilon_S}$ if and only if $S \oplus T \in H$. Thus, a *closed* walk of length $2r$ is a sequence $S_0, S_1, \ldots, S_{2r}$ such that $S_0 = S_{2r}$ and $S_{i-1} \oplus S_i = C_i \in H$ for all $i \in [2r]$. It then follows that

$$\emptyset = S_0 \oplus S_{2r} = (S_0 \oplus S_1) \oplus (S_1 \oplus S_2) \oplus \cdots \oplus (S_{2r-1} \oplus S_{2r}) = C_1 \oplus \cdots \oplus C_{2r} .$$

The critical observation: closed walks are even multicovers. We can thus make the following critical observation: every closed walk in the Kikuchi graph $W^{-1}A$ corresponds to an even multicover in H — an even cover where a $C \in H$ may be used multiple times. Furthermore, because we assumed that H has no even cover of length $\le 2r = O(\ell \log n)$, it follows that any closed walk must correspond to a *trivial even multicover* — an even multicover where each hyperedge $C \in H$ appears an even number of times. Indeed, for any closed walk, we can consider its corresponding multicover, and try to "pair up" the hyperedges used in the closed walk. The set of "unpaired" hyperedges in the walk clearly forms an even multicover and has no repeated edges, i.e., it is an even cover, and it clearly has length $\le 2r$. But, the hypergraph H has no such even cover, and so there can be no "unpaired" hyperedges.

To finish the proof, we will bound the total weight of all walks that correspond to trivial even multicovers. We can encode the walk as follows.

(1) We choose the starting vertex $S_0 \in \binom{[n]}{\ell}$.

(2) We choose a bit $z_i \in \{0,1\}$ where $z_i = 0$ indicates that C_i will be a "new" hyperedge that is distinct from all $C_1, \ldots, C_{i-1}$, and $z_i = 1$ indicates that C_i will be an "old" hyperedge, i.e., it is one of $C_1, \ldots, C_{i-1}$. We note that, as argued above, we must have at least r "old" hyperedges.

(3) We construct the walk by choosing $C_1, \ldots, C_{2r}$ in order. On the i-th step, if $z_i = 0$ then we pick C_i from one of the neighbors of S_{i-1}. If $z_i = 1$, then we pick C_i from $C_1, \ldots, C_{i-1}$ and set $S_i = S_{i-1} \oplus C_i$.

Note that it is possible that some choices will yield an invalid walk, i.e., we try to set $S_i = S_{i-1} \oplus C_i$ but $|S_i| \neq \ell$. This is acceptable because we are overcounting the number of walks.

Let us now count the total weight of all walks. We pay $N \cdot 2^{2r}$ to choose S_0 and $z_1, \ldots, z_{2r}$. For a fixed choice of S_0 and $z_1, \ldots, z_{2r}$, we pay $\frac{1}{W_{S_{i-1}}} \cdot \Upsilon_{S_{i-1}}$ on the i-th step if $z_i = 0$, as S_{i-1} has $\Upsilon_{S_{i-1}}$ neighbors (recall this is the definition of Υ_S), and the edge has weight $\frac{1}{W_{S_{i-1}}}$. Note that $W_{S_{i-1}} \ge \Upsilon_{S_{i-1}}$, so this is at most 1. If $z_i = 1$, then we pay at most $\frac{r}{W_{S_{i-1}}}$, as there are at most r distinct hyperedges used in the entire walk. Because $W_{S_{i-1}} \ge mD/N$, it then follows that we pay at most $\frac{Nr}{mD}$ for each step, and we have set $z_i = 1$ for at most r choices of i. Hence, in total, we have

$$\operatorname{tr}((W^{-1}A)^{2r}) \le N 2^{2r} \left(\frac{Nr}{mD}\right)^r .$$

Recall that by Fact 3.6.1, $N/D \sim (n/\ell)^{k/2}$, and so by our choice of m, it follows that $\frac{Nr}{mD} \leq \varepsilon$, where ε is a small constant to be chosen later. Taking $2r$-th roots, we thus conclude that

$$\|\tilde{A}\|_2 \leq O(\sqrt{\varepsilon}).$$

Setting ε to be a sufficiently small constant then finishes the proof.

9.2 Proof of Theorem 6 for all k

We now prove Theorem 6 for all k, and in particular when k is odd. Our proof closely mimics the steps taken in Sections 5.2 to 5.4 on the way to obtaining an efficient refutation algorithm for semirandom sparse multilinear polynomials. In the first step, we observe that without loss of generality, we can assume that H is a simple, p-bipartite, (ϵ, ℓ)-regular hypergraph for $\epsilon = 1/4$.

Lemma 9.2.1 (Reduction to Simple, p-bipartite, $(1/4, \ell)$-regular hypergraphs)**.** *Fix $k, \ell = \ell(n) \in \mathbb{N}$ with $2(k-1) \leq \ell \leq n$. Suppose that for every p-bipartite, $(1/4, \ell)$-regular, simple k-uniform hypergraph $H = \{H_u\}_{u \in [p]}$ with $m \geq \max\{c^k \left(\frac{n}{\ell}\right)^{\frac{k-1}{2}} \sqrt{p\ell \log_2 n}, 16p\}$ hyperedges for some absolute constant c and $|H_u| = \frac{m}{p}$ for all u, there exists an even cover in H of length at most r. Then, every k-uniform hypergraph H with $m \geq \Gamma^k \cdot n \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \log_2 n$ hyperedges has an even cover of length at most r.*

Proof. Let H be an arbitrary k-uniform hypergraph. First, note that if H is not simple, we are immediately done since any pair of parallel hyperedges yields an even cover of size 2. We thus assume that H is simple. Apply the decomposition algorithm from Lemma 5.2.7 to H to get bipartite hypergraphs $H^{(1)}, \ldots, H^{(k)}$; these hypergraphs must be simple, as H was. As $\sum_{t=1}^{k} m^{(t)} = m$, there must exist some t with $1 \leq t \leq k$ such that $m^{(t)} \geq m/k$. As $m^{(1)} \leq \varepsilon m/k$ always holds, we must have $t \neq 1$. The bound on $m^{(t)}/p^{(t)}$ in Lemma 5.2.7 implies that $m^{(t)} \geq m/k \geq \max\{c^k \left(\frac{n}{\ell}\right)^{\frac{k-1}{2}} \sqrt{p^{(t)}\ell \log_2 n}, 16p^{(t)}\}$. Thus, the $p^{(t)}$-bipartite $(1/4, \ell)$-regular hypergraph $H^{(t)}$ must contain an even cover, say $(u_1, C_1), \ldots (u_{r'}, C_{r'})$ for some $r' \leq r$. From Lemma 5.2.7, for each u_i, there is a Q_i such that each hyperedge (u_i, C_i) in $H^{(t)}$ is a bipartite contraction of the unique hyperedge $(Q_i \cup C_i)$ in H. We then observe that $(Q_1 \cup C_1), \ldots, (Q_{r'} \cup C_{r'})$ is trivially an even cover of length $r' \leq r$ in H, which finishes the proof. □

This brings us to the crux of the argument presented in the following lemma.

Lemma 9.2.2 (No even covers implies refutation for semirandom polynomials on regular bipartite hypergraphs)**.** *Fix an odd $k \in \mathbb{N}$ and $\ell = \ell(n)$ with $2(k-1) \leq \ell \leq n$. Let $H = \{H_u\}_{u \in [p]}$ be a p-bipartite $(1/4, \ell)$-regular simple k-uniform hypergraph with $m \geq m_0 = \max(c^k \left(\frac{n}{\ell}\right)^{\frac{k-1}{2}} \sqrt{p\ell \log_2 n}, 16p)$ hyperedges, where c is an absolute constant, and $|H_u| = \frac{m}{p}$ for all u. Let ψ be the polynomial $\frac{1}{m} \sum_{u \in [p]} \sum_{C \in H_u} b_{u,C} y_u x_C$ for arbitrary $b_{u,C} \in \{-1, 1\}$. Suppose that H has no even covers of length $\leq O(\ell \log n)$. Then,* $\mathrm{val}(\psi) \leq 0.5$.

Observe that this lemma has an absurd conclusion. Clearly, if one sets $b_{u,C} = 1$ for all u, C, then $\mathrm{val}(\psi)$ is trivially 1: simply set $x = 1^n$ and $y = 1^p$. Thus, this lemma immediately gives a contradiction, in that H must admit an even cover of length $O(\ell \log n)$.

The reason we state the (somewhat absurd) lemma is because as we will see, our proof mimics our refutation argument from Section 5.4 and shows that we can essentially carry out all the

steps for *arbitrary* $b_{u,C}$'s as long as we can assume that H has no even covers of length $O(\ell \log n)$. Lemma 9.2.2 effectively captures this argument and, in our opinion, is the most enjoyable way to present it.

It is easy to finish the proof of Theorem 6 assuming the Lemma 9.2.2.

Proof of Theorem 6. By Lemma 9.2.1, we can assume that $H := \cup_{u\in[p]} H_u$ is a $(1/4, \ell)$-regular, simple, k-uniform bipartite hypergraph with $p \leq n^k$ partitions and $m \geq m_0$ hyperedges.

Suppose for the sake of contradiction that the hypergraph H has no even cover of length $O(\ell \log n)$. We set $b_{u,C} = 1$ for every u, C, and consider the polynomial $\psi = \frac{1}{|H'|} \sum_{u\in[p]} \sum_{C\in H_u} b_{u,C} y_u x_C$ in x, y. Observe that by setting $x = 1^n, y = 1^p$, we obtain that $\mathrm{val}(\psi) = 1$. On the other hand, applying Lemma 9.2.2 to ψ yields that $\mathrm{val}(\psi) \leq 0.5$. This is a contradiction, and so H must have an even cover of length $\leq O(\ell \log n)$. □

We now focus on the proof of Lemma 9.2.2.

9.2.1 Proof of Lemma 9.2.2

Our proof follows the exact same outline as in Section 5.4 for finding an efficient refutation algorithm for the polynomial ψ. One important difference is that in this section, we will use the argument to argue an upper bound on $\mathrm{val}(\psi)$; we do not care about finding an efficient certificate for a bound on $\mathrm{val}(\psi)$ here.

The key observation that we use in this proof is that there is exactly one step of the proof in Section 5.4 that uses the randomness of the coefficients $b_{u,C}$'s – namely, Lemma 5.4.7. Our proof in this section is exactly the same with the key innovation being an analog of Lemma 5.4.7 that works for *arbitrary* $b_{u,C}$'s as long as H has no $O(\ell \log n)$-length even cover. Indeed, as the hypergraph H satisfies the assumptions of Theorem 5.3.4, with this observation we immediately see that in order to finish the proof, it suffices to show that the spectral norm bounds in Lemma 5.4.7 still hold. In what follows, we use the exact same notation and conventions as in Section 5.4.

Let f be the polynomial obtained in Lemma 5.4.1 to the polynomial ψ. Let B be the pruned Kikuchi matrix (Definition 5.4.2 and Lemma 5.4.4) corresponding to the polynomial f. Using Lemma 5.4.3 (and the fact that it holds for B as well), we obtain that:

$$\mathrm{val}(\psi)^2 \leq \frac{1}{12} + \mathrm{val}(f) \leq \frac{1}{12} + \frac{p}{m^2 D} \|W^{-1/2} B W^{-1/2}\|_2 \cdot \mathrm{tr}(W)\,,$$

where we use that $12p \leq m$, and W is the matrix defined in Lemma 5.4.7.[1] Recall also that $D := \binom{k-1}{\frac{k-1}{2}}^2 \binom{2n-2(k-1)}{\ell-(k-1)}$ if k is odd and $2\binom{k-1}{\frac{k}{2}}\binom{k-1}{\frac{k-2}{2}}\binom{2n-2(k-1)}{\ell-(k-1)}$ if k is even.

Then, following the steps in the proof of Section 5.4.4, all that remains to be shown is the conclusion of Lemma 5.4.7 holds. In Section 5.4.4, we proved Lemma 5.4.7 by crucially exploiting the randomness of $b_{u,C}$'s. Here, the $b_{u,C}$'s are allowed to be *arbitrary*. We nonetheless show that the same conclusion holds if we additionally assume that H has no small even cover. Formally, we prove the following lemma.

[1] We note that this is the only other part where we deviate at all from the proof in Section 5.4; here, we now have $12p \leq m$ instead of $16p \leq m$ because we removed $4p$ edges; this is not important.

Lemma 9.2.3 (Spectral Norm of $W^{-1/2}BW^{-1/2}$ when H has no small even cover)**.** *Suppose that the $(1/4, \ell)$-regular p-bipartite simple k-uniform hypergraph H associated to the polynomial ψ has no even cover of length $\leq c_0 \ell \log_2 n$ for some large enough constant c_0. Then, we have*

$$\|W^{-1/2}BW^{-1/2}\|_2 \leq O\left(\sqrt{\frac{pN\ell \log n}{m^2 D}} + \Delta \frac{pN\ell \log n}{m^2 D}\right).$$

Lemma 9.2.3 finishes the proof of Lemma 9.2.2. Indeed, via the identical calculation in Section 5.4, it implies that $\|W^{-1/2}BW^{-1/2}\|_2 \cdot \mathrm{tr}(W) \leq \varepsilon^2 = \frac{1}{16}$, and thus $\mathrm{val}(\phi) \leq \frac{1}{12} + \frac{1}{16} \leq \frac{1}{3}$, so we are done.

It thus remains to prove Lemma 9.2.3.

Proof of Lemma 9.2.3. We will follow the proof of Lemma 5.4.7 that uses the trace method (Section 5.4.4). We know that $\|W^{-1/2}BW^{-1/2}\|_2 \leq \mathrm{tr}((W^{-1/2}BW^{-1/2})^{2r})^{1/2r} = \mathrm{tr}((W^{-1}B)^{2r})^{1/2r}$ for every $r \in \mathbb{N}$. We prove Lemma 9.2.3 by upper bounding $\mathrm{tr}((W^{-1}B)^{2r})$ for some $r = O(\ell \log_2 n)$.

We remind the reader that the trace moment method is classically used in analyzing the spectral norms of *random* matrices. In that setting, one bounds the *expectation* of $\mathrm{tr}((W^{-1}B)^{2r})$ which is analyzed by understanding the terms on the expansion on the right-hand side above that contribute a nonzero expectation often by utilizing inherent independence in the random variables appearing as entries of the matrix B. In contrast, there is *no randomness* in the matrix B, and so we are not bounding the expectation. Instead, we will analyze the "contributing" terms on the right-hand side by appealing to a crucial (and hitherto unobserved) property of the contributing walks in the Kikuchi matrix. We stress that the analysis appearing below does (as in fact any such analysis must!) strongly rely on the combinatorial structure of the support of the nonzero entries in our Kikuchi matrix B and cannot work for arbitrary matrices.

In fact, our key observation is to show that if H has no short even covers, then our upper bound on the *expectation* of $\mathrm{tr}((W^{-1}B)^{2r})$ in the semirandom setting (Proposition 5.4.14) still holds for $\mathrm{tr}((W^{-1}B)^{2r})$, i.e., when the $b_{u,C}$'s are *arbitrary*. Formally, we show the following.

Proposition 9.2.4. *Suppose that the $(1/4, \ell)$-regular p-bipartite simple k-uniform hypergraph H associated to the polynomial ψ has no even cover of length $\leq 4c_0 \ell \log_2 n$ for some large enough constant c_0. Then, for $r \leq c_0 \ell \log_2 n$, it holds that*

$$\mathrm{tr}((W^{-1}B)^{2r}) \leq \sum_{S \in \binom{2n}{\ell}} \sum_{\substack{\text{even walk sequences} \\ (u_1, C_1, C_1'), \ldots, (u_{2r}, C_{2r}, C_{2r}') \text{ for } S}} \mathrm{wt}(S, (u_1, C_1, C_1'), \ldots, (u_{2r}, C_{2r}, C_{2r}')).$$

We note (at the cost of repetition) that Proposition 9.2.4 holds *regardless* of the $b_{u,C}$'s and is a consequence of the combinatorial structure of the support of Kikuchi matrices.

We now finish the proof of Lemma 9.2.3 assuming Proposition 9.2.4. This is immediate given the calculations in Section 5.4.4. By Lemma 5.4.15, we know that for each S, the total weight of such sequences is at most $(4r)^r \left(\frac{pN}{m^2 D}\right)^{2r} \left(\frac{2m^2 D}{pN} + r\Delta^2\right)^r$. Hence,

$$\|W^{-1/2}BW^{-1/2}\|_2^{2r} \leq \mathrm{tr}((W^{-1}B)^{2r}) \leq N(4r)^r \left(\frac{pN}{m^2 D}\right)^{2r} \left(\frac{2m^2 D}{pN} + r\Delta^2\right)^r.$$

Setting $r = c_0 \ell \log_2 n$ for c_0 a sufficiently large constant, the above implies that

$$\|W^{-1/2} B W^{-1/2}\|_2 \leq O\left(\sqrt{\frac{pNr}{m^2 D}} + r\Delta \frac{pN}{m^2 D}\right),$$

assuming that H has no even cover of length $\leq 4r = 4c_0 \ell \log_2 n$. This finishes the proof, up to Proposition 9.2.4. □

Proof of Proposition 9.2.4. We compute:

$$\mathrm{tr}((W^{-1}B)^{2r}) = \sum_{(u_1,S_1),\ldots,(u_{2r},S_{2r})} \mathbb{E}\left[\prod_{h=1}^{2r} \frac{1}{W_{S_{h-1}}} B_{u_h}(S_{h-1}, S_h)\right], \tag{9.1}$$

where we use the convention that $u_{2r+1} := u_1$ and $S_0 := S_{2r}$.

Observe that each term in (9.1), ignoring the weights from W, can contribute a value at most 1 since all $b_{u,C}$'s are $\{\pm 1\}$ and H is simple. Thus, the RHS of (9.1) is upper bounded by the total weight of nonzero "walk" terms, i.e., the total weight of the terms in the sum in (9.1).

The central observation is the following lemma that observes a combinatorial property of nonzero terms on the RHS in (9.1).

Claim 9.2.5 (nonzero terms are even multicovers). If the walk term corresponding to $(u_1, S_1, u_2, S_2, \ldots, u_{2r}, S_{2r})$ is nonzero, then for every $h \in [2r]$, there exist $C_h \neq C'_h \in H_{u_h}$ such that $S_{h+1} = S_h \oplus C_h^{(1)} \oplus C_h'^{(2)}$. Moreover, $\bigoplus_{h \leq 2r} (u_h, C_h) \oplus (u_h, C'_h) = \emptyset$, i.e., $\{(u_h, C_h), (u_h, C'_h)\}_{h \leq 2r}$ is an even multicover in H.

Proof. By definition of the Kikuchi matrix, the (unweighted) walk term equals

$$\prod_{h \leq 2r} B_{u_h}(S_{h-1}, S_h) \leq \prod_{h \leq 2r} \mathbf{1}(S_{h-1} \overset{C_h^{(1)}, C_h'^{(2)}}{\longleftrightarrow} S_h),$$

where for each h, $C_h, C'_h \in H_{u_h}$. Here, the inequality holds because B_u's are the pruned matrices (Lemma 5.4.4); we have equality if we used the A_u's instead and included the coefficients b_{u_h, C_h} and b_{u_h, C'_h}.

Clearly, if the term corresponding to $(u_1, S_1, u_2, S_2, \ldots, u_{2r}, S_{2r})$ is nonzero then $\mathbf{1}(S_{h-1} \overset{C_h^{(1)}, C_h'^{(2)}}{\longleftrightarrow} S_h) = 1$ for every $h \leq 2r$. Expanding the definition, this implies that $S_h = S_{h-1} \oplus C_h^{(1)} \oplus C_h'^{(2)}$.

To show the "moreover", we observe that by adding up all the aforementioned two equations, we obtain:

$$\bigoplus_{h=1}^{2r} S_h = \bigoplus_{h=0}^{2r-1} S_h \oplus \bigoplus_{h=1}^{2r} C_h^{(1)} \oplus C_h'^{(2)}.$$

As $S_0 := S_{2r}$, canceling the S_h's on both sides yields $\bigoplus_{h \leq 2r} C_h^{(1)} \oplus C_h'^{(2)} = \emptyset$. This then trivially implies that $\bigoplus_{h \leq 2r} C_h = \bigoplus_{h \leq 2r} C'_h = \emptyset$, and hence $\bigoplus_{h \leq 2r} (u_h, C_h) \oplus (u_h, C'_h) = \emptyset$, as $(u_h, C_h) \oplus (u_h, C'_h) = C_h \oplus C'_h$. □

Observe that the even multicover $\{(u_h, C_h), (u_h, C'_h)\}_{h \leq 2r}$ in Claim 9.2.5 need not be an even cover as the (u_h, C_h)'s need not be distinct. Indeed, the main punch of what follows is that when there are no small even covers in H, then the (u_h, C_h)'s must occur in pairs, i.e., each (u_h, C_h) appears an even number of times in the two multicovers obtained in Claim 9.2.5.

Claim 9.2.6 (No short even cover implies short multicovers are unions of pairs). Suppose $H = \{H_u\}_{u\in[p]}$ has no even cover of length $\leq 4r$. Then, if the walk term in (9.1) corresponding to $\{u_h, S_h, C_h, C'_h\}_{h\leq 2r}$ is nonzero, then each $(u, C) \in \cup_{u\in[p]} H_u$ occurs an even number of times in the multiset $\{(u_h, C_h), (u_h, C'_h)\}_{h\leq 2r}$. In particular, $\{(u_h, C_h, C'_h)\}_{h\leq 2r}$ is an *even walk sequence for* S_0, as defined in Definition 5.4.13.

Proof. From Claim 9.2.5, $\bigoplus_{h=1}^{2r} (u_h, C_h) \oplus (u_h, C'_h) = \emptyset$. Start from the multiset $\{(u_h, C_h), (u_h, C'_h)\}_{h\leq 2r}$, and remove pairs greedily until this is no longer possible. Observe that the symmetric difference of the resulting set must also be empty since we removed sets in equal pairs. If at the end of this process, we are left with a nonzero number of hyperedges, i.e., we assume that the conclusion does not hold, then we have at most $4r$ distinct hyperedges whose symmetric difference is empty. Thus, the remaining set must be an even cover of length $\leq 4r$ in H, which is a contradiction. □

Combining Claims 9.2.5 and 9.2.6, we thus see that the RHS of (9.1) is upper bounded by

$$\sum_{S} \sum_{\substack{\text{even walk sequences} \\ (u_1,C_1,C'_1),\ldots,(u_{2r},C_{2r},C'_{2r}) \text{ for } S}} \mathrm{wt}(S, (u_1, C_1, C'_1), \ldots, (u_{2r}, C_{2r}, C'_{2r}))\,,$$

which finishes the proof of Proposition 9.2.4. □

Part III

Lower Bounds for Locally Decodable and Correctable Codes

Chapter 10

Background and Results

A *locally decodable code* (LDC) is an error-correcting code that admits a local decoding algorithm that can recover any symbol of the original message by reading only a small number of randomly chosen symbols from the received corrupted codeword. A *locally correctable code* (LCC) is a closely-related notion: a code is locally correctable if it admits a *local* correction (or *self correction*) algorithm that can recover any symbol of the original *codeword* by reading only a small number of randomly chosen symbols from the received corrupted codeword. More formally, we say that a code $C\colon \{0,1\}^k \to \{0,1\}^n$ is q-locally decodable (correctable) if for any codeword x, a corruption y of x, and input $i \in [k]$ ($u \in [n]$), the local decoding (correction) algorithm reads at most q symbols (typically a small constant such as 2 or 3) of y and recovers the bit b_i (x_u) with probability $1/2 + \varepsilon$ whenever $\Delta(x, y) \coloneqq |\{v \in [n] : x_v \neq y_v\}| \leq \delta n$, where δ, the "distance" of the code, and ε, the decoding accuracy, are constants. Local correction is known to be the stronger notion: any LCC can be turned into an LDC with only a small loss in parameters.[1] The central two questions in the study of LDCs/LCCs are to (1) determine the smallest possible blocklength n as a function of the message length k for a fixed number of queries q, and (2) determine the relationship between local decoding and correction, i.e., determine if LDCs and LCCs are equivalent, or if LDCs are strictly weaker than LCCs.

Though formalized later in [KT00], local decoding/correction was first introduced for *program checking* [BK95], and early applications utilized that Reed–Muller codes are locally correctable via polynomial interpolation. Since then, LDCs/LCCs have been a mainstay in complexity and algorithmic coding theory with a long array of applications. An abridged list (the surveys [Tre04, Yek12, Dvi12] provide details) of applications includes sublinear algorithms and property testing [RS96, BLR93], probabilistically checkable proofs [ALM+98, AS98], IP=PSPACE [LFKN90, Sha90], worst-case to average-case reductions [BFNW93], constructions of explicit rigid matrices [Dvi10], derandomization [DS05], data structures [Wol09, CGW10], fault-tolerant computation [Rom06], secure multiparty computation [IK04], and t-private information retrieval protocols [IK99, BIW10]. The existence of LCCs turns out to have natural connections to incidence geometry [Dvi12], additive combinatorics [BDL13], and the theory of block designs [BIW10].

For any constant $q \in \mathbb{N}$, Reed–Muller codes (i.e., evaluations of $(q-1)$-degree polynomials)

[1] See Fact 3.3.8.

yield binary, linear[2] q-LCCs (and therefore also q-LDCs) with a blocklength $n \leq 2^{O(k^{\frac{1}{q-1}})}$. Given their extensive applications and connections, finding LDCs/LCCs of smaller blocklength has been a major project in theoretical computer science over the past three decades with some remarkable successes over the years. For example, *multiplicity codes* [KSY14] significantly beat the blocklength of Reed–Muller codes in the *super-constant* query regime. In the constant-query regime, *matching vector codes* [Efr09, Yek08] give a construction of linear 3-LDCs with a strictly subexponential (i.e., $n \leq \exp(\exp(O(\sqrt{\log k \log\log k})))$ blocklength; it is not known whether or not these codes are locally correctable. To sidestep the difficulty of finding more efficient LDCs/LCCs, the work of [BGH+04] introduced *relaxed LCCs* that soften the local correction property and has seen exciting recent developments [GRR20, AS21, CGS20, KM24c, CY23]. These successes notwithstanding, constructing better constant-query LDCs/LCCs has remained a major open question (see, e.g., Chapter 8 in [Yek12]), and for constant q, Reed–Muller codes remain the best-known construction of q-LCCs.

LDC and LCC lower bounds. The lack of progress on finding better constant-query LCCs has motivated a long-investigated conjecture that Reed–Muller codes might be *optimal* constant query LCCs. The work of [KW04, GKST06] essentially confirmed this conjecture for the "base case" of $q = 2$ by proving that $n \geq 2^{\Omega(k)}$ for any two-query LDC. This matches the construction of Hadamard codes, which are 2-LCCs (and therefore also 2-LDCs) with $n = 2^k$. For $q \geq 3$, however, the best-known lower bounds for LDCs/LCCs are substantially weaker, and in particular are only a (small) polynomial in k: the works of [KW04, Woo07] prove that q-LDCs (and therefore also q-LCCs) must have $n \geq \tilde{\Omega}(k^{1/(1-1/\lceil \frac{q}{2} \rceil)})$, which is $n \geq \Omega(\tilde{k}^2)$ in the case of $q = 3$.[3]

Limitations of prior lower bound techniques for LCCs. Beyond the weakness in the quantitative results, all the above lower bounds, when applied to LCCs, suffer from an important inherent limitation — they also hold even for the weaker setting of locally decodable codes (LDCs). As we mentioned above, there are subexponential length (and thus substantially beating Reed–Muller) 3-query binary, linear codes that are locally decodable [Yek08, Efr09]. Indeed, characterizing the limitations of prior proof techniques and finding methods that could separate LCCs and LDCs has itself been a major research goal. For example, Dvir, Gopi, Gu and Wigderson [DGGW19] formalize the limitations of prior lower bound techniques for LCCs by showing that the "random restriction" approach in [KT00] applies to a more general setting of "spanoids" where they are, in fact, tight. On the other hand, to show a strong separation between LCCs and LDCs, Barkol, Ishai and Weinreb [BIW10] build an approach for stronger LCC lower bounds via connections to the well-studied Hamada conjecture ([Ham73], see lecture notes [Ton11]) and its generalizations in the theory of block designs, while Dvir, Saraf and Wigderson [DSW14] develop new geometric techniques to prove a slightly superquadratic lower bound for an appropriate formulation of 3-LCCs over the reals.

Connections to the Hamada Conjecture. As mentioned above, locally correctable codes have a deep connection — first formalized by Barkol, Ishai and Weinreb [BIW10] — to the widely open Hamada conjecture from the 1970s in combinatorial design theory (with deep connections

[2] A code is *linear* over a field $\mathbb{F}$ if the encoding map C is an $\mathbb{F}$-linear map.

[3] These lower bounds all hold for non-linear codes over small (i.e., polylog(k)) size alphabets. A weaker polynomial lower bound [KT00, IS18] is known to hold for linear codes over all fields and for the specific case of $q = 3$, [Woo10] shows a lower bound of $\Omega(k^2)$ for linear 3-LDCs over all fields.

to coding theory, see [AK92] for a classical reference). For positive integers m, s, λ, a 2-(m, s, λ)-design is a collection $\mathcal{B} \subseteq [m]$ of subsets (called *blocks*) of size s, such that every pair of elements in $[m]$ appears in exactly λ subsets in $\mathcal{B}$. For any prime p, the p-rank of a design $\mathcal{B}$ is the rank, over $\mathbb{F}_p$, of the *incidence* matrix of $\mathcal{B}$: the 0-1 matrix with rows labeled by elements of $[m]$, columns labeled by elements of $\mathcal{B}$ and an entry (i, B) is 1 iff B contains i. A central question in algebraic design theory is understanding the smallest possible p-rank of a 2-(m, s, λ)-design.

In [BIW10], the authors showed that given any 2-(m, s, λ)-design $\mathcal{D}$ of p-rank $m - k$, the dual subspace to the column space of the incidence matrix of $\mathcal{D}$ yields a linear $(s-1)$-query locally correctable code on $\mathbb{F}_p^m$ of dimension k. In particular, applying this transformation to the well-studied *geometric designs* yields the folklore construction of Reed–Muller locally correctable codes discussed earlier. Specifically, the 3-query locally correctable *binary* code obtained from Reed–Muller codes on $\mathbb{F}_4$ corresponds to a 2-$(n, 4, 1)$-design over $\mathbb{F}_2$ (see Section 12.11).

In 1973, Hamada [Ham73] made a foundational conjecture (see [Jun11] for a recent survey) in the area that states[4] that affine geometric designs (i.e., duals to the Reed–Muller LCCs) minimize the p-rank among all algebraic designs of the same parameters. Over the past few decades, the conjecture has been confirmed in various special cases [HO75, DHV78, Tei80, Ton99] that all correspond to $s \leq 3$ or $s = n - 1$. In particular, the case of $s = 4$ (the setting of 3-LCCs) is widely open. The connection between Hamada's conjecture and LCC lower bounds was suggested in [BIW10] as evidence for *the difficulty* of proving LCC lower bounds.

Summary. To summarize, for 2-LDCs/LCCs, the best-known construction is the Hadamard code, which achieves $n = 2^k$ and is a linear code, and the matching lower bound of $n \geq 2^{\Omega(k)}$ shown in [KW04, GKST06] proves that this is optimal. For 3-LCCs, the best-known construction is the Reed–Muller code, which achieves $n = 2^{2\sqrt{2k}}$ and additionally is a *block design* (and hence also linear). For 3-LDCs, the best construction comes from the matching vector codes of [Yek08, Efr09], which achieve $n = 2^{2^{\sqrt{O(\log k \log\log k)}}}$ and are linear. For either 3-LDCs/LCCs, the best lower bound is $n \geq \tilde{\Omega}(k^2)$ [KW04].

More generally, there is an exponential gap between best-known constructions and lower bounds for q-LCCs for $q \geq 3$ and a subexponential gap for q-LDCs. Furthermore, the best known lower bound techniques for q-LCCs apply also to q-LDCs and thus provably cannot yield an exponential lower bound, and showing better lower bounds for q-LCCs would yield better bounds for the Hamada conjecture for block designs.

10.1 Our results

In this thesis, we prove a near-cubic lower bound of $n \geq k^3/\text{polylog}(k)$ for 3-LDCs and an exponential lower bound for 3-LCCs. Because there exist 3-LDCs of subexponential length, this gives the first *separation* between 3-LCCs and 3-LDCs. No such separation was known for q-LDCs and q-LCCs for any constant $q \geq 3$.[5] For 3-LCCs, our lower bounds are (1) $n \geq 2^{\Omega(k^{1/5})}$ for

[4] Hamada's original conjecture is that affine geometric designs, or, dual codes to Reed–Muller codes, are the unique optimal designs with the same parameters. This strong form has since then been disproved – there are non-affine geometric designs that achieve the *same* (but not better!) parameters [Jun84, Kan94, LLT00, LLT01, LT02, JT09]. The version of the problem we study here is called the *weak version* of Hamada conjecture.

[5] The work of [BGT17] shows a separation between 2-LCCs and 2-LDCs over poly(n)-sized alphabets. For 2-LCCs on small alphabets, a strong separation cannot exist. For example, on $\mathbb{F}_2$, the Hadamard code gives both an essentially

general nonlinear 3-LCCs with "high completeness", (2) $n \geq 2^{\Omega(k^{1/4})}$ for linear 3-LCCs, and (3) $n \geq 2^{(1-o(1))\sqrt{k}}$ for 3-LCCs that are designs. Because Reed–Muller codes give design 3-LCCs of length $n \leq 2^{\sqrt{8k}}$ (see Section 12.11), this last result is tight up to a factor of $\sqrt{8}$ in the exponent, and additionally proves Hamada's conjecture for 2-$(n, 4, 1)$-designs up to a factor of 8 in the codimension.

Our main tool is a new connection between the existence of locally decodable/correctable codes and refutation of instances of Boolean CSPs with limited randomness. This connection is similar in spirit to the connection between PCPs and hardness of approximation for CSPs, in which one produces a q-ary CSP from a PCP with a q-query verifier by adding, for each possible query set of the verifier, a local constraint that asserts that the verifier accepts when it queries this particular set. To refute the resulting CSP instance, our proof builds on the spectral analysis of *Kikuchi matrices* employed in the work of [GKM22] (and the refined argument in [HKM23]), which obtained strong refutation algorithms for semirandom and smoothed CSPs and proved the hypergraph Moore bound conjectured by Feige [Fei08] up to a single logarithmic factor.

10.1.1 A near-cubic lower bound for 3-LDCs

In our first result, we show a near-cubic lower bound $n \geq k^3/\mathrm{polylog}(k)$ on the blocklength of any 3-query LDC. This improves on the previous best lower bound by a $\tilde{O}(k)$ factor. More precisely, we prove:

Theorem 7. *Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a code that is $(3, \delta, \varepsilon)$-locally decodable. Then, it must hold that $k^3 \leq n \cdot O((\log^6 n)/\varepsilon^{32}\delta^{16})$. In particular, if δ, ε are constants, then $n \geq \Omega(k^3/\log^6 k)$.*

We have not attempted to optimize the dependence on ε and δ in Theorem 7; for the specific case of binary *linear* codes, one can obtain slightly better dependencies on $\log k, \varepsilon, \delta$, as we show in Theorem 11.3.2 and Corollary 11.3.3. It is straightforward to extend Theorem 7 to nonbinary alphabets with a polynomial loss in the alphabet size, and we do so in Theorem 11.2.2 in Section 11.2. Finally, using known relationships between locally correctable codes (LCCs) and LDCs (Fact 3.3.8), Theorem 7 implies a similar lower bound for 3-query LCCs.

Up to $\mathrm{polylog}(k)$ factors, the best known lower bound of $n \geq k^{\frac{q+1}{q-1}}/\mathrm{polylog}(k)$ for q-LDCs for odd q can be obtained by simply observing that a q-LDC is also a $(q+1)$-LDC, and then invoking the lower bound for $(q+1)$-query LDCs. Our improvement for $q = 3$ thus comes from obtaining the same tradeoff with q as in the case of even q, but now for $q = 3$. Our proof does not extend to odd $q \geq 5$; we briefly mention at the end of the proof overview in Chapter 11 the place in the proof where the natural generalization fails. We leave proving a lower bound of $n \geq k^{\frac{q}{q-2}}/\mathrm{polylog}(k)$ for all *odd* $q \geq 5$ as an intriguing open problem, which we discuss in more detail in Section 15.1.

10.1.2 Exponential lower bounds for 3-LCCs

In our second set of results, we prove an *exponential* lower bound for 3-LCCs. Our lower bounds vary slightly depending on whether one is willing to assume that the LCC is linear, or a block design. We note that the best-known constructions of LCCs and LDCs, namely Reed–Muller

optimal 2-LCC and 2-LDC.

codes and matching vector codes, are $\mathbb{F}_2$-linear, and the best-known construction of 3-LCCs from Reed–Muller codes is a block design (see Section 12.11).

Linear 3-LCCs. In our first result, we prove a lower bound of $n \geq 2^{\Omega(k^{1/4})}$ for linear 3-LCCs.

Theorem 8. *Let $\mathcal{L}\colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(3, \delta, \varepsilon)$-LCC. Then, $n \geq 2^{\Omega((\delta^2 k/(|\mathbb{F}|-1)^2)^{1/4})}$. In particular, if $\mathcal{L}\colon \mathbb{F}_2^k \to \mathbb{F}_2^n$ is a $(3, \delta, \varepsilon)$-LCC where δ is constant, then $n \geq 2^{\Omega(k^{1/4})}$.*

Theorem 8 first appeared in [KM24a] with an exponent of $1/8$. This was subsequently improved by [Yan24] to $1/4$ by optimizing the technical "row pruning" step of the proof. In this thesis, we incorporate the improvements of [Yan24] into our original proof to give the stronger theorem and the simpler proof.

As stated earlier, Theorem 8 also yields the first *separation* between (linear) 3-LCCs and 3-LDCs. In particular, Theorem 8 implies that matching vector codes that yield linear 3-LDCs over $\mathbb{F}_2$ of subexponential blocklength, such as the codes in [Yek08, Efr09], *cannot* admit a local correction algorithm, answering a question of Yekhanin (see Chapter 8 in [Yek12]).

Design 3-LCCs. In our second result, we prove a lower bound that is sharp up to a $\sqrt{8}$ factor in the exponent on the blocklength of any binary linear 3-LCC where the local correction query sets form a 2-$(n, 4, 1)$-design. This is equivalent to asking for the hypergraph of local correction sets H_u for correcting any bit x_u of the codeword to be a *perfect* 3-uniform hypergraph matching and that every pair of codewords bits appears in exactly 2 triples across all matchings.[6] Specifically, for such design 3-LCCs, we prove:

Theorem 9. *Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a design 3-LCC. Then, $n \geq 2^{(1-o(1))\sqrt{k}}$. Here, the $o(1)$-factor is $O(\log k/\sqrt{k})$.*

Reed–Muller codes, in particular, are design LCCs. In Section 12.11 we observe that the folklore best-known construction of binary 3-query LCCs — obtained by projecting Reed–Muller codes of degree-2 polynomials over $\mathbb{F}_4$ to $\mathbb{F}_2$ via the trace map — is a design 3-LCC with $n \leq 2^{\sqrt{8k}}$, or equivalently, a 2-$(n, 4, 1)$ design of rank $n - k$. Thus, the bound in Theorem 9 is *tight up to a factor of $\sqrt{8}$ in the exponent.* As a direct corollary, we also confirm the Hamada conjecture for 2-$(n, 4, 1)$-designs up to a factor of 8 in the co-dimension.

Nonlinear 3-LCCs. In final second result, we obtain improved lower bounds for smooth 3-LCCs with high completeness. These codes may be non-linear and may have adaptive correction algorithms.

A 3-LCC is said to be *δ-smooth* if no codeword bit is queried with probability more than $\frac{1}{\delta n}$ on any particular invocation of the decoder. Introduced by Katz and Trevisan [KT00], smooth codes provide a clean formalization of general locally correctable/decodable codes. We say that such a code has completeness $1 - \epsilon$, if, when running the δ-smooth local correction algorithm on an *uncorrupted* codeword, the algorithm succeeds with probability at least $1 - \epsilon$. Recall that the usual notion of completeness (e.g., in [KT00]) for LCCs is for an input with a δ-fraction of corruptions.

[6] The reason that this is 2 instead of $\lambda = 1$ is because a 4-tuple (u, v, s, t) yields 2 decoding triples, (u, s, t) for v and (v, s, t) for u, that contain the pair (s, t).

Our result shows that for any $(1-\epsilon)$-complete δ-smooth code where δ is a constant, $n \geq k^{O(1/\epsilon)}$. In particular, when $\epsilon \leq 1/\mathrm{polylog}(n)$, we obtain a exponential lower bound on the block length, and as ε approaches 0 the bound becomes $n \geq 2^{\Omega(k^{1/5})}$.

Theorem 10. *There is an absolute constant $\gamma > 0$ such that the following holds. Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a δ-smooth (possibly non-linear and adaptive) 3-LCC with completeness $1-\varepsilon$. Then for any $\eta \in (0,1)$, it holds that $k \leq \frac{\log(1/\delta)}{\eta^4\delta^3} \cdot O(n^{\frac{1}{r}} \log^5 n)$, where $r = \lfloor \frac{1-\eta}{2\varepsilon} \rfloor$.*

In particular, if δ is a constant and $\varepsilon = 0$, then $k \leq O(\log^5 n)$, i.e., $n \geq 2^{\Omega(k^{1/5})}$, and if $\varepsilon > 0$ is a small constant and $1/(2\varepsilon)$ is not an integer, then taking $\eta = 1/\log n$ implies that $k \leq \tilde{O}(n^{2\varepsilon})$, i.e., $k \geq \tilde{\Omega}(k^{\frac{1}{2\varepsilon}})$.

As we shall discuss towards at the end of this section, Theorem 10 implies a lower bound for general $(3,\delta,\varepsilon)$-LCCs that beats the best-known $n \geq \tilde{\Omega}(k^3)$ lower bound (Theorem 7, [AGKM23]) by a polynomial factor when ε is a small constant. Moreover, in the case of near-perfect completeness, our result above obtains the first exponential lower bound for (possibly adaptive and non-linear) smooth 3-LCCs.

Smooth vs. general LCCs. Smooth LCCs (Definition 3.3.6) were defined in the work of [KT00], motivated by their connection to general LCCs (Definition 3.3.5). A simple reduction in [KT00] shows that any $(3,\delta,1-\varepsilon)$-LCC, i.e., an LCC with distance δ and completeness $1-\varepsilon$, can be turned into a $(3,\delta/3,1-\varepsilon)$-smooth LCC, i.e., a $\delta/3$-smooth 3-LCC with completeness $1-\varepsilon$. Conversely, any $(3,\delta,1-\varepsilon)$-smooth LCC is a $(3,\eta\delta,1-\varepsilon-\eta)$-LCC for any $\eta > 0$ (see Remark 3.3.7).

Thus, when ε is a small constant, Theorem 10 implies a lower bound for general $(3,\delta,1-\varepsilon)$-LCCs that beats the prior best $n \geq \tilde{\Omega}(k^3)$ lower bound (Theorem 7, [AGKM23]) by a polynomial factor.

However, in the setting of perfect completeness (and $\varepsilon = o(1)$ more generally), the comparison between smooth LCCs and general LCCs begins to break down. This is because, for a general LCC, δ is the fraction of errors one can tolerate while still decoding correctly with probability $1-\varepsilon$; the parameters δ and ε are coupled! In particular, it is likely not possible to simultaneously have $\varepsilon = o(1)$, $\delta = O(1)$ and $q = O(1)$. On the other hand, for a smooth LCC, δ is the smoothness parameter, and $1-\varepsilon$ is the probability that the decoder succeeds *on an uncorrupted codeword*. Thus, for smooth codes, it is perfectly sensible to set $\delta = O(1)$, $\varepsilon = 0$, and $q = O(1)$.

In retrospect, the definition of LCCs inherently couples δ and the completeness ε, whereas for smooth codes these parameters become independent. In particular, a smooth code allows us to seamlessly trade off between the fraction of errors $\eta\delta$ tolerated and the success probability $1-\varepsilon-\eta$ of the decoder in the presence of this fraction of errors. For this reason, a smooth code is a stronger object, but also perhaps a more natural one.

Indeed, in some important applications of LDCs/LCCs, smooth LDCs/LCCs are the right notion to consider. For example, a *perfectly smooth* $(q,1,1-\varepsilon)$-smooth LDC gives a q-server information-theoretically secure private information retrieval scheme with completeness $1-\varepsilon$.

The subtle definitional issues above did not affect prior lower bound (or upper bound) techniques. Indeed, known constructions of q-LDCs and LCCs are *perfectly* smooth and satisfy *perfect* completeness, i.e., $(q,1,1)$-smooth LDCs/LCCs, and the lower bound techniques of [KT00, KW04, AGKM23] succeed for smooth LDCs/LCCs even with *low* completeness.

Concurrent work. In concurrent work, [AG24] builds on [KM24a, Yan24] and proves an $n \geq 2^{\Omega(\sqrt{k/\log k})}$ lower bound for all linear 3-LCCs over $\mathbb{F}_2$, improving on the $2^{\Omega(k^{1/4})}$ shown in [Yan24], and their result can be extended to linear 3-LCCs over any small field $\mathbb{F}$ of characteristic 2. This is

incomparable to Theorem 8, as the lower bound is stronger but it requires that the field $\mathbb{F}$ has characteristic 2. It is also incomparable to Theorem 9, as it proves a weaker (and possibly not tight) lower bound, as compared to the sharp statement in Theorem 9, but it applies for all linear 3-LCCs over $\mathbb{F}_2$, not just design 3-LCCs. The work of [AG24] does not prove any lower bound for nonlinear codes.

Chapter 11

A Near-Cubic Lower Bound for 3-Query Locally Decodable Codes

In this chapter, we will prove Theorem 7, our improved lower bound for 3-LDCs.

Setup. We follow the proof strategy and setup that we introduced in Section 2.3. Namely, we define a 3-XOR instance corresponding to the normal LDC decoder. By Definition 3.3.2, the 3-XOR instance we define has a high value, i.e., there is an assignment to the variables satisfying a nontrivial fraction of the constraints. To finish the proof, we show that if $n \ll k^3$, then the 3-XOR instance must have small value, which is a contradiction.

By Fact 3.3.3, in order to show that $k^3 \leq n \cdot \frac{O(\log^6 n)}{\varepsilon^{32}\delta^{16}}$, it suffices for us to show that for any code $C\colon \{-1,1\}^k \to \{-1,1\}^n$ that is $(3,\delta,\varepsilon)$-normally decodable, it holds that $k^3 \leq n \cdot \frac{O(\log^6 n)}{\varepsilon^{16}\delta^{16}}$. As C is $(3,\delta,\varepsilon)$-normally decodable, this implies that there are 3-uniform hypergraph matchings $H_1,\ldots,H_k$ satisfying the property in Definition 3.3.2. Let $m := \sum_{i=1}^{k}|H_i|$ be the total number of hyperedges in the hypergraph $H := \cup_{i=1}^{k} H_i$.

We define the relevant family of 3-XOR instances below.

The Key 3-XOR Instances

For each $b \in \{-1,1\}^k$, we define the 3-XOR instance Ψ_b, where:

(1) The variables are $x_1,\ldots,x_n \in \{-1,1\}$,

(2) The constraints are, for each $i \in [k]$ and $C \in H_i$, $\prod_{v\in C} x_v = b_i$.

We associate an instance Ψ_b with the polynomial $\Psi_b(x) := \frac{1}{m}\sum_{i=1}^{k} b_i \sum_{C\in H_i}\prod_{v\in C} x_v$, and define $\mathrm{val}(\Psi_b) := \max_{x\in\{-1,1\}^n}\Psi_b(x)$. We note that the maximum fraction of constraints in the 3-XOR instance Ψ_b satisfied by any assignment x is $\frac{1}{2}+\frac{1}{2}\mathrm{val}(\Psi_b)$.

We first observe that Definition 3.3.2 immediately implies that every 3-XOR instance Ψ_b in the above family (indexed by $b \in \{-1,1\}^k$) must have a non-trivially large value. Formally, we have that

$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Psi_b)] \geq \mathbb{E}_{b\leftarrow\{-1,1\}^k}[\Psi_b(C(b))] \geq 2\varepsilon \ , \tag{11.1}$$

where the first inequality is by definition of $\mathrm{val}(\cdot)$, and the second inequality uses Definition 3.3.2, as for each constraint $C \in H_i$ for some i, the encoding $C(b)$ of b satisfies this constraint with probability $\frac{1}{2}+\varepsilon$ for a random b.

Overview: refuting the XOR instances. To finish the proof, it thus suffices to argue that $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(\Psi_b)]$ is small. We would like to do this using Kikuchi matrices, similar to those defined in Definition 2.3.2. However, when $q = 3$, or more generally when q is odd, the matrices A_i defined in Definition 2.3.2 are no longer meaningful, as the condition $S \oplus T = C$ is never satisfied. A naive attempt to salvage the above approach is to simply allow the columns of A_i to be indexed by sets of size $\ell + 1$, rather than ℓ. However, this asymmetry in the matrix causes the spectral certificate to obtain a worse dependence in terms of q, leading to a final bound of $k \leq n^{1-2/(q+1)}O(\log n)$, the same as the current state-of-the-art lower bound for odd q. This is precisely the issue that in general makes refuting q-XOR instances for odd q technically more challenging than even q. The asymmetric matrix effectively pretends that q is $q+1$, and thus obtains the "wrong" dependence on q.

Instead, our main idea is to transform a 3-LDC into a 4-XOR instance and then use an appropriate Kikuchi matrix to find a refutation for the resulting 4-XOR instance. The transformation works as follows. We randomly partition $[k]$ into two sets, L, R, and fix $b_j = 1$ for all $j \in R$. Then, for each *intersecting pair* of constraints C_i, C_j that intersect with $C_i \in H_i, i \in L$, $C_j \in H_j, j \in R$, we add the derived constraint $C_i \oplus C_j$ to our new 4-XOR instance, with right-hand side b_i.[1] Because the 3-XOR instance has high value, by the Cauchy-Schwarz inequality the 4-XOR instance also has high value. Moreover, the 4-XOR instance has $\sim k^2 n$ constraints, as a typical $v \in [n]$ participates in $\sim k$ hyperedges in $\cup_{i=1}^k H_i$, and hence can be "canceled" to form k^2 derived constraints.

We can then apply the Kikuchi matrix method and CSP refutation machinery to try to refute this 4-XOR instance. However, because each H_i' is no longer a matching, the resulting Kikuchi matrices (which, recall, are "zeroed out" versions of the original matrices in Definition 2.1.1) may have very few nonzero entries. Namely, the analogue of Lemma 2.3.3 does not necessarily hold. However, it does hold if we assume that any pair $p = (u, v)$ of vertices appears in at most $\mathrm{polylog}(n)$ hyperedges in the original 3-uniform hypergraph $\cup_{i=1}^k H_i$. If we make this assumption, then we can prove that $n \geq k^3/\mathrm{polylog}(k)$. We note that a recent work [BCG20] managed to reprove that $n \geq k^2/\mathrm{polylog}(k)$ under a similar assumption about pairs of vertices.

Thus, the final step of the proof is to remove the assumption by showing that no pair of vertices can appear in too many hyperedges. Suppose that we do have many "heavy" pairs $p = (u, v)$ that appear in $\gg \log n$ clauses in the original 3-uniform hypergraph $H := \cup_{i=1}^k H_i$. Now, we transform the 3-XOR instance into a bipartite 2-XOR instance ([AGK21, GKM22]) by replacing each heavy pair p with a new variable y_p. That is, the 3-XOR clause $C = (u, v, w)$ in H_i now becomes the 2-XOR clause (p, w), where p is a new variable. In other words, the constraint $x_u x_v x_w = b_i$ is replaced by $y_p x_w = b_i$. Each clause in the bipartite 2-XOR instance now uses one variable from the set of heavy pairs, and one from the original set of variables $[n]$. We then show that if there are too many heavy pairs, then this instance has a sufficient number of constraints in order to be refuted, and is thus not satisfiable, which is again a contradiction.

Finally, we note that for larger odd $q \geq 5$, the proof showing that there not too many heavy pairs breaks down, and this is what prevents us from generalizing Theorem 7 to all odd q.

Formally, the argument to bound $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(\Psi_b)]$ proceeds in two steps:

(1) **Decomposition:** First, we take any pair $Q = \{u, v\}$ of vertices that appears in $\gg \log n$ of the hyperedges in $H := \cup_{i=1}^k H_i$, and we replace this pair with a new variable y_Q in all the

[1] If $|C_i \cap C_j| = 2$, then the derived constraint is a 2-XOR constraint, not 4-XOR. This is a minor technical issue that can be circumvented easily, so we will ignore it for the proof overview.

constraints containing this pair. This process decomposes the 3-XOR instance into a *bipartite* 2-XOR instance ([AGK21, GKM22]), and a residual 3-XOR instance where every pair of variables appears in at most $O(\log n)$ constraints.

(2) **Refutation:** We then produce a "strong refutation" for each of the bipartite 2-XOR and the residual 3-XOR instances that shows that the average value of the instance over the draw of $b \leftarrow \{-1,1\}^k$ is small. This implies that each of the two instances produced and thus the original 3-XOR instance has a small expected value and finishes the proof.

The decomposition and refuting the XOR instances. We now formally define the decomposition process. We recall a notion of degree in hypergraphs that turns out to be useful in our argument (similar to the analysis in Section 5.2).

Definition 11.0.1 (Degree)**.** Let H be a q-uniform hypergraph on n vertices, and let $Q \subseteq [n]$. The degree of Q, $\deg_H(Q)$, is the number of $C \in H$ with $Q \subseteq C$.

Lemma 11.0.2 (Hypergraph Decomposition)**.** *Let $H_1, \ldots, H_k$ be 3-uniform hypergraphs on n vertices, and let $H := \cup_{i=1}^k H_i$. Let $d \in \mathbb{N}$ be a threshold. Let $P := \{\{u,v\} : \deg_H(\{u,v\}) > d\}$. Then, there are 3-uniform hypergraphs $H'_1, \ldots, H'_k$ and bipartite graphs $G_1, \ldots, G_k$, with the following properties.*

(1) Each G_i is a bipartite graph with left vertices $[n]$ and right vertices P.

(2) Each H'_i is a subset of H_i.

(3) For each $i \in [k]$, there is a one-to-one correspondence between hyperedges $C \in H_i \setminus H'_i$ and edges e in G_i, given by $e = (w, \{u,v\}) \mapsto C = \{u,v,w\}$.

(4) Let $H' := \cup_{i=1}^k H'_i$. Then, for any $u \neq v \in [n]$, it holds that $\deg_{H'}(\{u,v\}) \leq d$.

(5) If H_i is a matching, then H'_i and G_i are also matchings.

The proof of Lemma 11.0.2 is simple, and is given in Section 11.0.1.

Given the decomposition, the two main steps in our refutation are captured in the following two lemmas, which handle the 2-XOR and 3-XOR instances, respectively.

Lemma 11.0.3 (2-XOR refutation)**.** *Fix $n \in \mathbb{N}$ and $k \leq n$. Let $G_1, \ldots, G_k$ be bipartite matchings with left vertices $[n]$ and a right vertex set P of size $|P| \leq nk/d$ for some $d \in \mathbb{N}$. For $b \in \{-1,1\}^k$, let $g_b(x,y)$ be a homogeneous quadratic polynomial defined by*

$$g_b(x,y) := \sum_{i=1}^{k} b_i \sum_{e=\{v,p\} \in G_i : v \in [n], p \in P} x_v y_p \,,$$

and let $\mathrm{val}(g_b) := \max_{x \in \{-1,1\}^n, y \in \{-1,1\}^P} g_b(x,y)$. Then, $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(g_b)] \leq O(nk\sqrt{(\log n)/d})$.

Lemma 11.0.4 (3-XOR refutation)**.** *Let $H_1, \ldots, H_k$ be 3-uniform hypergraph matchings on n vertices, and let $H := \cup_{i=1}^k H_i$. Suppose that for any $\{u,v\} \subseteq [n]$, $\deg_H(\{u,v\}) \leq d$. Let $f_b(x) := \sum_{i=1}^k b_i \sum_{C \in H_i} \prod_{v \in C} x_v$. Then, it holds that*

$$\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(f_b)] \leq n\sqrt{kd} \cdot O\left((nk)^{1/8} \log^{1/4} n\right) \,.$$

We prove Lemma 11.0.3 in Section 11.0.2, and we prove Lemma 11.0.4 in Section 11.1.

With the above ingredients, we can now finish the proof of Theorem 7.

Proof of Theorem 7. Applying Lemma 11.0.2 with $d = O((\log n)/\varepsilon^2\delta^2)$ for a sufficiently large

constant, we decompose the instance Ψ_b into 2-XOR and 3-XOR subinstances.[2] Note that as $m \le nk$, we will have $|P| \le m/d \le nk/d$. We have that $m \operatorname{val}(\Psi_b) \le \operatorname{val}(f_b) + \operatorname{val}(g_b)$ because of the one-to-one correspondence property in Lemma 11.0.2. We also note that $m \ge \delta nk$, as $|H_i| \ge \delta n$ for each i. By Lemma 11.0.3 and by taking the constant in the choice of d sufficiently large, we can ensure that $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\operatorname{val}(g_b)] \le \varepsilon\delta nk/3$. Hence, by Eq. (11.1) and Lemma 11.0.4, we have

$$\begin{aligned}
&2\varepsilon\delta nk \le 2\varepsilon m \le m\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\operatorname{val}(\Psi_b)] \le \mathbb{E}_{b \leftarrow \{-1,1\}^k}[\operatorname{val}(f_b) + \operatorname{val}(g_b)] \\
&\le \frac{\varepsilon\delta nk}{3} + n\sqrt{k} \cdot O(\sqrt{\log n})/(\varepsilon\delta) \cdot (nk)^{1/8} \log^{1/4} n \\
&\implies \varepsilon^2\delta^2\sqrt{k} \le O(\sqrt{\log n}) \cdot (nk)^{1/8} \log^{1/4} n \\
&\implies k^3 \le n \cdot O(\log^6 n)/(\varepsilon^{16}\delta^{16}) \ .
\end{aligned}$$

We thus conclude that $k^3 \le n \cdot O\left(\frac{\log^6 n}{\varepsilon^{16}\delta^{16}}\right)$, which finishes the proof. □

11.0.1 Hypergraph decomposition: proof of Lemma 11.0.2

We prove Lemma 11.0.2 by analyzing the following greedy algorithm.

Algorithm 11.0.5.
Given: 3-uniform hypergraphs $H_1, \ldots, H_k$.
Output: 3-uniform hypergraphs $H'_1, \ldots, H'_k$ and bipartite graphs $G_1, \ldots, G_k$.
Operation:

1. **Initialize:** $H'_i = H_i$ for all $i \in [k]$, $P = \{\{u,v\} : \deg_{H'}(\{u,v\}) > d\}$, where $H' = \cup_{i\in[k]} H'_i$.
2. **While** P **is nonempty**:
   (1) Choose $p = \{u,v\} \in P$ arbitrarily.
   (2) For each $i \in [k]$, $C \in H'_i$ with $p \in C$, remove C from H'_i, and add the edge $(C \setminus p, p)$ to G_i.
   (3) Recompute $P = \{\{u,v\} : \deg_{H'}(\{u,v\}) > d\}$.
3. Output $H'_1, \ldots, H'_k, G_1, \ldots, G_k$.

Indeed, properties (1), (2) and (5) in Lemma 11.0.2 trivially hold. Property (4) holds because otherwise the algorithm would not have terminated, as the set P would still be nonempty. Property (3) holds because each hyperedge $C \in H_i$ starts in H'_i, and is either removed exactly once and added to G_i as $(C \setminus p, p)$, or remains in H'_i for the entire operation of the algorithm. This finishes the proof.

11.0.2 Refuting the 2-XOR instance: proof of Lemma 11.0.3

We now prove Lemma 11.0.3. We do this as follows. For each $e = \{v,p\}$, with $v \in [n]$, $p \in P$, define the matrix $A^{(e)} \in \mathbb{R}^{n\times P}$, where $A^{(e)}(v',p') = 1$ if $v' = v$ and $p' = p$, and 0 otherwise. Let

[2] We remark that it is possible that one (but not both!) of the 2-XOR or 3-XOR subinstances has very few constraints, or even no constraints at all. This is not a problem, however, as then the upper bound on the value of the instance shown in corresponding lemma (either Lemma 11.0.3 or Lemma 11.0.4) becomes trivial.

$A_i \coloneqq \sum_{e \in G_i} A^{(e)}$, the bipartite adjacency matrix of G_i. Finally, let $A \coloneqq \sum_{i=1}^{k} b_i A_i$.

First, we observe that $\mathrm{val}(g_b) \leq \sqrt{n|P|}\|A\|_2$. Indeed, this is because for any $x \in \{-1,1\}^n, y \in \{-1,1\}^P$, we have $g_b(x,y) = x^\top A y \leq \|x\|_2 \|y\|_2 \|A\|_2 = \sqrt{n|P|}\|A\|_2$. Thus, in order to bound $\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(g_b)]$, it suffices to bound $\mathbb{E}_b[\|A\|_2]$.

We use Fact 3.4.2 to bound $\mathbb{E}[\|A\|_2]$. Indeed, we observe that $\|A_i\|_2 \leq 1$ for each i, as each row/column of A_i has at most one nonzero entry of magnitude 1 because each G_i is a matching. Thus, $\max(\|\sum_{i=1}^{k} A_i A_i^\top\|, \|\sum_{i=1}^{k} A_i^\top A_i\|) \leq k$. As the b_i's are i.i.d. from $\{-1,1\}$, by Fact 3.4.2 we have that $\mathbb{E}[\|A\|_2] \leq O(\sqrt{k \log n})$. It thus follows that $\mathbb{E}[\mathrm{val}(g_b)] \leq \sqrt{n|P|} O(\sqrt{k \log n}) \leq O(nk\sqrt{(\log n)/d})$.

11.1 Refuting the 3-XOR instance: proof of Lemma 11.0.4

In this section, we will omit the subscript and write f instead of f_b. We will also let $m \coloneqq |H| = \sum_{i=1}^{k} |H_i|$.

For a vertex $u \in [n]$ and a subset $C \in \binom{[n]}{2}$, we will use the notation (u, C) to denote the set $\{u\} \cup C$. We will assume that $k \leq n/c$ for some sufficiently large absolute constant c. This is without loss of generality, as otherwise we can partition k into at most c disjoint blocks of size $\leq n/c$, and refute each of these subinstances separately.

The main idea is inspired by the "Cauchy-Schwarz" trick in the context of refuting odd-arity XOR instances. Specifically, we will construct a 4-XOR instance by "canceling" out every x_u that appears in two different clauses. Concretely, include every element in $[k]$ into one of two sets L, R uniformly at random. Then, for any $(u, C) \in H_i$ with $i \in L$ and $(u, C') \in H_j$ with $j \in R$, we construct the "derived clause" $C \oplus C'$ by XOR-ing both sides of the two constraints. We then relate the value of the instance with such derived constraints to the original 3-XOR instance and produce a spectral refutation for the derived instance via an appropriate subexponential-sized matrix. This will show that the expected value of the derived instance, over the randomness of the b_i's, is small, and complete the proof.

Relating the derived 4-XOR to the original 3-XOR. First, let (L, R) be a partition of $[k]$ into two sets of equal size $k/2$. Let $f_{L,R}(x)$ be the following polynomial:

$$f_{L,R}(x) \coloneqq \sum_{\substack{i \in L \\ j \in R}} \sum_{u \in [n]} \sum_{\substack{(u,C) \in H_i \\ (u,C') \in H_j}} b_i b_j x_C x_{C'} \ ,$$

where x_C is defined as $\prod_{v \in C} x_v$. We note that because the H_i's are matchings, after fixing i, j, and u, there is at most one pair (C, C') in the inner sum. Informally speaking, only working with clauses derived across the partition allows us to "preserve" $\sim k$ independent bits of randomness in the right-hand sides of the 4-XOR instance while eliminating nontrivial correlations. This is crucial in eventually applying the Matrix Khintchine inequality to produce a spectral refutation.

The following lemma relates $\mathrm{val}(f_{L,R})$ to $\mathrm{val}(f)$.

Lemma 11.1.1 (Cauchy-Schwarz Trick). *Let f be as in Lemma 11.0.4 and let $L, R \subseteq [k]$ be constructed by including every element in $[k]$ to be in L with probability $1/2$ independently and defining $R = [k] \setminus L$. Then, it holds that $9 \cdot \mathrm{val}(f)^2 \leq 3nm + 4n\mathbb{E}_{(L,R)} \mathrm{val}(f_{L,R})$. In particular, $\mathbb{E}_{b \in \{-1,1\}^k}[9 \cdot \mathrm{val}(f)^2] \leq 3nm + 4n\mathbb{E}_{(L,R)}\mathbb{E}_{b \in \{-1,1\}^k}[\mathrm{val}(f_{L,R})]$.*

Proof. Fix any assignment to $x \in \{-1,1\}^n$. We have that

$$(3f(x))^2 = \left(\sum_{u\in[n]} x_u \sum_{i\in[k]} \sum_{(u,C)\in H_i} b_i x_C\right)^2 \le \left(\sum_{u\in[n]} x_u^2\right)\left(\sum_{u\in[n]}\left(\sum_{i\in[k]}\sum_{(u,C)\in H_i} b_i x_C\right)^2\right)$$

$$= n\sum_{u\in[n]}\sum_{i,j\in[k]}\sum_{\substack{(u,C)\in H_i\\(u,C')\in H_j}} b_i b_j x_C x_{C'} = n\left(3\sum_{i\in[k]}|H_i| + \sum_{u\in[n]}\sum_{i,j\in[k],i\neq j}\sum_{\substack{(u,C)\in H_i\\(u,C')\in H_j}} b_i b_j x_C x_{C'}\right)$$

$$= 3nm + 4n\cdot \mathbb{E}_{(L,R)} f_{L,R}(x)\ ,$$

where the first equality is because there are 3 ways to decompose a set $C_i \in H_i$ with $|C_i| = 3$ into a pair (u, C), the inequality follows by the Cauchy-Schwarz inequality, and the last equality follows because for a pair of hypergraphs H_i and H_j, we have $i \in L$ and $j \in R$ with probability $1/4$. Finally, $\max_{x\in\{-1,1\}^n} \mathbb{E}_{(L,R)} f_{L,R}(x) \le \mathbb{E}_{(L,R)} \max_{x\in\{-1,1\}^n} f_{L,R}(x) = \mathbb{E}_{(L,R)} \mathrm{val}(f_{L,R})$. Thus, we have that $9\cdot \mathrm{val}(f)^2 \le 3nm + 4n\cdot \mathbb{E}_{(L,R)} \mathrm{val}(f_{L,R})$. □

11.1.1 Bounding $\mathrm{val}(f_{L,R})$ using CSP refutation

It remains to bound $\mathbb{E}_{b\in\{-1,1\}^k} \mathrm{val}(f_{L,R})$ for each choice of partition (L, R). We will do this by introducing a matrix B for each $b \in \{-1,1\}^k$ and partition (L, R), and then we will relate $\mathrm{val}_{f_{L,R}}$ to $\|B\|_2$. Note that B will depend on the choice of b and the partition (L, R). Then, we will bound $\mathbb{E}_{b\in\{-1,1\}^k}[\|B\|_2]$.

To define the matrix B, we introduce the following definitions.

Definition 11.1.2. Let $u \in [n]$ be a vertex. We let $u^{(1)}$ and $u^{(2)}$ denote the elements $(u, 1)$ and $(u, 2)$ of $[n]\times[2]$, i.e., if we think of $[n]\times[2]$ as two copies of $[n]$, then $u^{(1)}$ is the first copy and $u^{(2)}$ is the second one. We use similar notation for sets, so if $C \subseteq [n]$, then $C^{(1)}$ and $C^{(2)}$ denote the subsets of $[n]\times[2]$ defined as $C^{(b)} = \{(i, b) : i \in C\}$ for $b \in [2]$.

Definition 11.1.3 (Half clauses)**.** For $i \in L$, $j \in R$, we define the set $P_{i,j}$ of "half clauses" to consist of all pairs $(v^{(1)}, w^{(2)})$ such that there exist clauses $(u, C) \in H_i$, $(u, C') \in H_j$ where $v \in C$ and $w \in C'$.

We let $P_i := \cup_{j\in R} P_{i,j}$.

Our matrix is easiest to define in two steps. We first define a matrix A. Then, we will specify some modifications to A that yield the final matrix B.

Definition 11.1.4 (Our initial Kikuchi matrix)**.** Let $\ell := (\sqrt{n/k})/c$ for some sufficiently large constant c,[3] and let $N := \binom{2n}{\ell}$. For any two sets $S, T \subseteq [n]\times[2]$ and sets $C, C' \in \binom{[n]}{2}$, we say that $S \overset{C,C'}{\leftrightarrow} T$ if

1. $S \oplus T = C^{(1)} \oplus C'^{(2)}$,
2. $|S \cap C^{(1)}| = |S \cap C'^{(2)}| = |T \cap C^{(1)}| = |T \cap C'^{(2)}| = 1$.

Note that $C^{(1)} \oplus C'^{(2)} = C^{(1)} \cup C'^{(2)}$, as $C^{(1)}$ and $C'^{(2)}$ are disjoint by construction.

[3] We note that the matrix is only well-defined if $\ell \ge 2$, but this holds because we assumed that $k \le n/c'$ for some sufficiently large absolute constant c'. This is the only place where we will use this assumption.

For each $i \in L$ and $C, C' \in \binom{[n]}{2}$, define the $N \times N$ matrix $A^{(i,C,C')}$, indexed by sets $S \subseteq [n] \times [2]$ of size ℓ, by setting $A^{(i,C,C')}(S,T) = 1$ if (1) $S \overset{C,C'}{\leftrightarrow} T$, and (2) each of S and T contains at most one half clause from P_i. Otherwise, we set $A^{(i,C,C')}(S,T) = 0$.

Finally, we let

$$A_{i,j} \coloneqq \sum_{u \in [n]} \sum_{(u,C) \in H_i, (u,C') \in H_j} A^{(i,C,C')}, \quad A_i \coloneqq \sum_{j \in R} b_j A_{i,j}, \quad \text{and} \quad A \coloneqq \sum_{i \in L} b_i A_i \ .$$

Remark 11.1.5. For a fixed choice of $(u,C) \in H_i$, $(u,C') \in H_j$ with $j \in R$, the matrix $A^{(i,C,C')}$ has exactly $4\binom{2n-4}{\ell-2}$ nonzero entries, *if we ignore* the additional condition that S and T each contain at most one half clause from P_i. Indeed, this is because $S \overset{C,C'}{\leftrightarrow} T$ if and only if S and T each contain one entry of C and C' (2 choices per clause), and the remaining part of S and T is the same set $Q \subseteq [n] \times [2] \setminus (C^{(1)} \oplus C'^{(2)})$ of size $\ell - 2$ (which has $\binom{2n-4}{\ell-2}$ choices).

We note that this fact is the reason for using subsets of $[n] \times [2]$ rather than just $[n]$. If we used subsets of $[n]$ only, the number of nonzero entries in $A^{(i,C,C')}$ would depend on $|C \oplus C'|$, whereas with subsets of $[n] \times [2]$ we always have $|C^{(1)} \oplus C'^{(2)}| = 4$.

Observe that if $S \overset{C,C'}{\leftrightarrow} T$, then S and T each contain at least one half clause from P_i, namely coming from (C, C'). Thus, the additional condition on S and T is that they contain *no other* half clauses. As we shall show below, this additional condition implies that A_i has at most $2d$ nonzero entries per row and thus $\|A_i\|_2 \le 2d$, where d is the parameter in the statement of Lemma 11.0.4, *without* meaningfully affecting the number of nonzero entries in each of the $A^{(i,C,C')}$'s. We note that without this condition, one can show that $\|A_i\|_2 \ge \Omega(\ell)$, which is large.

Lemma 11.1.6 (Nonzero entry bound)**.** *For $i \in L$, let A_i be defined as in Definition 11.1.4. Then, A_i has at most $2d$ nonzero entries per row/column.*

We postpone the proof of Lemma 11.1.6 to Section 11.1.3, and now continue with the proof.

The following lemma shows that the number of nonzero entries in $A^{(i,C,C')}$ is at least $2\binom{2n-4}{\ell-2}$, i.e., half of $4\binom{2n-4}{\ell-2}$; thus, the additional condition only decreases the number of nonzero entries by a factor of 2 per derived constraint. The factor of 2 is not important and is chosen for convenience, and determines the constant c in the parameter ℓ.

Lemma 11.1.7 (Counting nonzero entries)**.** *For some $(u,C) \in H_i$ and $(u,C') \in H_j$ with $j \in R$, let $A^{(i,C,C')}$ be as in Definition 11.1.4. Then, the number of nonzero entries in $A^{(i,C,C')}$ is at least $2\binom{2n-4}{\ell-2}$.*

We postpone the proof of Lemma 11.1.7 to Section 11.1.2, and now continue with the proof.

We obtain the final matrix B by, for each $B^{(i,C,C')}$, zero-ing out entries of $A^{(i,C,C')}$ until it has *exactly* $2\binom{2n-4}{\ell-2}$ nonzero entries. This is identical to the "equalizing step" of the edge deletion process in [HKM23].

Definition 11.1.8 (Our final Kikuchi matrix)**.** For each $i \in L$ and each pair of clauses $(u,C) \in H_i$ and $(u,C') \in H_j$ with $j \in R$, let $B^{(i,C,C')}$ be the matrix obtained from $A^{(i,C,C')}$ by arbitrarily zero-ing out entries of $A^{(i,C,C')}$ until the resulting matrix has exactly $D \coloneqq 2\binom{2n-4}{\ell-2}$ nonzero entries.

We let

$$B_{i,j} \coloneqq \sum_{u \in [n]} \sum_{(u,C) \in H_i, (u,C') \in H_j} B^{(i,C,C')}, \quad B_i \coloneqq \sum_{j \in R} b_j B_{i,j}, \quad \text{and} \quad A \coloneqq \sum_{i \in L} b_i B_i \ .$$

We are now ready to finish the proof. First, we relate $\|B\|_2$ to $\mathrm{val}(f_{L,R})$. Fix an assignment $x \in \{-1,1\}^n$, and let $z \in \{-1,1\}^N$ be defined as $z_S := \prod_{u\in S_1} x_u \prod_{v\in S_2} x_v$ for $S = S_1^{(1)} \cup S_2^{(2)} \subseteq [n]\times[2]$ satisfying $|S| = \ell$.

We observe that $D \cdot f_{L,R}(x) = z^\top B z$. This is because:

(1) For $S, T \subseteq [n]\times[2]$ with $S \oplus T = C^{(1)} \oplus C'^{(2)}$, we have $z_S z_T = \prod_{u\in S_1} x_u \prod_{v\in S_2} x_v \prod_{u'\in T_1} x_u \prod_{v'\in T_2} x_v = \prod_{u\in S_1\oplus T_1} x_u \prod_{v\in S_2\oplus T_2} x_v = \prod_{u\in C} x_u \prod_{v\in C'} x_v$,

(2) For a pair of clauses $(u, C) \in H_i$ and $(u, C') \in H_j$ with $i \in L$ and $j \in R$, there are exactly $D = 2\binom{2n-4}{\ell-2}$ nonzero entries (S, T) of $B^{(i,C,C')}$, and these entries have $S \oplus T = C^{(1)} \oplus C'^{(2)}$,

which implies that $z^\top B^{(i,C,C')} z = D x_C x_{C'}$. Hence,

$$z^\top B z = \sum_{\substack{i\in L\\ j\in R}} \sum_{u\in[n]} \sum_{\substack{(u,C)\in H_i\\ (u,C')\in H_j}} b_i b_j \cdot z^\top B^{(i,C,C')} z = \sum_{\substack{i\in L\\ j\in R}} \sum_{u\in[n]} \sum_{\substack{(u,C)\in H_i\\ (u,C')\in H_j}} b_i b_j \cdot D x_C x_{C'} = D \cdot f_{L,R}(x) \ .$$

In particular, this implies

$$\mathrm{val}(f_{L,R}) \le \frac{N}{D} \cdot \|B\|_2 \ . \tag{11.2}$$

It thus remains to bound $\mathbb{E}_{b\in\{-1,1\}^k}[\|B\|_2]$, which we do in the following lemma.

Lemma 11.1.9 (Spectral norm bound)**.** $\mathbb{E}_{b\in\{-1,1\}^k}[\|B\|_2] \le d \cdot O(\sqrt{k\ell \log n})$.

We postpone the proof of Lemma 11.1.9 to Section 11.1.3, and now finish the proof of Lemma 11.0.4.

Proof of Lemma 11.0.4. By Eq. (11.2) and Lemma 11.1.9, we have that

$$\begin{aligned}
\mathbb{E}_{b\in\{-1,1\}^k}[\mathrm{val}(f_{L,R})] &\le \frac{N}{D}\mathbb{E}_{b\in\{-1,1\}^k}[\|B\|_2] \\
&\le \frac{N}{D}\left(d \cdot O(\sqrt{k\ell\log n})\right) \le \frac{n^2}{\ell^2} d \cdot O(\sqrt{k\ell\log n}) \\
&= nkd \cdot O((nk)^{1/4}\sqrt{\log n}) \ ,
\end{aligned}$$

where we use that $\ell = (\sqrt{n/k})/c$ for some constant c, and we use Fact 3.6.1 to bound N/D. Finally, combining with Lemma 11.1.1 and using that $m \le nk$, we have that

$$\begin{aligned}
\mathbb{E}[\mathrm{val}(f)]^2 \le \mathbb{E}[\mathrm{val}(f)^2] &\le \frac{1}{9} \cdot \left(3n^2k + 4n\mathbb{E}_{(L,R)}\mathbb{E}_{b\in\{-1,1\}^k}[\mathrm{val}(f_{L,R})]\right) \\
&\le n^2kd \cdot O((nk)^{1/4}\sqrt{\log n}) \ .
\end{aligned}$$

Hence,

$$\mathbb{E}[\mathrm{val}(f)] \le n\sqrt{kd} \cdot O\left((nk)^{1/8}\log^{1/4} n\right) \ ,$$

which finishes the proof of Lemma 11.0.4. □

11.1.2 Counting nonzero entries: proof of Lemma 11.1.7

Proof of Lemma 11.1.7. Let $j \in R$ and clauses $(u, C) \in H_i$ and $(u, C') \in H_j$. Recall that in Remark 11.1.5, we observed that there are exactly $4\binom{2n-4}{\ell-2}$ pairs (S, T) with $S \overset{C,C'}{\leftrightarrow} T$. Indeed, this is because $S \overset{C,C'}{\leftrightarrow} T$ if and only if S and T each contain one entry of C and C' (2 choices per clause), and the remaining part of S and T is the same set $Q \subseteq [n] \times [2] \setminus (C^{(1)} \oplus C'^{(2)})$ of size $\ell - 2$ (which has $\binom{2n-4}{\ell-2}$ choices).

From the above, we observe that for each $Q \subseteq [n] \times [2] \setminus (C^{(1)} \oplus C'^{(2)})$ of size $\ell - 2$, we can identify Q with 4 different pairs (S, T) with $S \overset{C,C'}{\leftrightarrow} T$; namely, each pair (S, T) corresponds to a subset of size 2 of (C, C') containing exactly one entry from each of C, C'. We note that these 4 choices of (S, T) correspond exactly to the 4 half clauses in P_i contributed by the derived clause (C, C'). We will show that for at least $\frac{1}{2}\binom{2n-4}{\ell-2}$ choices of Q, *all* 4 corresponding choices of (S, T) will contain exactly one derived clause from P_i: namely, the half clause of (C, C') that we add to Q to obtain S or T. This clearly suffices to finish the proof.

Call such a set Q *bad* if it does not have the above property, i.e., there is some pair (S, T) identified with Q such that one of S or T contains more than one half clause from P_i. Since $S \overset{C,C'}{\leftrightarrow} T$ already implies that each of S and T has exactly one half clause from $C^{(1)} \oplus C'^{(2)}$, there are three ways that Q can be bad:

(1) Q contains a half clause from P_i,
(2) there is $v^{(1)} \in C^{(1)}$ and $w^{(2)} \in Q$ such that $(v^{(1)}, w^{(2)}) \in P_i$,
(3) there is $v^{(1)} \in Q$ and $w^{(2)} \in C'^{(2)}$ such that $(v^{(1)}, w^{(2)}) \in P_i$.

We thus have that the number of bad Q's is at most

$$p_0\binom{2n-6}{\ell-4} + p_1\binom{2n-5}{\ell-3} + p_2\binom{2n-5}{\ell-3},$$

where $p_0 = |P_i|$, $p_1 = |\{(v^{(1)}, w^{(2)}) \in P_i : v^{(1)} \in C^{(1)}\}|$, $p_2 = |\{(v^{(1)}, w^{(2)}) \in P_i : w^{(2)} \in C'^{(2)}\}|$.

We now upper bound p_0, p_1, p_2. Recall that a half clause in P_i is a pair $(v^{(1)}, w^{(2)})$ such that there are clauses $(u, C_1) \in H_i$, $(u, C_2) \in H_j$ with $j \in R$, and $v \in C_1$, $w \in C_2$.

(1) We have $p_0 \leq 4nk$, as for each $u \in [n]$, because the H_i's are matchings, there is at most one C_1 such that $(u, C_1) \in H_i$, and at most k choices of $(u, C_2) \in H_j$ with $j \in R$, as $|R| \leq k$. Finally, each choice of (C_1, C_2) yields 4 half clauses.
(2) We have $p_1 \leq 8k$. First, there are at most 2 choices for v, each coming from C. For each such v, there is at most one $C_i \in H_i$ with $v \in C_i$. (Note that $|C_i| = 3$.) Once C_i is fixed, we have at most 2 choices for u, given by $C_i \setminus \{v\}$, and there are at most k hyperedges $(u, C_2) \in H_j$ for $j \in R$ (as each H_j is a matching and $|R| \leq k$). Finally, for each such C_2 there are 2 possible choices for w.
(3) We have $p_2 \leq 8k$. First, there are at most 2 choices for w, each coming from C'. For each such w, there are at most k choices of $C_j \in \cup_{j \in R} H_j$ with $w \in C_j$, as each H_j is a matching and $|R| \leq k$. (Note that $|C_j| = 3$.) For each such C_j, there are at most 2 choices for u, given by $C_j \setminus \{w\}$, and for each u, there is at most one choice of C_1 such that $(u, C_1) \in H_i$. Finally, such a C_1, if it exists, gives 2 choices for v.

Combining, we thus have that the number of bad Q's is at most

$$4nk\binom{2n-6}{\ell-4}+16k\binom{2n-5}{\ell-3}\ .$$

We have that

$$\frac{4nk\binom{2n-6}{\ell-4}+16k\binom{2n-5}{\ell-3}}{\binom{2n-4}{\ell-2}}=\frac{4nk\frac{(2n-6)!}{(\ell-4)!(2n-2-\ell)!}+16k\frac{(2n-5)!}{(\ell-3)!(2n-2-\ell)!}}{\frac{(2n-4)!}{(\ell-2)!(2n-2-\ell)!}}$$
$$=4nk\frac{(\ell-2)(\ell-3)}{(2n-4)(2n-5)}+16k\frac{\ell-2}{2n-4}\leq\frac{1}{2}\ ,$$

as we have $\ell\leq(\sqrt{n/k})/c$, for some sufficiently large constant c, $\ell\geq 2$, and $k\leq\sqrt{nk}$ since $k\leq n$. □

11.1.3 Spectral norm bound: proof of Lemmas 11.1.6 and 11.1.9

Proof of Lemma 11.1.6. Fix $i\in L$. We show that each row/column of A_i has at most $2d$ nonzero entries. Indeed, this is because if S is a nonzero row (or column) in A_i, then S contains at most one half clause from P_i. If (C,C') is a derived clause where $S\overset{C,C'}{\leftrightarrow}T$ for some T, then S must contain a half clause in P_i that is contained in $C^{(1)}\oplus C'^{(2)}$, i.e., a half clause coming from (C,C'). As S contains at most one half clause, it follows that the number of nonzero entries in the S-th row is upper bounded by the maximum, over all half clauses, of the number of derived clauses (C,C') that contain this half clause. One can observe that this is $2d$. Indeed, if we fix $v^{(1)}$ and $w^{(2)}$, there is at most one clause $C\in H_i$ containing v. Once v is fixed, there are two choices for u in $C\setminus\{v\}$. Once we have chosen u, the second clause must be $(u,C')\in H_j$ for some $j\in R$, where C' contains w. By assumption, the number of hyperedges in $\cup_{i=1}^{k}H_i$ containing the pair $\{u,w\}$ is at most d, so there are at most d choices for C'. □

Proof of Lemma 11.1.9. We have that $B=\sum_{i\in L}b_iB_i$, where the b_i's are i.i.d. from $\{-1,1\}$. By Lemma 11.1.6, we know that the number of nonzero entries in a row/column of A_i is at most $2d$. As B_i is obtained by zero-ing out entries of A_i, it follows that this also holds for B_i. It thus follows that the ℓ_1-norm of any row/column of B_i is at most $2d$, and thus $\|B_i\|_2\leq 2d$. This additionally implies that $\|\sum_{i\in L}B_iB_i^\top\|_2\leq|L|(2d)^2\leq k(2d)^2$, and that $\|\sum_{i\in L}B_i^\top B_i\|_2\leq|L|(2d)^2\leq k(2d)^2$. Applying Matrix Khintchine (Fact 3.4.2), we conclude that $\mathbb{E}[\|B\|_2]\leq d\cdot O(\sqrt{k\log N})$. As $\log N=O(\ell\log n)$, Lemma 11.1.9 follows. □

11.2 Improved lower bounds for 3-LDCs over larger alphabets

In this appendix, we will extend Theorem 7 to 3-query LDCs over larger alphabets, which will follow from combining Theorem 7 with standard results from [KT00, KW04]. We first define LDCs over general alphabets.

Definition 11.2.1 (LDCs over general alphabets)**.** Given a positive integer q, constants $\delta,\varepsilon>0$, and an alphabet Σ, we say a code $C\colon\{0,1\}^k\to\Sigma^n$ is (q,δ,ε)-locally decodable code (abbreviated

(q, δ, ε)-LDC) if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ with the following properties. The algorithm $\mathrm{Dec}(\cdot)$ is given oracle access to some $y \in \Sigma^n$, takes an $i \in [k]$ as input, and satisfies the following: (1) the algorithm Dec makes at most q queries to the string y, and (2) for all $b \in \{0,1\}^k$, $i \in [k]$, and all $y \in \Sigma^n$ such that $\Delta(y, C(b)) \leq \delta n$, $\Pr[\mathrm{Dec}^y(i) = b_i] \geq \frac{1}{2} + \varepsilon$.

Our extension of Theorem 7 to larger alphabets is the following theorem.

Theorem 11.2.2. *Let $C\colon \{0,1\}^k \to \Sigma^n$ be a $(3, \delta, \varepsilon)$-LDC. Then, it must hold that $k^3 \leq |\Sigma|^{41} n \cdot O(\log^6(|\Sigma| n)/\varepsilon^{32}\delta^{16})$. In particular, if δ, ε are constants and $|\Sigma| \leq n$, then $n \geq \Omega(k^3/(|\Sigma|^{41} \log^6 k))$.*

Note that the conclusion of Theorem 11.2.2 is trivial if $|\Sigma| = \tilde{\Omega}(k^{3/41})$. To prove Theorem 11.2.2, it suffices to show the following lemma.

Lemma 11.2.3. *Let $C\colon \{0,1\}^k \to \Sigma^n$ be a $(3, \delta, \varepsilon)$-LDC. Then, there exists a binary code $C'\colon \{0,1\}^k \to \{0,1\}^{n'}$ with $n' \leq 4n|\Sigma|$ and 3-uniform matchings $H'_1, \ldots, H'_k$ over n' vertices such that for all $i \in [k]$, we have $|H'_i| \geq \varepsilon\delta n'/(36|\Sigma|)$. Furthermore, for any query set $C \in H'_i$, we have that $\Pr_{b \leftarrow \{0,1\}^k}[b_i = \oplus_{v \in C} C(b)_v] \geq \frac{1}{2} + \frac{\varepsilon}{8|\Sigma|^{3/2}}$.*

Indeed, once we have Lemma 11.2.3, then by applying Theorem 7 on the resulting normal LDC,[4] we obtain Theorem 11.2.2. Now, to prove Lemma 11.2.3, we first need the following result from [KT00].

Lemma 11.2.4 (Theorem 1 + Lemma 4 in [KT00])**.** *Let $C\colon \{0,1\}^k \to \Sigma^n$ be a (q, δ, ε)-LDC. Then, there exists q-uniform matchings $H_1, \ldots, H_k$ over $[n]$ such that for all $i \in [k]$, we have $|H_i| \geq \varepsilon\delta n/q^2$. Furthermore, for any query set $C \in H_i$, there exists a function $f_C\colon \Sigma^q \to \{0,1\}$ such that $\Pr_{b \leftarrow \{0,1\}^k}[b_i = f_C(C(b)|_C)] \geq \frac{1}{2} + \frac{\varepsilon}{2}$.*

Note that formally the statement in [KT00] only guarantees that each query set in H_i has size at most q rather than *exactly* q. However, we can trivially make each set be of size exactly q by padding each codeword of C with n zeros.

Next, we need the following lemma, which is a generalized and improved version of a similar lemma appearing in [KW04].

Lemma 11.2.5 (Lemma 2 of [KW04])**.** *Let $q \geq 2$ be an integer and let $C\colon \{0,1\}^k \to \Sigma^n$ be a code. Let $H_1, \ldots, H_k$ be q-uniform matchings over $[n]$ such that for each $i \in [k]$, we have $|H_i| \geq \varepsilon\delta n/q^2$, and suppose that for each $C \in H_i$, there exists a function $f_C\colon \Sigma^q \to \{0,1\}$ such that $\Pr_{b \leftarrow \{0,1\}^k}[b_i = f_C(C(b)|_C)] \geq \frac{1}{2} + \frac{\varepsilon}{2}$.*

Then, there exists a binary code $C'\colon \{0,1\}^k \to \{0,1\}^{n'}$ with $n' \leq 4n|\Sigma|$ and q-uniform matchings $H'_1, \ldots, H'_k$ over n' vertices such that for all $i \in [k]$, we have $|H'_i| \geq \varepsilon\delta n'/(4q^2|\Sigma|)$. Furthermore, for any query set $C \in H'_i$, we have that $\Pr_{b \leftarrow \{0,1\}^k}[b_i = \oplus_{v \in C} C'(b)_v] \geq \frac{1}{2} + \frac{\varepsilon}{2^q|\Sigma|^{q/2}}$.

Combining Lemma 11.2.4 and Lemma 11.2.5, we immediately obtain Lemma 11.2.3; Theorem 11.2.2 then follows by applying Theorem 7. Thus, it remains to prove Lemma 11.2.5. In what follows, we use conventional notations of Boolean analysis from [O'D14].

Proof of Lemma 11.2.5. Consider a natural number $\ell \in \mathbb{N}$ such that $|\Sigma| < 2^\ell \leq 2|\Sigma|$, and let $n' := n2^{\ell+1}$. Without loss of generality, say that $\Sigma \subseteq \{0,1\}^\ell$. Consider the first-order Reed-Muller encoding $\mathrm{RM}_1\colon \{0,1\}^\ell \to \{0,1\}^{2^{\ell+1}}$ defined as $\mathrm{RM}_1(\sigma) = (\langle a \rangle \sigma + t)_{a \in \{0,1\}^\ell, t \in \{0,1\}}$.[5] We define our new code $C'\colon \{0,1\}^k \to \{0,1\}^{n'}$ as $C'(b) := (\mathrm{RM}_1(C(b)_1), \ldots, \mathrm{RM}_1(C(b)_n))$.

[4] Note that we obtain a better dependence on ε in Theorem 7 when our initial LDC is in normal form, as shown at the beginning of Chapter 11.

[5] Here, $\langle \cdot \rangle \cdot$ denotes the pointwise inner product over $\mathbb{F}_2^\ell$.

Consider any message index $i \in [k]$ and query set $C \in H_i$. We are going to find a corresponding query set for C in C'. Write $C = \{v_1, \ldots, v_q\}$. Arbitrarily extend our function f_C to a function over $(\{0,1\}^\ell)^q$ by setting $f_C(\sigma) = 0$ for $\sigma \in \{0,1\}^\ell \setminus \Sigma$. For any message $b \in \{0,1\}^k$, set $x := C(b)$. Switching from $\{0,1\}$ to $\{-1,1\}$ in the natural way, we find that

$$\Pr_{b \leftarrow \{0,1\}^k}[b_i = f_C(C(b)|_C)] \geq \frac{1}{2} + \frac{\varepsilon}{2} \iff \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k}[b_i f_C(x_{v_1}, \ldots, x_{v_q})] \geq \varepsilon \, .$$

Consider the Fourier expansion of f_C, written as $f_C(y_1, \ldots, y_q) = \sum_{S_1,\ldots,S_q \subseteq [\ell]} \widehat{f_C}(S_1, \ldots, S_q) \prod_{t=1}^q \prod_{j \in S_t} (y_t)_j$. Using the Fourier expansion of f_C, the Cauchy-Schwarz inequality, and Parseval's identity, we have

$$\begin{aligned}
\varepsilon^2 &\leq \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k}[b_i f_C(x_{v_1}, \ldots, x_{v_q})]^2 \\
&= \left(\sum_{S_1,\ldots,S_q \subseteq [\ell]} \widehat{f_C}(S_1, \ldots, S_q) \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right] \right)^2 \\
&\leq \left(\sum_{S_1,\ldots,S_q \subseteq [\ell]} \widehat{f_C}(S_1, \ldots, S_q)^2 \right) \left(\sum_{S_1,\ldots,S_q \subseteq [\ell]} \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right]^2 \right) \\
&= \left(\mathop{\mathbb{E}}_{y_1,\ldots y_q \leftarrow \{-1,1\}^\ell} [f_C(y_1, \ldots, y_q)^2] \right) \left(\sum_{S_1,\ldots,S_q \subseteq [\ell]} \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right]^2 \right) \\
&= \sum_{S_1,\ldots,S_q \subseteq [\ell]} \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right]^2 \\
&\leq 2^{q\ell} \max_{S_1,\ldots,S_q \subseteq [\ell]} \left\{ \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right]^2 \right\}
\end{aligned}$$

Thus we can find sets $R_1^C, \ldots, R_q^C \subseteq [\ell]$ and bit $t_C \in \{0,1\}$ such that

$$(-1)^{t_C} \mathop{\mathbb{E}}_{b \leftarrow \{-1,1\}^k} \left[b_i \prod_{t=1}^q \prod_{j \in S_t} (x_{v_t})_j \right] \geq \frac{\varepsilon}{2^{q\ell/2}} \geq \frac{\varepsilon}{2^{q-1}|\Sigma|^{q/2}} \, .$$

Reverting back from $\{-1,1\}$ to $\{0,1\}$ in the natural way, the last expression is equivalent to

$$\Pr_{b \leftarrow \{0,1\}^k} \left[t_C + \sum_{i=1}^q \langle \mathbf{1}_{R_1^C} \rangle x_{v_i} = b_i \right] \geq \frac{1}{2} + \frac{\varepsilon}{2^q |\Sigma|^{q/2}} \, .$$

Thus, we can form a new query set $C' := \{(v_1, (\mathbf{1}_{R_1^C}, t_C)), (v_2, (\mathbf{1}_{R_2^C}, 0)), \ldots, (v_q, (\mathbf{1}_{R_q^C}, 0))\}$ for C' that recovers b_i with probability $1/2 + \varepsilon/(2^q|\Sigma|^{q/2})$. Indeed, this is how we construct our new hypergraphs $H_1', \ldots, H_k'$. Since we are mapping each query set to a new one, then we see that $|H_i| = |H_i'| \geq \varepsilon\delta n/q^2 \geq \varepsilon\delta n'/(4q^2|\Sigma|)$ for all $i \in [k]$. Furthermore, the query mapping preserves disjointness and size, implying that the new hypergraph is a collection of k q-uniform matchings. This finishes the proof. □

11.3 Our proof as a black-box reduction to 2-LDC lower bounds

In this appendix, we reinterpret our proof of Theorem 7 in the specific case of *linear* 3-LDCs by formulating it as a black-box reduction to existing linear 2-LDC lower bounds. Because we are reinterpreting the proof, we will assume familiarity with the proof in Chapter 11 and Section 11.1. Formally, we show that our proof of Theorem 7 in fact provides the following transformation: given a *linear* 3-LDC $\mathcal{L}$, we produce 2 different linear codes $\mathcal{L}_2$ and $\mathcal{L}_3$ corresponding to the 2-XOR instance g_b and 3-XOR instance f_b from Chapter 11, with the guarantee that at least one of these codes is a linear 2-LDC. We note that unlike Theorem 7, this reduction-based proof will only apply to *linear* 3-LDCs. However, in this case we will obtain slightly better dependencies on $\log n$, ε, and δ than that in Theorem 7; this comes entirely from the fact that 2-LDC lower bounds for linear codes have slightly better dependencies on ε and δ than 2-LDC lower bounds for general, nonlinear codes.

Our transformation naturally produces objects that are formally not quite linear 2-LDCs, which we call "weak LDCs", defined below.

Definition 11.3.1 (Linear weak LDC)**.** Given a code $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$, we say that $\mathcal{L}$ is a linear (q,δ)-*weakly locally decodable code* (or, (q,δ)-wLDC) if $\mathcal{L}$ is a linear code and there are q-uniform hypergraph matchings $H_1,\dots,H_k$ over $[n]$ such that (1) $\sum_{i=1}^k |H_i| \geq \delta nk$ for any $i \in [k]$, and (2) $C \in H_i$, we have that $\bigoplus_{v\in C} \mathcal{L}(b)_v = b_i$ for all messages $b \in \{0,1\}^k$.

We note that we work with weak LDCs solely for notational convenience, as it is straightforward to observe that they are equivalent to LDCs, up to constant factors in parameters. Indeed, the difference between a weak LDC and a true LDC is that the weak LDC only requires that $\sum_{i=1}^k |H_i| \geq \delta nk$, rather than the stronger condition that $|H_i| \geq \delta n$ for all $i \in [k]$. So, by removing all hypergraphs H_i with $|H_i| \leq \delta n/2$ and setting the corresponding b_i's to 0, we obtain a new code $\mathcal{L}'\colon \{0,1\}^{k'} \to \{0,1\}^n$ where $k' \geq \delta k$ and $|H_i| \geq \delta n/2$ for all $i \in [k']$.

Regardless, we note that the linear 2-LDC lower bound of [GKST06] (Fact 3.3.4), which here we will use as a black-box, holds for linear weak 2-LDCs as well.

As the main theorem in this section, we will prove the following theorem.

Theorem 11.3.2. *Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(3,\delta)$-wLDC, and let $d \in \mathbb{N}$. Then, there are codes $\mathcal{L}_2\colon \{0,1\}^{k_2} \to \{0,1\}^n$ and $\mathcal{L}_3\colon \{0,1\}^{k_3} \to \{0,1\}^N$ such that either $\mathcal{L}_2$ is a linear $(2,\Omega(\delta \cdot \frac{d}{d+k}))$-wLDC or $\mathcal{L}_3$ is a linear $(2,\Omega(\delta^2/d))$-wLDC, where $k_2,k_3 \geq k/2$, $N = \binom{2n}{\ell}$ and $\ell = \sqrt{n/k}/c$, where c is an absolute constant.*

We note that by applying Fact 3.3.4 twice, we immediately obtain the following corollary.

Corollary 11.3.3. *Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a $(3,\delta)$-linear LDC. Then, $n \geq \Omega\left(\frac{\delta^6 k^3}{\log^4 k}\right)$.*

Proof. Apply Theorem 11.3.2 with $d = c\log_2 n/\delta$ for a sufficiently large constant c. If $k \leq d$, then we are done, so suppose that $k \geq d$. If $\mathcal{L}_2$ is a linear weak $(2,\Omega(\delta \cdot \frac{d}{d+k}))$-LDC, then by Fact 3.3.4 we conclude that $\log_2 n \geq \Omega(\delta dk/(k+d)) \geq \Omega(\delta d)$, as $k+d \leq 2k$. As $d = c\log_2 n/\delta$ for a sufficiently large constant c, this is a contradiction.

It thus cannot be the case that $\mathcal{L}_2$ is a linear weak $(2,\Omega(\delta \cdot \frac{d}{d+k}))$-LDC, and therefore it must be the case that $\mathcal{L}_3$ is a linear weak $(2,\Omega(\delta^2/d))$-LDC. By Fact 3.3.4, this implies that $O(\sqrt{n/k}\log n) \geq \ell \log_2 n \geq \Omega(\delta^2/d \cdot k)$, and therefore we conclude that $n \geq \Omega(\delta^6 k^3/\log^4 n)$. Finally, we have $\log_2 n = \Theta(\log k)$ or else Corollary 11.3.3 trivially holds, and so this finishes the proof. □

We now prove Theorem 11.3.2.

Proof of Theorem 11.3.2. Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(3,\delta)$-wLDC, so that there exist 3-uniform hypergraph matchings $H_1,\dots,H_k$ such that $\sum_{i=1}^k |H_i| \geq \delta nk$, and for every $i \in [k]$ and $C \in H_i$, it holds that $\bigoplus_{v\in C} \mathcal{L}(b)_v = b_i$ for all $b \in \{0,1\}^k$.

We now define the codes $\mathcal{L}_2$ and $\mathcal{L}_3$. Let $G_1,\dots,G_k,H'_1,\dots,H'_k$ denote the output of the hypergraph decomposition algorithm Lemma 11.0.2 applied with the parameter d chosen in the statement of Theorem 11.3.2.

Constructing $\mathcal{L}_2$. Let $L_2 \subseteq [k]$ be a subset of size $|L_2| \geq k/2$ to be specified later. We let $\mathcal{L}_2\colon \{0,1\}^{L_2} \to \{0,1\}^n$ be the code that encodes a message $b' \in \{0,1\}^{L_2}$ as $\mathcal{L}(b)$, where b is obtained by padding b' with 0's to obtain $b \in \{0,1\}^k$. Formally, $\mathcal{L}_2(b') := \mathcal{L}(b)$, where $b \in \{0,1\}^k$ satisfies $b_i = b'_i$ for all $i \in L_2$ and $b_j = 0$ otherwise.

We will now show that if $\sum_{i=1}^k |G_i| \geq \delta nk/2$, then there exists a set $L_2 \subseteq [k]$ of size $|L_2| \geq k/2$ such that $\mathcal{L}_2$ is a linear $(2,\Omega(\delta \cdot \frac{d}{d+k}))$-wLDC. Recall that each G_i is a bipartite matching on $[n]\times P$, where $P = \{p = (u,v) : \deg_H(p) \geq d\}$, where $H = \cup_{i=1}^k H_i$. First, we can furthermore assume that each $p \in P$ appears not just in at least d edges across all G_i's, but also in at most $2d$ edges. Indeed, if some p violates this condition and has $t > d$ edges, then we split p into $t' = \lfloor t/d \rfloor$ new elements $p_1,\dots,p_{t'}$, each of which is adjacent to *exactly* d of the original edges, and then we connect the "residual" $t - t'd < d$ edges to p_1. Each new p_h obtained by splitting p now appears in exactly d edges across all G_i's except for p_1, which appears in at least d edges and at most $2d$ edges.

Next, partition $[k]$ into $L_2 \cup R_2$, and without loss of generality assume $|L_2| \geq k/2$. For $i \in L_2$, let G'_i denote the graph on n vertices with edges $E_i = \{(u,v) : \exists p \in P, j \in R_2, (u,p) \in G_i, (v,p) \in G_j\}$. Observe that $\sum_{i\in L_2} |G'_i| \geq \Omega(\delta nkd)$ in expectation over a random partition $L_2 \cup R_2$, and hence there exists such a partition $L_2 \cup R_2$ with $\sum_{i\in L_2} |G'_i| \geq \Omega(\delta nkd)$.

Next, we observe that for any vertex $u \in [n]$ and $i \in L_2$, u has degree at most $2d + k$ in G'_i. Indeed, since the G_i's are matchings and each p appears in at most $2d$ edges, it follows that for each u, there are at most $2d$ edges (u,v) in G'_i formed from the edge (u,p) in G_i. Second, for each v, there are at most k edges (u,v) in G'_i, as these can only be formed from the edges (v,p) in G_j, for $j \in R_2$, and each G_j is a matching so there is at most one edge per choice of $j \in R_2$. Hence, each G'_i has a matching M'_i of size at least $\Omega(|G'_i|/(d+k))$, and so $\sum_{i=1}^k |M'_i| \geq \Omega(\delta nk \cdot \frac{d}{d+k})$.

Finally, for each $i \in L_2$ and each edge $(u,v) \in M'_i$, it holds that $\mathcal{L}_2(b')_u \oplus \mathcal{L}_2(b')_v = b'_i$. Indeed, this is because $\mathcal{L}(b)$ satisfies $\mathcal{L}(b)_u \oplus \mathcal{L}(b)_p = b_i$ and $\mathcal{L}(b)_v \oplus \mathcal{L}(b)_p = b_j = 0$, where $p \in P$ is the shared pair used to add (u,v) to G'_i in the definition, $j \in R_2$, and $(u,p) \in G_i, (v,p) \in G_j$. We have thus shown that if $\sum_{i=1}^k |G_i| \geq \delta nk/2$, then $\mathcal{L}_2$ is a linear $(2,\Omega(\delta \cdot \frac{d}{d+k}))$-wLDC.

Constructing $\mathcal{L}_3$. Let $L_3 \subseteq [k]$ be a subset of size $|L_3| \geq k/2$ to be specified later. Let $\ell = \sqrt{n/k}/c$ for a sufficiently large constant c, and identify $N = \binom{2n}{\ell}$ with the collection of sets $\binom{[n]\times[2]}{\ell}$. We let $\mathcal{L}_3\colon \{0,1\}^{L_3} \to \{0,1\}^N$ be the code that encodes a message $b' \in \{0,1\}^{L_3}$ with the string $\mathcal{L}_3(b')$, where the S-th entry, for $S \in \binom{[n]\times[2]}{\ell}$, is

$$\mathcal{L}_3(b')_S := \Big(\bigoplus_{u^{(1)}\in S} \mathcal{L}(b)_u\Big) \oplus \Big(\bigoplus_{v^{(2)}\in S} \mathcal{L}(b)_v\Big),$$

where $b \in \{0,1\}^k$ satisfies $b_i = b'_i$ for all $i \in L_3$ and $b_j = 0$ otherwise.

We now argue that if $\sum_{i=1}^k |H'_i| \geq \delta nk/2$, then there exists a set $L_3 \subseteq [k]$ of size $|L_3| \geq k/2$ such that $\mathcal{L}_3$ is a linear $(2,\Omega(\delta^2/d))$-wLDC. Recall that each H'_i is a 3-uniform hypergraph matching

on n vertices, where $\deg_{H'}(\{u,v\}) \leq d$ for all $u,v \in [n]$, where $H' := \cup_{i=1}^{k} H'_i$. Partition $[k]$ into $L_3 \cup R_3$, and without loss of generality assume $|L_3| \geq k/2$. Following Section 11.1, we set $\ell = \sqrt{n/k}/c$ for a sufficiently large constant c and let $B_i \in \mathbb{R}^{N \times N}$ for $i \in L_3$ be the matrices defined in Definition 5.4.2.

Let G''_i denote the graph with adjacency matrix B_i, i.e., for $S,T \in [N]$, we have (S,T) as an edge in G''_i if $B_i(S,T) \neq 0$. By Lemma 11.1.6, the max degree of any vertex in G''_i is at most $2d$. Hence, G''_i contains a matching M''_i where $|M''_i| \geq \Omega(|G''_i|/d)$. Now, since $|H'| \geq \delta nk/2$, then by double counting, the number of clauses $C_1, C_2 \in H'$ with $|C_1 \cap C_2| \geq 1$ is at least $\Omega(\delta^2 nk^2)$. Thus, by picking a random partition and using Lemma 12.6.4, we find that $\sum_{i=1}^{k} |G''_i| \geq \Omega(D\delta^2 nk^2)$ in expectation, where $D = 2\binom{2n-\ell}{\ell-4}$, and hence there is a partition $L_3 \cup R_3$ achieving this. By applying Fact 3.6.1, we see that $D/N \geq \Omega(\ell^2/n^2)$, and so we have $\sum_{i=1}^{k} |M''_i| \geq \Omega(\delta^2 Nk/d)$, using that $\ell = \sqrt{n/k}/c$.

It is now straightforward to observe that, for each $i \in L_3$ and $(S,T) \in M''_i$, it holds that $b'_i = \mathcal{L}_3(b')_S \oplus \mathcal{L}_3(b')_T$; indeed, this is because $\mathcal{L}_3(b')_S \oplus \mathcal{L}_3(b')_T = \mathcal{L}(b)_S \oplus \mathcal{L}(b)_T = b_i \oplus b_j = b'_i$, as $b'_i = b_i$ and $b_j = 0$ because $j \in R_2$. We have thus shown that if $\sum_{i=1}^{k} |H'_i| \geq \delta nk/2$, then $\mathcal{L}_3$ is a linear $(2, \Omega(\delta^2/d))$-wLDC.

By Lemma 11.0.2, we thus have that either $\sum_{i=1}^{k} |G_i| \geq \delta nk/2$ or $\sum_{i=1}^{k} |H'_i| \geq \delta nk/2$. Hence, at least one of $\mathcal{L}_2$ and $\mathcal{L}_3$ must have the desired property, which finishes the proof. □

Remark 11.3.4 (A note on the linearity of $\mathcal{L}$). In Theorem 11.3.2, we assumed that the code $\mathcal{L}$ was linear. The reason that this assumption is necessary is because of the following. The constraints used to locally decode $\mathcal{L}_2$ and $\mathcal{L}_3$ are obtained by XORing two clauses C_1 and C_2 in the original set of local constraints defining $\mathcal{L}$. We then observe that by using $C_1 \oplus C_2$, we can decode, e.g., $b_i \oplus b_j$, and so by setting $\sim k/2$ of the b_j's to be hardcoded to 0, we have many constraints to recover b_i. The issue for nonlinear codes is that this "hardcoding" procedure does not work, as even though we can set b_j to be 0, the individual constraints C_1 and C_2 are only guaranteed to decode b_i and b_j, respectively, *in expectation* over a random choice of $b \in \{0,1\}^k$. Thus, when we hardcode some bits, we are no longer guaranteed that the derived constraint $C_1 \oplus C_2$ decodes b_i in expectation over the remaining "free" bits b_i for $i \in L$.

Chapter 12

Exponential Lower Bounds for 3-Query Locally Correctable Codes

In this chapter, we prove Theorems 8 to 10. We first give an overview of the strategy that leads to the proofs of Theorems 8 to 10. We will then give a standalone proof of Theorem 9, which is substantially simpler than the more general cases handled by Theorems 8 and 10. Then, as a warmup to the proof of Theorem 8, we give a proof sketch of a $n \geq \tilde{\Omega}(k^4)$ lower bound for linear 3-LCCs. Finally, we prove Theorems 8 and 10.

12.1 The proof strategy

We will start by giving a high-level overview of the proof strategy that we will use to prove Theorems 8 to 10. We will focus on the case of linear 3-LCCs, i.e., the case of Theorem 8, as well as on the case of $\mathbb{F} = \mathbb{F}_2$. Without loss of generality, we can assume that $\mathcal{L}$ is a systematic linear map $\mathcal{L}\colon \{-1,1\}^k \to \{-1,1\}^n$, so that the first k bits in any codeword are the message bits themselves, i.e., for any $b \in \{-1,1\}^k$, $x = \mathcal{L}(b)$ satisfies $x_i = b_i$ for all $i \in [k]$. We will use the notation $\gtrsim$ and $\lesssim$ to suppress a multiplicative polylog(n) factor.

The Kikuchi matrix method. Our proof uses the Kikuchi matrix method developed in this thesis. This method works in two steps: (1) formulate a hypergraph possessing some relevant structure as a family of satisfiable XOR formulas, and, (2) construct a spectral refutation (i.e., a certificate of unsatisfiability) of a randomly chosen member of this family. The spectral refutations in the second step rely on appropriate *Kikuchi* matrices — a term that we have been loosely using to describe induced subgraphs of an appropriately chosen Cayley graph associated with the hypergraph. The success of the spectral refutation naturally relies on the structure of the XOR instances. The power of the method comes from the ease (at least in hindsight, given Parts I and II and Chapter 11, i.e., [GKM22, HKM23, AGKM23]) in identifying the relevant combinatorial structure that is sufficient for the success of the spectral refutations.

Our proof can be seen as an upgrade on the methods we developed in Chapters 2 and 11, which we used to show a lower bound of $n \geq \tilde{\Omega}(k^3)$ on the block length n of a 3-query LDC (and therefore also a 3-query LCC) of dimension k and constant distance. The key conceptual idea that helps us move beyond the cubic to an exponential lower bound for 3-LCCs (a bound that provably cannot hold for 3-LDCs [Efr09, Yek08]) is a new family of XOR instances that crucially exploits

the additional structure in LCCs. Our new family of XOR instances is produced by performing a certain structured variant of *low-width* resolution (well-studied in proof complexity [Gri01, Sch08]) on the "basic" family. We call this process *long chain derivations*.

In the following, we will first recall the conceptual crux of the lower bound for q-LDCs in Section 2.3 and Chapter 11 and then use it to motivate our approach for 3-LCCs.

12.1.1 The naive XOR instance and LDC lower bounds

To begin, we will summarize the approach of Section 2.3 and Chapter 11 for the case of q-LDCs. Let us start by recalling the combinatorial characterization (formalized as the *normal form* in Definition 3.3.9). A code $\mathcal{L}\colon \{-1,1\}^k \to \{-1,1\}^n$ is a (q,δ)-LDC if for every $1 \le i \le k$, there exists a q-uniform hypergraph matching H_i over $[n]$ of size δn such that for every $b \in \{-1,1\}^k$ and codeword $x = \mathcal{L}(b)$, for every $i \in [k]$ and every $C \in H_i$, it holds that $x_C = b_i$. The combinatorial characterization above can be easily seen to be equivalent to the satisfiability of a family of q-XOR instances.

Observation 12.1.1 (LDCs and a Family of XOR Instances). Let $H_1, H_2, \ldots, H_k$ be q-uniform hypergraph matchings on $[n]$ of size δn. For every $b \in \{-1,1\}^k$, define the following q-XOR instance Ψ_b in n variables $x_1, x_2, \ldots, x_n$.

$$\forall i \in [k],\ \forall C \in H_i\ , x_C = b_i\,. \tag{12.1}$$

Then, there exists a (normal form) linear LDC $\mathcal{L} : \{-1,1\}^k \to \{-1,1\}^n$ described by the collection of q-uniform matchings $H_1, H_2, \ldots, H_k$ on $[n]$ if and only if Ψ_b is satisfiable for every $b \in \{-1,1\}^k$.

If $\mathcal{L}$ is a (q,δ)-LDC described by matchings $H_1, H_2, \ldots, H_k$, then $x = \mathcal{L}(b)$ satisfies all the constraints in Ψ_b. Conversely, if Ψ_b is satisfiable for every b, then one can easily construct a linear map $\mathcal{L}$ (easily seen to be a linear (q,δ)-LDC) where $\mathcal{L}(b)$ is some satisfying assignment to Ψ_b.

The main idea of Section 2.3 and Chapter 11 is to show that for any collection of δn-size q-matchings $H_1, H_2, \ldots, H_k$, if k is large enough as a function of n, then for a randomly chosen b, Ψ_b is unsatisfiable with high probability. This implies an upper bound on k. Now, when b is random, Ψ_b is XOR formula generated via $k \ll n$ bits, i.e., much smaller than the number of variables. Thus, a naive union bound argument cannot establish unsatisfiability of Ψ_b. In Section 2.3 and Chapter 11, we established unsatisfiability of Ψ_b for a random b via a *spectral refutation* using *Kikuchi* matrices.

Spectral refutations for Ψ_b. Let us now recall how the spectral refutation in Section 2.3 and Chapter 11 works. For our purpose of illustrating the conceptual idea, we will focus on the simpler setting of even q and sketch the proof that $k \le \tilde{O}(n^{1-2/q})$ for q-LDCs, which we saw in Section 2.3.

First, we observe that for the XOR instance Ψ_b, there is an associated "instance polynomial" $\Psi_b(x) := \sum_{i=1}^{k} \sum_{C \in H_i} b_i x_C$. We note that $\Psi_b(x)$ is the number of constraints satisfied by x minus the number of constraints violated, and thus Ψ_b is unsatisfiable if and only if $\mathrm{val}(\Psi_b) := \max_{x \in \{-1,1\}^n} \Psi_b(x)$ is less than $\sum_{i=1}^{k} |H_i| = k \cdot \delta n$. Thus, to show that Ψ_b is unsatisfiable, we will bound $\mathrm{val}(\Psi_b)$.

To do this, we define a Kikuchi matrix whose quadratic form is equal to $\Psi_b(x)$ using the strategy we developed in Chapter 2.

Definition 12.1.2 (Kikuchi matrix and graphs, Definition 2.1.1 restated)**.** Let $C \in \binom{[n]}{q}$, let ℓ be a parameter, and let $N := \binom{n}{\ell}$. Let $A_C \in \{0,1\}^{N \times N}$ be the matrix indexed by sets $S \in \binom{[n]}{\ell}$ where

$A_C(S,T) = 1$ if $S \oplus T = C$, and 0 otherwise. Let $A_i := \sum_{C \in H_i} A_C$, and let $A := \sum_{i=1}^{k} b_i A_i$. We naturally interpret (and by abuse of notation, also call) A_C, A_i and A as adjacency matrices of "Kikuchi graphs" on the vertex set $\binom{[n]}{\ell}$.

Observe that A_C is a matching on vertex set $\binom{[n]}{\ell}$ of size $D = \binom{n-q}{\ell-q/2}\binom{q}{q/2}$ (see Proposition 2.1.2). For any $x \in \{-1,1\}^n$, let $x^{\odot \ell}$ denote the ℓ-wise monomial vector indexed by $S \in \binom{[n]}{\ell}$ with corresponding entry equal to x_S. Then, ${x^{\odot \ell}}^\top A_C x^{\odot \ell} = D x_C$. Consequently, ${x^{\odot \ell}}^\top A x^{\odot \ell} = D \Psi_b(x)$. Thus, if $x \in \{-1,1\}^n$ satisfies Ψ_b, then we have the following inequality that upper bounds k in terms of $\|A\|_2$:

$$k\delta n = \Psi_b(x) \leq \frac{1}{D} \left\|x^{\odot \ell}\right\|_2^2 \|A\|_2 = \frac{\binom{n}{\ell}}{D} \|A\|_2 \leq O((n/\ell)^{q/2}) \|A\|_2 \,. \tag{12.2}$$

We now choose $b \in \{-1,1\}^k$ uniformly at random and consider $A = \sum_i b_i A_i$, which is a matrix Rademacher series of the A_i's. By the Matrix Khintchine inequality (Fact 3.4.2), $\|A\|_2 \leq O(\sqrt{\log N}) \left\|\sum_i A_i^2\right\|_2^{1/2}$ with high probability.

A combinatorial proxy for $\|A\|_2$. Let Δ_i be the maximum degree of any node in the Kikuchi graph A_i, and let $\Delta = \max_{1 \leq i \leq k} \Delta_i$. Then, we can naively bound $\left\|\sum_i A_i^2\right\|_2 \leq \sum_i \|A_i\|_2^2 \leq k\Delta^2$. Thus, the maximum degree of the A_i's naturally controls the spectral norm of A as $\|A\|_2 \leq \Delta \cdot O(\sqrt{k\ell \log n})$.

Let us now investigate bounds on Δ. Since for each $C \in H_i$, A_C contributes D edges to A_i, the average degree of A_i is clearly $\delta n D / N \sim n(\ell/n)^{q/2}$. Thus, $\Delta \geq O(1) \max\{1, n(\ell/n)^{q/2}\}$. If Δ happens to be equal to this minimum possible value, then substituting it in Eq. (12.2) yields:

$$k\delta n \leq O(1) \left(\frac{n}{\ell}\right)^{q/2} \sqrt{k\ell \log n} \cdot \max\{1, n(\ell/n)^{q/2}\} \,,$$

which implies that $k \leq O(\ell \log n) \cdot \max\{n^{q-2}/\ell^q, 1\}$. This is minimized at $\ell = n^{1-2/q}$ to give the lower bound of $k \leq \tilde{O}(n^{1-2/q})$, i.e., $n \geq \tilde{\Omega}(k^{q/(q-2)})$.

Handling irregularities: row pruning via polynomial concentration. We will now (for the first time in the argument) use that the H_i's are matchings to argue that while the A_i's are certainly not approximately regular (i.e., max degree Δ_i at most a polylog(n) factor larger than the average-degree), there is only a small fraction of nodes in any A_i that have a large degree. Of course, a small fraction of rows can still cause $\|A\|_2$ to be too large. In order to circumvent this issue, we observe that the argument in Eq. (12.2) works even if we were to replace $N \|A\|_2$ (maximum over arbitrary quadratic forms) by $\|A\|_{\infty \to 1}$ (maximum over quadratic forms on ± 1-coordinate vectors). The latter quantity is insensitive to dropping a small fraction of rows since ± 1-coordinate vectors when restricted to a small number of rows must have correspondingly small ℓ_2-norm.

To prove that only a small fraction of nodes can have a large degree in any A_i, we view the degree of any node S as a polynomial in the corresponding indicator variables $z \in \{0,1\}^n$ with $\sum_i z_i = \ell$ and use tail inequalities for low-degree polynomials (that generalize concentration of Lipschitz functions) of Kim and Vu and extensions [KV00, SS12] to bound the chance that it takes a value polylog(n) times the average. This relies on establishing strong bounds on the expected partial derivatives of the degree polynomial by using that the H_i's are matchings.

The key heuristic: high density for Kikuchi graphs at low levels. Let's summarize the crucial steps of the above argument as follows: (1) q-LDCs naturally yield XOR instances of arity q, (2)

to obtain our lower bound, we need that the Kikuchi matrices A_i corresponding to a matching H_i are approximately regular (after dropping a negligible fraction of rows), and (3) the argument can only yield a bound of the form $k \lesssim \ell$ where ℓ is the smallest level of the Kikuchi graphs A_i with an average degree $\gg 1$. More precisely, if there are m_i constraints of arity q in H_i, then the threshold ℓ is the smallest integer satisfying $m_i(\ell/n)^{q/2} \gg 1$ for all $i \in [k]$. Note that this threshold ℓ increases as q increases.

We assert that even though the argument Chapter 11 for the case when $q = 3$ requires more work (in both the design of the Kikuchi matrix itself and its analysis), the heuristic above continues to hold. Let us also note that ensuring approximate regularity is usually the trickiest aspect of the proof. In particular, while the heuristic above makes sense for all odd q (and not just $q = 3$), and in Chapter 11 we failed to obtain an improved lower bound for odd $q > 3$ because we were unable to find an appropriate "decomposition" that ensures approximate regularity of the resulting Kikuchi matrices.

Thus, in order to obtain an exponential lower bound, as in Theorem 8, via the schema above, we must construct Kikuchi graphs that have constant density (i.e., average degree) at much a lower level ℓ. Specifically, we will need to be able to take $\ell = \text{polylog}(n)$.[1]

12.1.2 Long chain derivations: stronger spectral refutations by increased density

Given the key heuristic above, we now show how to build XOR instances from 3-LCCs that yield constant density Kikuchi matrices at level $\ell = \text{polylog}(n)$. Our instances will balance two opposing concerns. On the one hand, the constraints will be of large arity (in fact, $O(\log n)$ arity) which, given the discussion above, hurts the density at lower levels. Nonetheless, we will show that the number of higher arity constraints that we produce grows fast enough to compensate for this and gives us an overall increase in density at lower ℓ. We note (with the hope of pointing the reader to the trickiest part of the proof that motivates all our setup) that the analysis of "row pruning" i.e., arguing approximate regularity after removing a negligible fraction of rows, will get significantly more involved and motivates all our design choices. This includes the specific type of Kikuchi matrices that we will choose and a new decomposition for the constraints that, while a bit unnatural at the outset, helps guarantee approximate regularity. Let us see these ideas in more detail next.

Like 3-LDCs, 3-LCCs can, without loss of generality, be assumed to be $(3, \delta)$-normal. Thus, for any 3-LCC $\mathcal{L}\colon \{-1,1\}^k \to \{-1,1\}^n$, there are 3-uniform hypergraph matchings $H_1, \ldots, H_n$ on $[n]$, each of size δn, such that for every $b \in \{-1,1\}^k$, $u \in [n]$, and $C \in H_u$, the encoding $x = \mathcal{L}(b)$ satisfies $x_C = x_u$. Note that the key difference between LCCs and LDCs is that here we have a "local correcting" hypergraph H_u for each $u \in [n]$, instead of only a hypergraph for each $i \in [k]$ in the case of LDCs.

The naive XOR instances. Similar to Observation 12.1.1, the combinatorial characterization

[1] We note that while our lower bounds appear to get weaker as ℓ grows, generic convergence results about the Kikuchi matrices imply that taking $\ell \sim n$ and bounding Ψ_b in terms of $\|A\|_2$ yields the *optimal* bound on k, whatever it may be! The reason the current argument (which is likely suboptimal) does not extend beyond $\ell = n^{1-2/q}$ is the potentially superfluous $\sqrt{\log N}$ multiplicative loss in the matrix Khintchine inequality. Investigating when this $\sqrt{\log N}$ factor (which is tight in the worst case) can be removed is the topic of an ongoing research effort in random matrix theory [BBH23] and is naturally related to other problems such as resolving the matrix Spencer conjecture [Zou12, Mek14].

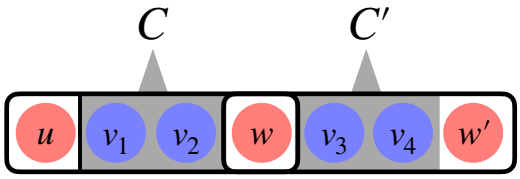

Figure 12.1: A 2-chain with head u. Note that $C \cup \{w\} \in H_u$ and $C' \cup \{w'\} \in H_w$, and that $x = \mathcal{L}(b)$ satisfies $x_C x_w = x_u$ and $x_{C'} x_{w'} = x_w$, and therefore $x_C x_{C'} x_{w'} = x_u$.

yields that the XOR instance with constraints $x_C = x_u$ for every $C \in H_u$ and $u \in [n]$ (where on the right-hand side, we set $x_u = b_u$ whenever $u \in [k]$) is satisfiable for every $b \in \{-1,1\}^k$. If we focus only on the constraints corresponding to H_u for $u \in [k]$ (i.e., the "systematic" bits in the codeword), then we recover the same XOR instance as in the case of 3-LDCs and our method from above yields $k \leq \tilde{O}(n^{1/3})$. To improve on this significantly lossy formulation, we must make use of the additional constraints H_u for $u \notin [k]$. More specifically, if we were to only use the hypergraphs H_u for $u \in [k]$, then any lower bound we could prove would hold for LDCs as well, and in particular one could not hope to prove Theorem 8, which is false for LDCs.

Long chain derivations. We now show how to use the additional constraints in order to build a higher arity XOR instance that is (1) approximately regular (after an appropriate decomposition), and (2) results in high-density Kikuchi graphs at polylog(n) levels. We will construct higher arity XOR instances that use the additional constraints above using a structured variant of low-width XOR resolution [Gri01, Sch08] that we call *long chain derivations*.

Let us start by forming extra constraints via 2-chains. Observe that for any $u \in [n]$ and $C \in H_u$, we have that for any $b \in \{-1,1\}^k$, $x = \mathcal{L}(b) \in \{-1,1\}^n$ satisfies the equation $x_u x_C = 1$. Now, let us choose $w \in C$ and $C' \in H_w$. We also have that $x_w x_{C'} = 1$. As $x_C = x_{C \setminus \{w\}} x_w$, it follows that the "derivation" $x_u x_{C \setminus \{w\}} x_{C'} = 1$ also holds, since $x_w^2 = 1$. We shall call such a constraint a "2-chain" — it connects two constraints intersecting in one variable. We can think of such a 2-chain as a tuple (u, C, w, C', w'), where $C \cup \{w\} \in H_u$ and $C' \cup \{w'\} \in H_w$, and this yields the constraint $x_C x_{C'} x_{w'} = x_u$ (see Fig. 12.1).

Consider now the 2-chains $\cup_{i \in [k]} \mathcal{H}_i^{(2)}$, i.e., 2-chains of the form (i, C, w, C', w') where $i \in [k]$. Then, the constraints have the form $x_C x_{C'} x_{w'} = b_i$, so they decode the i-th independent bit b_i. We have thus formed a new set of constraints with "right-hand side" b_i.

A heuristic calculation. Let us now do a heuristic calculation (that ignores the key issue of approximate regularity) to see if we improve the density at lower Kikuchi levels by taking the XOR instances corresponding to 2-chains. For any fixed "head" $i \in [k]$, there are $(3\delta n)^2$ 2-chains. This is because we have δn choices for $C \cup \{w\} \in H_i$, followed by 3 ways to choose w from $C \cup \{w\}$, and then similarly $3\delta n$ choices in total for (C', w'). Let $\mathcal{H}_i^{(2)}$ denote the set of 2-chains with head i. We have thus produced $\sim n^2$ constraints and each constraint has arity 5,[2] as $|C| = |C'| = 2$.

The Kikuchi matrix in Definition 12.1.2 only makes sense for even q, but let us still do a "pretend" calculation of the relative density for the arity 5 constraints we have produced. This

[2]Some constraints may have additional variable cancellations and thus have arity < 5. However, as the density gets worse as the arity increases, this is only "better" for us.

can be made precise with a slightly more sophisticated Kikuchi matrix (which we have seen in Chapter 11), so this is still a meaningful heuristic.

The density (i.e., average degree) for the Kikuchi matrix A_i is now $n^2(\ell/n)^{q/2} \sim n^2(\ell/n)^{5/2} \sim \ell^{2.5}/n^{0.5}$. This density is $\gg 1$ whenever $\ell \gg n^{1/5}$, so one might expect to obtain a bound of $k \lesssim n^{1/5}$ (beating the $n^{1/3}$ bound for the naive XOR instance) when working with 2-chains — a construction that crucially relies on additional structure in 3-LCC! While there are lot of details that we have simply ignored in doing this calculation, it does suggest that we are able to achieve a constant-density Kikuchi matrix A_i at a lower level ℓ. A similar calculation (that we will omit here) for chains of larger length, say r, shows that the smallest level ℓ at which we can obtain constant density Kikuchi matrices is $\ell \sim n^{1/2r}$, and this suggests that we might be able to obtain constant density at level $\ell = \text{polylog}(n)$ if we work with $r \sim \log n$ length chains.

In Section 12.3, as a warmup to our somewhat technical proof of the main theorem, we present a complete analysis of the 2-chains (with extended commentary) to obtain a $k \leq \tilde{O}(n^{1/4})$ bound (giving a polynomial improvement on the $\sim n^{1/3}$ lower bound on 3-LDCs already!) in order to illustrate (a simplified version of) the set of new tools that go into the analysis.

12.1.3 From the heuristic to a proof

In the remaining part of this overview, we briefly discuss the technical tools we develop to turn the above heuristic calculation into a full proof. We note that the actual parameters become rather delicate. For readers familiar with the literature on random CSP refutation (our setting resembles semirandom XOR refutation with complicated correlations in the right-hand sides), this is similar to the analysis getting rather delicate when dealing with XOR instances with super-constant arity.

Setting up the Kikuchi matrix. The instances produced by forming r-chains yield XOR instances of (odd) arity $2r+1$. We build a different Kikuchi matrix by first applying the "Cauchy–Schwarz" trick — a standard idea in CSP refutation also utilized in Section 5.1 and Chapter 11. In our case, the XOR instance produced after this trick corresponds to constraints formed by joining two r-chains at their "tails" whenever the tails match. We choose a variant of the Kikuchi matrix for the "Cauchy–Schwarzed instance" except for the key difference that it is indexed by $2r$-tuples of sets of size ℓ (instead of a single set of size ℓ) in the sketch above.

Regularity decomposition. If $H_1, H_2, \ldots H_n$ are such that no pair of variables appears in more than one hyperedge ("no heavy pairs") across all the H_i's, then it turns out that the resulting Kikuchi matrices satisfy approximate regularity after pruning a negligible fraction of rows. This "no heavy pair" property holds, e.g., if H_i's are uniformly random and independent hypergraph matchings of size δn. It also holds in the design case (Theorem 9) by assumption.

However, when the H_i's are arbitrary, and in particular when there are "heavy pairs" (i.e. pairs of variables that appear in $\gg \log n$ hyperedges across the H_i's), the resulting Kikuchi matrices are *far* from being approximately regular. Our key technical idea is a new *decomposition* procedure that operates directly on the chains. Such a decomposition procedure partitions the chains into $\sim r$ different groups such that each group admits a (different, appropriately defined) Kikuchi matrix that satisfies approximate regularity. Regularity decompositions have been used many times already in this thesis (see Sections 5.2, 7.3 and 11.0.1). However, our notion of regularity is (necessarily) significantly weaker (we call it "smoothed partitioning") that, unlike the method in, say, Section 5.2, does not "by design" ensure approximate regularity of the Kikuchi matrices after

removing only a negligible fraction of rows. Instead, our argument for approximate regularity relies on combining the guarantees of the decomposition with (1) an appropriate choice of Kikuchi matrix for each piece in the partition, and (2) the structure in the chains arising by virtue of H_i's being matchings.

Polynomial concentration: bounding expected derivatives. Our main technical step (the subject of Section 12.7) is proving that our weak notion of regularity combined with the fact that H_i's are matchings is enough to control expected partial derivatives of the "degree-polynomial" that computes the degrees of nodes in the Kikuchi graph.

Our original proof of approximate regularity of the Kikuchi graph from the smoothed partitioning of the chains (which appeared in [KM24a]) used a "partite" version of the Kim–Vu inequality [KV00, SS12]. This original proof results in a final bound of $k \leq O(\log^8 n)$, or a lower bound of $n \geq 2^{\Omega(k^{1/8})}$. In the proof presented in this thesis, we shall incorporate the second moment method row pruning argument of [Yan24], which saves a few polylog(n) factors and results in a $2^{\Omega(k^{1/4})}$ bound.

We note that the analysis of the expected partial derivatives of the "degree polynomial" (which we use to prove approximate regularity) and the interplay of these bounds with our decomposition of chains is the key technical part (and the focus of Section 12.7) of our proof. In order to illustrate this technical part in a "base" case that still captures some of the complications, we present the case of 2-chains as a warmup in the next section.

12.2 Proof of Theorem 9

In this section, we prove Theorem 9. The proof is substantially simpler than the proofs of Theorems 8 and 10. The proof here will be self-contained, and will also serve as a partial warmup to Theorems 8 and 10.

The proof presented follows the overall blueprint described in Sections 12.1 and 12.3, although we will present it via a slightly different lens. Namely, we will use the design 3-LCC $\mathcal{L}$ to construct a 2-query linear locally decodable code, and then we will apply the lower bound of [GKST06].[3] We will incorporate the clever second moment method proof of the row pruning step due to [Yan24], which is very similar to the edge deletion method of [HKM23] done in the context of semirandom and smoothed CSP refutation (Part I). The key reason that we save the final $\log n$ factors is by using a more carefully chosen Kikuchi graph, a sharp accounting of binomial coefficients, and the crucial use of the fact that in the design case, the hypergraph matchings are perfect.

Let us now proceed with the proof. Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a design 3-LCC. Namely, there exists a 4-uniform hypergraph design $H \subseteq \binom{[n]}{4}$ such that for all $C \in H$, $\sum_{v \in C} x_v = 0$ for all $x \in \mathcal{L}$. Without loss of generality, we may assume that $\mathcal{L}$ is systematic, i.e., for each $b \in \{0,1\}^k$, $\mathcal{L}(b)_i = b_i$. To bound k, we will give another linear map $\mathcal{L}'\colon \{0,1\}^n \to \{0,1\}^{2nN}$, where $N = \binom{n}{\ell}$ for some parameter $\ell = (1 + o(1)) \log_2 n$, and we will show that $\mathcal{L}' \circ \mathcal{L}\colon \{0,1\}^k \to \{0,1\}^N$ is

[3] The proof overview in Sections 12.1 and 12.3 is presented using the perspective of spectral refutation and matrix concentration bounds, even though the final proof in the case of linear LCCs (Theorem 8) can be phrased as a reduction to a 2-LDC. Here, we present the proof as a reduction as it is a more accessible and combinatorial analysis, although we note that one could prove the same result using matrix concentration as well. The proof of the nonlinear case (Theorem 10) requires the spectral refutation perspective.

a 2-query linear locally decodable code with $\delta = \frac{1}{2}(1 - o(1))$. We can then apply Fact 3.3.4 to conclude that $(1 - o(1))k \leq 2\delta k \leq \log_2 N \leq (\ell + 1)\log_2 n$ where $\ell = (1 + o(1))\log_2 n$.

For each $u \in [n]$, we let H_u denote the 3-uniform hypergraph defined from H as specified in Remark 3.3.12, i.e., $H_u = \{C : C \cup \{u\} \in H\}$. As shown in Remark 3.3.12, H_u is a matching of size $\delta n = \frac{n-1}{3}$, i.e., $\delta := \frac{1}{3} - \frac{1}{3n}$.

Step 1: forming long chain derivations. In the first step of the proof, we use the initial system of constraints H to define a larger system of constraints, called long chain derivations.

Definition 12.2.1. Let $H_1, \ldots, H_n$ be the 3-uniform hypergraph matchings defined from the 4-design H. An r-chain with *head* u_0 is an ordered sequence of vertices of length $3r + 1$, given by $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r)$, such that all the v_h's are *distinct*[4] and for each $h = 0, \ldots, r - 1$, it holds that $\{v_{2h+1}, v_{2h+2}, u_{h+1}\} \in H_{u_h}$. We let $\mathcal{H}_u^{(r)}$ denote the set of r-chains with head u.

We let $C_L = (v_1, v_3, v_5, \ldots, v_{2(r-1)+1})$ denote the "left half" of the chain, and $C_R = (v_2, v_4, v_6, \ldots, v_{2(r-1)+2})$ denote the "right half". We call u_r the "tail".

We observe that $\mathcal{H}_u^{(r)}$ has size at most $(6\delta n)^r$ and size at least $(6\delta n - 4r)^r$. Indeed, the upper bound follows because, given a partial chain $(u_0, v_1, v_2, \ldots, u_h)$, there are exactly $6\delta n$ choices of $(v_{2h+1}, v_{2h+2}, u_{h+1})$ (which we note are ordered), and the lower bound follows because there are always at least $6\delta n - 4h \geq 6\delta n - 4r$ choices, as each vertex v can appear in either the first or second spot in at most 2 *ordered* hyperedges in $H_{u'}$ for any $u' \in [n]$.

The following observation asserts that the system of linear equations given by the chains are satisfied by every $x \in \mathcal{L}$.

Observation 12.2.2. Let $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$ be an r-chain, with left half C_L and right half C_R. Then, for any $x \in \mathcal{L}$, it holds that $x_{u_r} + \sum_{v \in C_L} x_v + \sum_{v \in C_R} x_v = x_{u_0}$.

Proof. For any chain C, we have that for all $h = 0, \ldots, r - 1$, it holds that $\{v_{2h+1}, v_{2h+2}, u_{h+1}\} \in H_{u_h}$, which implies that $x_{v_{2h+1}} + x_{v_{2h+2}} + x_{u_{h+1}} = x_{u_h}$ for all $x \in \mathcal{L}$. By taking the product over all these equations, Observation 12.2.2 follows. □

Step 2: defining the Kikuchi graphs. In this step, we will define two linear maps $\mathcal{L}_1 \colon \{0,1\}^n \to \{0,1\}^L$ and $\mathcal{L}_2 \colon \{0,1\}^n \to \{0,1\}^R$, where $L = \binom{[n]}{\ell} \times [n]$, $R = \binom{[n]}{\ell}$, and ℓ is a parameter, as follows. Let $\mathcal{L}_1(x)_{(S,v)} := x_v + \sum_{v' \in S} x_{v'}$, and let $\mathcal{L}_2(x)_T := \sum_{v' \in T} x_{v'}$. Note that $|L| = nN$ and $|R| = N$, where $N = \binom{n}{\ell}$.

Now, for each $u \in [n]$, we will use the set of r-chains $\mathcal{H}_u^{(r)}$ to define a bipartite graph G_u with left vertices L and right vertices R such that, for each edge $((S, v), T)$ in G_u, it holds that $\mathcal{L}_1(x)_{(S,v)} + \mathcal{L}_2(x)_T = x_u$. This graph G_u will be the following Kikuchi graph.

Definition 12.2.3 (Kikuchi graph)**.** Let ℓ be a parameter, to be determined later, and let G_u be the graph with left vertex set $L = \binom{[n]}{\ell} \times [n]$ and right vertex set $R = \binom{[n]}{\ell}$. For a chain $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$ with left half C_L and right half C_R, we add an edge $((S, w), T)$ to G_u "labeled" by C if $S = C_L \cup U$, $T = C_R \cup U$ where $|U| = \ell - r$[5] and $w = u_r$. Two distinct chains may produce the same edge — we add edges with multiplicity.

[4] In this section only, we will enforce that all the v_h's are distinct, as this will be slightly more convenient.

[5] Note that here we will use that all the v_h's are distinct, so that $|C_L| = |C_R| = r$ and $|C_L| + |C_R| = 2r$.

We now make the following simple observations about the graph G_u.

Observation 12.2.4. For any chain $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$, the number of edges in G_u "labeled" by C is exactly $\binom{n-2r}{\ell-r}$.

In particular, the average left degree of G_u, denoted by $d_{u,L}$ is $\binom{n-2r}{\ell-r}/nN$, and the average right degree, denoted by $d_{u,R}$ is $\binom{n-2r}{\ell-r}/N$.

Proof. Let C_L be the left half of C and let C_R be the right half. Because all the v_h's are distinct, we have $|C_L| = |C_R| = r$ and $|C_L \cup C_R| = 2r$. It follows that the number of pairs $((S, w), T)$ such that $((S, w), T)$ is an edge in G_u labeled by C is simply the number of choices for the set U, which is a subset of $[n] \setminus (C_L \cup C_R)$ of size $\ell - r$. Thus, there are exactly $\binom{n-2r}{\ell-r}$ choices. □

Observation 12.2.5. For every edge $((S, w), T)$ in G_u and $x \in \mathcal{L}$, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}_2(x)_T = x_u$.

Proof. Suppose that $((S, w), T)$ in G_u is an edge labeled by the chain C, which has left half C_L and right half C_R. We then have that $w = u_r$, $u = u_0$, and $S = C_L \cup U$, $T = C_R \cup U$. Therefore,

$$\begin{aligned}\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}_2(x)_T &= x_{u_r} + \sum_{z \in S} x_z + \sum_{z \in T} x_z \\ &= x_{u_r} + \sum_{z \in C_L} x_z + \sum_{z \in C_r} x_z + \sum_{z \in U} (x_z + x_z) = x_{u_r} + \sum_{z \in C_L} x_z + \sum_{z \in C_r} x_z = x_u\,,\end{aligned}$$

where the last equality uses Observation 12.2.2. □

The plan for the remainder of the proof. Let us now take a brief moment to outline the steps for the remainder of the proof. To construct a 2-LCC, it suffices to show that G_u admits a matching M_u of size $\Omega(N)$. Indeed, if this were the case, then the matching M_u would be the matching that we require to invoke Fact 3.3.4 and thus finish the proof.

To show that G_u has a large matching, it suffices bound the maximum degree of the graph by d, as then G_u must admit a matching of size at least $|E(G_u)|/d$. However to do this, there are two issues to resolve. The most obvious issue is that the bipartite graph is unbalanced, i.e., $|L| = n|R|$, and so this prevents us from obtaining a matching of size $\Omega(|L|)$. This issue can be easily fixed by the following trick:[6] for each right vertex $T \in R$, we can create n copies of T, denoted by $T^{(1)}, \ldots, T^{(n)}$, and split the edges adjacent to T evenly across the copies. This decreases the average (and maximum) right degree by a factor of $(1 - o(1))n$, and fixes the issue.

The second, and much more challenging problem, is that the graph G_u need not be approximately biregular. Indeed, if the graph G_u was exactly *biregular*, then apply the above "splitting trick" would imply that the resulting graph has a *perfect* matching of size $nN/2$.

This irregularity issue is a common problem for Kikuchi matrices and has arisen many times in this thesis. The way to handle this issue is to show that G_u admits a subgraph G'_u that is *approximately* biregular and still contains a significant fraction of the edges of G_u, i.e., $|E(G'_u)| \geq \Omega(|E(G_u)|)$. This is the "row pruning" step (Section 2.3), which is so named because it involves pruning rows (and columns) of the adjacency matrix of G_u. This row pruning step is the crucial, and by far the most technical, component of the proof.

[6] This is a nice trick of [Yan24] that, while it does not affect the final bounds, saves a use of the Cauchy–Schwarz inequality and thus makes the graph G_u a bit simpler to describe.

Step 3: Finding a near-perfect matching in G_u. We now argue that G_u admits a degree-bounded subgraph G'_u containing $(1-o(1))|E(G_u)|$ edges. The strategy in Sections 12.1 and 12.3 is to use the moment method to argue that with high probability, a random left (or right) vertex of the graph has degree at most $O(d_{u,L})$ (or $O(d_{u,R})$) with high probability. Here, we will follow the approach of [HKM23, Yan24], which is to observe that it suffices to compute first and second moments only. Indeed, it is computing higher moments that causes the loss of several extra $\log n$ factors in the original proof of [KM24a], as compared to [Yan24].

The key reason we shall save the final $\log n$ factor is because the matchings H_u are nearly perfect, i.e., they have size δn where $\delta = \frac{1}{3} - \frac{1}{3n}$. This, combined with the careful choice of the matrix, allows us to take $\ell = O(r)$ instead of $\ell = O(r^2)$, which saves a $\log n$ factor. We note that in order to get the sharp constant achieved in Theorem 9, we need to show that G_u contains a *near-perfect* matching.

Let $\deg_{u,L}(S,w)$ denote the left degree of (S,w) in G_u, and let $\deg_{u,R}(T)$ denote the right degree of T in G_u. In the following lemma, we compute the first[7] and second moments of the degree functions. This lemma is the key technical lemma of the proof, and immediately implies the existence of a degree-bounded subgraph of G_u of comparable density, as we shall shortly see.

Lemma 12.2.6 (Second moment bounds for the left and right degree)**.** *Let ℓ be a parameter with $\ell \geq r$ such that $r, \ell = o(n^{1/4})$. Let G_u be the graph defined in Definition 12.2.3. Then, it holds that*

$$\mathbb{E}_{(S,w)}[\deg_L(S,w)^2] \leq (1+o(1)+\eta)\mathbb{E}_{(S,w)}[\deg_L(S,w)]\,,$$
$$\mathbb{E}_T[\deg_R(T)^2] \leq (1+o(1))\mathbb{E}_T[\deg_R(T)]\,.$$

Here, the $o(1)$ is $O(\ell^2)/n$ and $\eta = n/\binom{\ell}{r}$.

We note that when we apply Lemma 12.2.6, we will take $r = \frac{1}{2}\log_2 n + O(\log\log n)$ and $\ell = 2r-1$, which will end up satisfying the conditions with $\eta = 1/\mathrm{polylog}(n)$.

We postpone the proof of Lemma 12.2.6 to Section 12.2.1. Let us now use Lemma 12.2.6 to extract a near-perfect matching from G_u. We will assume that ℓ, r are chosen so that $\eta \leq 1/O(\log^2 n) = o(1)$, which will be the case when we choose parameters.

Using Lemma 12.2.6, we apply Chebyshev's inequality to observe that for the graph G_u:

1. There are at least $(1-o(1))|L|$ left vertices with degree $d_{u,L}(1\pm o(1))$. Let L'_u denote these left vertices.
2. There are at least $(1-o(1))|R|$ right vertices with degree $d_{u,R}(1\pm o(1))$. Let R'_u denote these right vertices.

Let $G'_u = G_u[L'_u, R'_u]$ be the induced subgraph. First, we observe that $|E(G'_u)| \geq (1-o(1))|E(G_u)|$. This is because there are at least $(1-o(1))d_{u,L}|L'_u| \geq (1-o(1))(1-o(1))d_{u,L}|L| \geq (1-o(1))|E(G)|$ edges in $G[L', R]$ and at least $(1-o(1))d_{u,R}|R'_u| \geq (1-o(1))(1-o(1))d_{u,R}|R| \geq (1-o(1))|E(G)|$ edges in $G[L, R']$, and therefore $G[L', R']$ must have at least $(1-o(1))|E(G)|$ edges. Furthermore, each left vertex in G' has degree at most $(1+o(1))d_{u,L}$, and similarly each right vertex has degree at most $(1+o(1))d_{u,R}$.

Recall that $n \cdot d_{u,L} = d_{u,R}$ and $|L| = |R| \cdot n$. Therefore, by making n copies $T^{(1)}, \ldots, T^{(n)}$ of each vertex T in R and splitting the edges equally across all copies (and doing the same induced transformation on G'_u), we can create a new bipartite graph G''_u with left vertex set L and right vertex set $R \times [n]$ where G''_u has max left (or right!) degree $(1+o(1))d_{u,L}$ and at least $(1-o(1))|E(G)|$

[7] Note that Observation 12.2.4 computes the first moments already.

edges. Therefore, G''_u contains a matching M_u of size at least $(1-o(1))|E(G)|d_{u,L} \geq (1-o(1))|L|$. Note that this matching is *nearly perfect*, as the graph G''_u has $2|L|$ vertices, $|L|$ left vertices and $|L|$ right vertices.

Step 4: proving the final bound. Recall that we began with a linear map $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ that is a design 3-LCC. We then built the maps $\mathcal{L}_1\colon \{0,1\}^n \to \{0,1\}^L$ and $\mathcal{L}_2\colon \{0,1\}^n \to \{0,1\}^R$, where $L = \binom{[n]}{\ell} \times [n]$ and $R = \binom{[n]}{\ell}$, and the matchings M_u for each $u \in [n]$ on the left vertex set L and the right vertex set $R \times [n]$. To do this, we needed to apply Lemma 12.2.6, which requires that $\ell, r = o(n^{1/4})$. We thus set $r = \lceil \frac{1}{2}\log_2 n + \Gamma \log_2 \log_2 n \rceil$ for a sufficiently large constant Γ and $\ell = 2r-1$, which satisfies the conditions. We additionally have $\eta = 1/\log_2^2 n$, as

$$\binom{\ell}{r} = \binom{2r-1}{r} \geq \frac{2^{2r-1}}{2r} \geq \frac{n \cdot 2^{\Gamma \log_2 \log_2 n}}{O(\log n)} \geq n \cdot (\log_2 n)^{\Gamma - 1 - o(1)} \geq n(\log_2^2 n)\,,$$

where we use that $\binom{2r-1}{t}$ is maximized at $t = r$ and $t = r-1$.

Let $\mathcal{L}'_2\colon \{0,1\}^n \to \{0,1\}R \times [n]$ be the map where $\mathcal{L}'_2(x)_{T^{(h)}} = \mathcal{L}_2(x)_T$, where $T^{(h)}$ is the h-th copy of T in $R \times [n]$. A simple corollary of Observation 12.2.5 is that, for any $x \in \mathcal{L}$, $u \in [n]$, and edge $((S,w),T^{(h)})$ in M_u, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}'_2(x)_{T^{(h)}} = x_u$. In particular, since $\mathcal{L}$ is systematic, for any $i \in [k]$, edge $((S,w),T^{(h)})$ in M_u, and $b \in \{0,1\}^k$, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}'_2(x)_{T^{(h)}} = x_i = b_i$.

Let $\mathcal{L}'\colon \{0,1\}^n \to \{0,1\}^{L \cup (R \times [n])} \cong \{0,1\}^{2nN}$ be the map where $\mathcal{L}'(x)_{(S,w)} = \mathcal{L}_1(x)$ and $\mathcal{L}'(x)_{T^{(h)}} = \mathcal{L}'_2(x)_{T^{(h)}}$. We have that $\mathcal{L} \circ \mathcal{L}'$ is linear map from $\{0,1\}^k \to \{0,1\}^{2nN}$ and that M_i is a matching of size $\geq (1-o(1))nN = \frac{1}{2}(1-o(1)) \cdot 2nN$ that decodes b_i. Therefore, by Fact 3.3.4, we conclude that $(1-o(1))k \leq \log_2 N \leq (\ell+1)(\log_2 n) = 2r \log_2 n = (1+o(1))(\log_2 n)^2$, which proves Theorem 9.

12.2.1 Bounding the second moment of the degrees: proof of Lemma 12.2.6

In this subsection, we compute upper bounds on the second moments of degree functions. This constitutes the main technical component of the proof.

As one can imagine, computing second moments requires counting the number of chains $C \in \mathcal{H}_u^{(r)}$ where the left half C_L (or right half C_R) contains a particular set Z. Because of this, we first prove the following claim.

Claim 12.2.7 (Ideal smoothness of chains from designs). Let H be a design 3-LCC and let $H_1, \ldots, H_n$ be the 3-uniform hypergraphs defined in Remark 3.3.12. Let $r \geq 1$ be an integer, and let $Z \subseteq [n]$ be a subset of size t, for some $0 \leq t \leq r$. Then, the number of chains $C \in \mathcal{H}_u^{(r)}$ with $Z \subseteq C_R$ is at most $\binom{r}{t} t! (3\delta n)^{r-t} \cdot 2^r$. And, for any $w \in [n]$, the number of chains $C \in \mathcal{H}_u^{(r)}$ with tail w and $Z \subseteq C_L$ is at most $\binom{r}{t} t! (3\delta n)^{r-t-1} \cdot 2^r$ if $t \leq r-1$ and $r! \cdot 2^r$ if $|Z| = r$.

Proof. First, let us count the number of chains $C \in \mathcal{H}_u^{(r)}$ with $Z \subseteq C_R$. We compute this in a similar way to our upper bound on $|\mathcal{H}_u^{(r)}|$. First, we pick the $\binom{r}{t}$ locations in C_R (recall that C_R is implicitly ordered by the order that the vertices appear in the chain) that will contain Z, and then we pick one of the $t!$ ways of ordering the entries of Z in these locations. Formally, we view this as fixing an ordered tuple $Q \in \{[n] \cup \star\}^r$, where the set of non-$\star$ elements of Q is equal to Z. The notation $Q_h = \star$ means that the element $v_{2(h-1)+2}$ in the chain C is "free", and $Q_h = v$ means that we must have $v_{2(h-1)+2} = v$.

Next, we count the number of chains as follows. We start with $u_0 = u$, and then we choose an ordered constraint $(v_1, v_2, u_1) \in H_{u_0}$ as follows. If $Q_1 \neq \star$, then we clearly have at most 2 choices, as we have forced $v_2 = v$ for where $v = Q_1$, which leaves at most one (unordered) $C \in H_{u_0}$ that contains v, and then we have 2 ways to order C. If this is not one of the locations where we have placed an entry of Z, i.e., $Q_1 = \star$, then we have at most $6\delta n$ choices. In total, we pay at most $\binom{r}{t} t!(6\delta n)^{r-|Z|} 2^{|Z|} = \binom{r}{t} t!(3\delta n)^{r-|Z|} 2^r$.

Now, we fix $w \in [n]$ and count the number of chains $C \in \mathcal{H}_u^{(r)}$ with tail w and $Z \subseteq C_L$. We first observe that if $|Z| = r$, then we have at most $2^r \cdot r!$ choices. Indeed, this means that $Z = C_L$, so we first pick an ordering on Z (to determine the ordering of the vertices in C_L), and then we pay a factor of 2 per step in the chain (as in the analysis in the previous paragraph). In total, there are $2^r \cdot r!$ choices.

Next, suppose that $|Z| \leq r - 1$. As before, we pay $\binom{r}{t} \cdot t!$ to determine Q, i.e., the locations and ordering of Z within the (ordered) set C_L. Let us now consider a fixed choice of the locations and ordering. We have two cases.

In the first case, suppose that $Q_r = \star$, i.e., the vertex of C_L in the "last link" (namely, $v_{2(r-1)+1}$), is not one of the locations chosen. Then, we can proceed as in the case of C_R, where we pay a factor of 2 to choose a link where v_{2h+1} is determined by Q, and a factor of $6\delta n$ on the other steps. There is one exception, which is the last step of the chain. Now, because we have also fixed the tail w, there are again only 2 choices for this step, even though $Q_r = \star$. Thus, in total, we have paid at most $2^{|Z|+1}(6\delta n)^{r-|Z|-1} = (3\delta n)^{r-|Z|} \cdot 2^r$.

In the second case, suppose that $Q_r \neq \star$, so that the vertex $v_{2(r-1)+1}$ is one of the locations chosen. Let h^* denote the index of the last $\star$ in Q, so $Q_{h^*} = \star$ and $Q_h \neq \star$ for all $h^* < h \leq r$. We now start *at the tail* of the chain and work our way backwards until we reach the h-th link in the chain. In the first step, we have already fixed the tail w and the vertex $v_{2(r-1)+1}$, and so because H is a *design*, there are at most 2 *ordered* tuples $(v, v', v_{2(r-1)+1}, w)$ where $\{v, v', v_{2(r-1)+1}, w\} \in H$, as there is one such unordered tuple and then we can swap the locations of v and v'. We continue backwards along the chain in this way until we reach the location h^*, so that $v_{2(h^*-1)+1}$ is not determined by Q since $Q_{h^*} = \star$. In particular, we have completely determined u_{h^*}, along with the all elements *after* u_{h^*} in the chain, namely $(v_{2h^*+1}, v_{2h^*+2}, \ldots, u_r)$.

Next, we proceed from the start of the chain, again paying 2 for each non-$\star$ entry and $6\delta n$ for each $\star$ entry, until we reach the h^*-th link. We have thus determined the chain up until (and including) u_{h^*-1}, i.e., $(u_0, v_1, v_2, \ldots, u_{h^*-1})$. For the final 2 vertices $(v_{2(h^*-1)+1}, v_{2(h^*-1)+2})$, we have at most 2 choices, because there is at most one hyperedge in $H_{u_{h^*-1}}$ that contains u_{h^*}, and then we have 2 ways to order the vertices. In total, we have paid $(6\delta n)^{r-|Z|-1} \cdot 2^{|Z|+1} = (3\delta n)^{r-|Z|-1} \cdot 2^r$, the same as in the other case.

In total, when $|Z| = t \leq r - 1$, we have at most $\binom{r}{t} t!(3\delta n)^{r-|Z|-1} \cdot 2^r$ choices. □

With Claim 12.2.7 in hand, we are almost ready to compute the second moments. To begin, we will first compute good upper bounds on the first moments $\mathbb{E}_{(S,w)}[\deg_{u,L}(S, w)]$ and $\mathbb{E}_T[\deg_{u,R}(T)]$. For the remainder of the proof, we may omit the subscript u in some places for convenience.

We have

$$\frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n-4r)^r \le d_R = \mathbb{E}_T[\deg_R(T)] \le \frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r\,,$$
$$\frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n-4r)^r \le d_L = \mathbb{E}_{(S,v)}[\deg_L(S,v)] \le \frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r\,.$$

This is because each chain C contributes $\binom{n-2r}{\ell-r}$ edges to the graph G, and we have already computed $(6\delta n-4r)^r \le |\mathcal{H}_u^{(r)}| \le (6\delta n)^r$. We also clearly have $(6\delta n-4r)^r \ge (6\delta n)^r(1-O(r^2/n))$, and so we have:

$$\left(1-\frac{O(r^2)}{n}\right)\frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r \le d_R = \mathbb{E}_T[\deg_R(T)] \le \frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r\,, \tag{12.3}$$
$$\left(1-\frac{O(r^2)}{n}\right)\frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r \le d_L = \mathbb{E}_{(S,v)}[\deg_L(S,v)] \le \frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r\,. \tag{12.4}$$

Computing second moment of the right degree. We now compute the second moments. We will begin with $\mathbb{E}_T[\deg_R(T)^2]$, as this case is simpler. We have

$$\begin{aligned}
&\mathbb{E}_T[\deg_R(T)^2]\\
&\le \sum_{\substack{C=(C_L,C_R,w)\\ C'=(C'_L,C'_R,w')}} \Pr[C_R, C'_R \subseteq T] \quad (T \text{ adjacent to edge labeled by } C \text{ implies } C_R \subseteq T)\\
&= \sum_{C=(C_L,C_R,w)}\ \sum_{t=0}^{r}\ \sum_{\substack{C'=(C'_L,C'_R,w')\\ |C_R\cap C'_R|=t}} \Pr[C_R, C'_R \subseteq T]\\
&= \sum_{C=(C_L,C_R,w)}\ \sum_{t=0}^{r}\ \sum_{\substack{C'=(C'_L,C'_R,w')\\ |C_R\cap C'_R|=t}} \frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}} \quad (\text{as } C_R\cup C'_R \subseteq T \text{ and } |C_R\cup C'_R| = 2r-t)\\
&\le \sum_{C=(C_L,C_R,w)}\ \sum_{t=0}^{r}\binom{r}{t}\cdot\binom{r}{t}t!(3\delta n)^{r-t}\cdot 2^r\cdot\frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}} \quad (\text{by Claim 12.2.7 and } \tbinom{r}{t} \text{ to pick } Z\subseteq C_R \text{ where } C_R\cap C'_R = Z)\\
&\le \sum_{t=0}^{r}(6\delta n)^r\binom{r}{t}\binom{r}{t}t!(3\delta n)^{r-t}\cdot 2^r\cdot\frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}}\\
&\le \left(1+\frac{O(r^2)}{n}\right)d_R^2\sum_{t=0}^{r}\binom{r}{t}\binom{r}{t}t!(3\delta n)^{-t}\frac{\binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \quad (\text{by Eq. (12.3)}\,.
\end{aligned}$$

Now, we apply Fact 3.6.3 to conclude that

$$\begin{aligned}
\mathbb{E}_T[\deg_R(T)^2] &\le \left(1+\frac{O(\ell^2)}{n}\right)d_R^2\sum_{t=0}^{r}\binom{r}{t}(3\delta n)^{-t}n^t\frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}}\\
&= \left(1+\frac{O(\ell^2)}{n}\right)d_R^2\sum_{t=0}^{r}(3\delta)^{-t}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}}\,.
\end{aligned}$$

Now, we observe that $\sum_{t=0}^{r} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$, as this is the probability mass function of a hypergeometric distribution, and that $3\delta = 1 - \frac{1}{n}$ (as H is a *design*), and so $(3\delta)^{-t} \leq (3\delta)^{-r} \leq \left(1 + \frac{O(r)}{n}\right)$. Thus,

$$\mathbb{E}_T[\deg_R(T)^2] \leq \left(1 + \frac{O(\ell^2)}{n}\right) d_R^2 ,$$

which gives the desired bound on the second moment.

Computing second moment of left degree. We now compute $\mathbb{E}_{(S,v)}[\deg_L(S,v)^2]$. We have

$$\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] \leq \sum_{C=(C_L,C_R,w),C'=(C'_L,C'_R,w)} \Pr[C_L, C'_L \subseteq S \wedge v = w] \text{ (both chains have same fixed tail } w)$$

$$= \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r} \sum_{\substack{C'=(C'_L,C'_R,w) \\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w]$$

$$= \left(\sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w) \\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w] \right) + \frac{\binom{n-2r}{\ell-r}}{n \cdot \binom{n}{\ell}} \cdot (6\delta n)^r \cdot r! 2^r ,$$

where the last equality is because when $t = r$, then $C_L = C'_L$, and so $\Pr[C_L \subseteq S \wedge v = w] = \frac{\binom{n-2r}{\ell-r}}{n \cdot \binom{n}{\ell}}$, and by Claim 12.2.7, there are $r!2^r$ choices for C'.

Let us quickly handle this second term. We have by Eq. (12.4),

$$\frac{\binom{n-2r}{\ell-r}}{n \cdot \binom{n}{\ell}} \cdot (6\delta n)^r \cdot r! 2^r \leq \left(1 + \frac{O(r^2)}{n}\right) d_L \cdot r! 2^r .$$

We now compare d_L and $r!2^r$. By Eq. (12.4), we have

$$d_L \geq \left(1 - \frac{O(r^2)}{n}\right) \frac{\ell!}{n^{\ell+1}} \cdot \frac{(n-2r)^{\ell-r}}{(\ell-r)!} \cdot (6\delta n)^r \geq \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) (6\delta)^r \cdot \frac{1}{n} \cdot \frac{\ell!}{(\ell-r)!} .$$

Therefore,

$$\frac{d_L}{2^r r!} \geq \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) (3\delta)^r \cdot \frac{1}{n} \cdot \frac{\ell!}{(\ell-r)!r!} = \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) \left(1 - \frac{1}{n}\right)^r \cdot \frac{1}{n} \cdot \binom{\ell}{\ell-r}$$
$$= \left(1 - \frac{O(r\ell)}{n}\right) \left(1 - \frac{1}{n}\right)^r \cdot \frac{1}{n} \cdot \binom{\ell}{\ell-r} .$$

As $\binom{\ell}{\ell-r} = \eta n$ is the definition of η in Lemma 12.2.6, we conclude that

$$\frac{d_L}{2^r r!} \geq \eta \left(1 - \frac{O(r\ell)}{n}\right) ,$$

and so the second term is $\eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right)$.

We now return to the main calculation. We have

$$\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] \le \left(\sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w) \\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w] \right) + \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right)$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w) \\ |C_L \cap C'_L|=t}} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \text{ (as } C_L \cup C'_L \subseteq S \text{ and } |C_L \cup C'_L| = 2r-t)$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! 2^r (3\delta n)^{r-t-1} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \text{ (by Claim 12.2.7 and } \binom{r}{t} \text{ to pick } Z = C_L \cap C'_L)$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{t=0}^{r-1} (6\delta n)^r \binom{r}{t}\binom{r}{t} t! 2^r (3\delta n)^{r-t-1} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}}$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \frac{(6\delta n)^{2r}}{3\delta n} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! (3\delta n)^{-t} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}}$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(r^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! (3\delta n)^{-t} \frac{\binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \text{ (by Eq. (12.4))}$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(\ell^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} \binom{r}{t} (3\delta n)^{-t} n^t \frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}} \text{ (by Fact 3.6.3)}$$

$$\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(\ell^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} (3\delta)^{-t} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}}.$$

Now, we have $\sum_{t=0}^{r} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$ as this is the probability mass function of a hypergeometric distribution. As $3\delta = 1 - 1/n$, it follows that $(3\delta)^{-t-1} \le (3\delta)^{-r} \le 1 + O(r/n)$, and therefore we conclude that $\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] \le \left(1 + \frac{O(\ell^2)}{n} + \eta\right) d_L^2$.

12.3 Warmup: an $n \ge \tilde{\Omega}(k^4)$ lower bound via 2-chains

In this section, we give a detailed sketch of the proof of the following theorem, which is a weaker version of Theorem 8. Notice that this theorem already improves the prior best known 3-LCC lower bound, established in Chapter 11, by a polynomial factor in k.

Theorem 12.3.1 (Weak version of Theorem 8)**.** *Let $\mathcal{L} : \{-1,1\}^k \to \{-1,1\}^n$ be a linear $(3,\delta)$-LCC in normal form with $\delta = O(1)$. Then, $n \ge \tilde{\Omega}(k^4)$.*

The theorem above obtains a lower bound of $n \gtrsim k^4$ — worse than the bound of $n \gtrsim k^5$ predicted by the heuristic but still beating $n \gtrsim k^3$ from Theorem 7; we discuss the reason that we do not match the heuristic in Remark 12.3.2.

Proof. As before, we have 3-uniform hypergraph matchings $H_1, \dots, H_n$, where for any $u \in [n]$ and $C \in H_u$, we have that for any $b \in \{-1,1\}^k$, $x = \mathcal{L}(b)$ satisfies $x_C = x_u$. Following Section 12.1.2,

we shall let $\mathcal{H}_i^{(2)}$ denote the set of 2-chains with head i. We define the 5-XOR instance $\Psi_b(x)$ as

$$\Psi_b(x) := \sum_{i=1}^{k} b_i \sum_{\vec{C}=(i,C_0,w_0,C_1,w_1)\in\mathcal{H}_i^{(2)}} x_{C_0}x_{C_1}x_{w_1} \ .$$

We note that $\mathrm{val}(\Psi_b) = k(3\delta n)^2$ for any $b \in \{-1,1\}^k$, as the instance is satisfiable and has $k(3\delta n)^2$ constraints in total. Following the strategy in Section 12.1.1, we shall use spectral refutation via Kikuchi matrices to bound $\mathrm{val}(\Psi_b)$ with high probability for a random $b \in \{-1,1\}^k$.

12.3.1 Step 1: the Cauchy–Schwarz trick

As we have observed, the basic Kikuchi matrices in Definition 12.1.2 are only defined for constraints of even arity, but the constraints in $\mathcal{H}_i^{(2)}$ have arity 5, i.e., odd arity. The standard way to handle odd arity XOR instances is to use the "Cauchy–Schwarz trick", which produces even arity instances as follows. Let $\vec{C} \in \mathcal{H}_i^{(2)}$ and $\vec{C}' \in \mathcal{H}_j^{(2)}$ for $i \neq j \in [k]$ be two constraints in our initial 5-XOR instance, where $\vec{C} = (i, C_0, w_0, C_1, w_1)$ and $\vec{C}' = (j, C_0', w_0', C_1', w_1')$ where $w_1 = w_1'$, i.e., the last element of both chains is the same. From this pair, we can "cancel" $w_1 = w_1'$, producing the derived constraint $x_{C_0}x_{C_1}x_{C_0'}x_{C_1'} = b_ib_j$, which has arity 8. We do this for all pairs of chains with the same "tail" vertex w. We note that this process produces at least $(k(3\delta n)^2)^2/n \sim k^2n^3$ constraints.

We now define the following "Cauchy–Schwarzed instance" polynomial:

$$f_b(x) = \sum_{i\neq j\in[k]} b_ib_j \sum_{w\in[n]} \sum_{\vec{C}\in\mathcal{H}_i^{(2)},\vec{C}'\in\mathcal{H}_j^{(2)}:w_1=w_1'=w} x_{C_0}x_{C_1}x_{C_0'}x_{C_1'} \ .$$

The phrase "Cauchy–Schwarz trick" refers to the fact that one can show $k^2n^4 \sim \Psi_b(x)^2 \leq n \cdot f_b(x) + o(k^2n^4)$ via a simple application of the Cauchy–Schwarz inequality and a bound on the "diagonal terms" where $i = j$. This reduces the task to bounding the cross-term polynomial f_b.

We now observe that the "right-hand sides" of the constraints in f_b are no longer independent, as they are of the form b_ib_j for $i \neq j \in [k]$, and this will cause an issue "downstream" when we apply matrix concentration bounds, as the matrices will not be independent. To recover independence, we consider the polynomial $f_{M,b}(x)$ defined for a (directed) matching M on $[k]$:

$$f_{M,b}(x) = \sum_{(i,j)\in M} b_ib_j \sum_{w\in[n]} \sum_{\vec{C}\in\mathcal{H}_i^{(2)},\vec{C}'\in\mathcal{H}_j^{(2)}:w_1=w_1'=w} x_{C_0}x_{C_1}x_{C_0'}x_{C_1'} \ .$$

Because we now sum over a matching, we have that b_ib_j and $b_{i'}b_{j'}$ are independent for different directed edges (i,j) and (i',j') in M. And, we can easily relate f_b and $f_{M,b}$, as $f_b(x) = 2(k-1)\mathbb{E}_M f_{M,b}(x)$ when k is even, and $f_b(x) = 2k\mathbb{E}_M f_{M,b}(x)$ when k is odd, where the expectation is over a maximum matching M. This is because the chance that M contains a directed edge (i,j) is $\frac{1}{2(k-1)}$ if k is even and $\frac{1}{2k}$ if k is odd. In particular, there exists a maximum matching M such that $\mathrm{val}(f_{M,b}) \geq \frac{2}{k}\,\mathrm{val}(f_b) \sim kn^3$.

Remark 12.3.2. Restricting to a matching M loses a factor of k in the number of constraints. This leads to a factor k "loss" in the density of the corresponding Kikuchi matrix and is the main reason

why we obtain weaker bound of $n \geq \tilde{O}(k^4)$ instead of k^5 suggested by our heuristic calculation in Section 12.1.2. A better bound could be obtained by instead following the setup in Chapter 11, where we split $[k]$ randomly into a left and right set L and R and only consider constraints where $i \in L$ and $j \in R$ (thereby losing only $\sim 1/2$ of the constraints instead of a factor k). This careful setup is necessary in Chapter 11 to achieve our goal of obtaining a cubic (as opposed to the known quadratic) bound, but this makes the "row pruning" step (i.e., arguing approximate regularity of Kikuchi graphs after removing a negligible fraction of constraints) significantly more challenging. In our case, the effect of this loss on the final lower bound diminishes as the length of the chain r grows and when $r \sim \log n$, disappears asymptotically, and so we pick a matching M to make the row pruning easier.

12.3.2 Step 2: spectral refutation via Kikuchi matrices

Let us now bound $\mathrm{val}(f_{M,b})$ (with high probability over $b \in \{-1,1\}^k$) for any maximum matching M. We introduce our Kikuchi matrices:

Definition 12.3.3. For $i \neq j \in [k]$ and $\vec{C} = (i, C_0, w_0, C_1, w_1)$ and $\vec{C}' = (j, C_0', w_0', C_1', w_1')$ with $w_1 = w_1'$, we define the matrix $A_{i,j}^{(\vec{C},\vec{C}')}$ as follows. The rows/columns of the matrix $A_{i,j}^{(\vec{C},\vec{C}')}$ are indexed by a 4-tuple of sets (S_0, S_1, S_0', S_1'), each in $\binom{[n]}{\ell}$, and the $((S_0, S_1, S_0', S_1'), (T_0, T_1, T_0', T_1'))$-th entry is 1 if $S_0 \oplus T_0 = C_0$, $S_1 \oplus T_1 = C_1$, $S_0' \oplus T_0' = C_0'$, $S_1' \oplus T_1' = C_1'$, and is 0 otherwise.

We let $A_{i,j} = \sum_{\vec{C} \in \mathcal{H}_i^{(2)}, \vec{C}' \in \mathcal{H}_j^{(2)} : w_1 = w_1'} A_{i,j}^{(\vec{C},\vec{C}')}$ and $A = \sum_{(i,j) \in M} b_i b_j A_{i,j}$.

We now observe that each matrix $A_{i,j}^{(\vec{C},\vec{C}')}$ has exactly D^4 nonzero entries, where $D = 2 \cdot \binom{n-2}{\ell-1}$, and the matrix has N^4 rows/columns, where $N = \binom{n}{\ell}$. We note that $D/N \sim \ell/n$, and so the average number of nonzero entries per row (or column), i.e., the density, is $(D/N)^4 \sim (\ell/n)^4 = (\ell/n)^{q/2}$, as the arity of the constraints is 8.

We also observe that for any $x \in \{-1,1\}^n$, $D^4 f_{M,b}(x) = x'^{\top} A x'$, where x' is the vector with (S_0, S_1, S_0', S_1')-th entry equal to $\prod_{v \in S_0} x_v \prod_{v \in S_1} x_v \prod_{v \in S_0'} x_v \prod_{v \in S_1'} x_v$. We thus have that

$$kn^3 \cdot D^4 \leq D^4 \cdot \mathrm{val}(f_{M,b}) \leq \|A\|_{\infty \to 1} \leq N^4 \|A\|_2 \ .$$

For any $i \neq j$, the matrix $A_{i,j}$ has density $\sim m_{i,j}(D/N)^4 \sim (\ell/n)^4$, where $m_{i,j}$ is the number of the constraints in f_b with right-hand side $b_i b_j$. Let us now argue that each $m_{i,j}$ is at most $O(n^3)$. Indeed, $m_{i,j}$ is the number of pairs of 2-chains $(i, C_0, w_0, C_1, w_1) \in \mathcal{H}_i^{(2)}$ and $(j, C_0', w_0', C_1', w_1') \in \mathcal{H}_j^{(2)}$ where $w_1 = w_1'$. To show that $m_{i,j} \leq O(n^3)$, we pick w_0, w_1 and w_0', for a total of n^3 choices, and observe that this completely determines both chains. Indeed, because H_i is a matching, there is at most one constraint C in H_i that contains w_0, and then C_0 must be $C \setminus \{w\}$. This similarly shows that we have at most one choice of C_1 and also C_0'. Finally, because $w_1' = w_1$, and we know w_1, we thus know w_1' as well, which by similar reasoning gives us at most one choice for C_1', and we have determined the entire chain. We note that we have a lower bound of $\sim kn^3$ on the total number of constraints $\sum_{(i,j) \in M} m_{i,j}$, so this calculation also shows that no $m_{i,j}$ can be much larger than the average.

Returning to the density calculation, we have shown that $A_{i,j}$ has density at most $n^3(\ell/n)^4 = \ell^4/n$. Again, following the blueprint in Section 12.1.1, we will set $\ell = n^{1/4} \cdot \mathrm{polylog}(n)$, and we

want to show that the matrices $A_{i,j}$ satisfy the approximate regularity condition, i.e., the number of rows/columns with more than $\Delta = \ell^4 \cdot \text{polylog}(n)/n$ nonzero entries is at most $N^4/\text{poly}(n)$. Let us finish the proof, assuming that this holds.

Proof assuming approximate regularity. Let $\mathcal{B}$ denote the set of rows/columns that are "bad" for some pair (i,j), i.e., the matrix $A_{i,j}$ has more than Δ nonzero entries in that row. Let $B_{i,j}$ be the matrix where the rows and columns in $\mathcal{B}$ have been all set to 0. Let $B = \sum_{(i,j)\in M} b_i b_j B_{i,j}$. We have that B is the sum of mean 0 independent matrices, each with spectral norm $\|B_{i,j}\|_2 \leq \Delta$. Therefore, by matrix Khintchine (Fact 3.4.2), we have that with high probability over b, $\|B\|_2 \leq O(\Delta\sqrt{k\log(N^4)}) = O(\Delta\sqrt{k\ell\log n})$.

Now, we observe that $\|A - B\|_{\infty\to 1} \leq o(N)$. This is because the number of nonzero entries that we have removed from A to produce B is at most $k \cdot n^3 \cdot N^4/\text{poly}(n) = o(N^4)$ (there are k edges (i,j) in the matching M, each has $m_{i,j} \leq n^3$ constraints, and each row of $A_{i,j}$ has at most $m_{i,j} \leq n^3$ nonzero entries) provided that the $\text{poly}(n)$ factor is large enough. We thus conclude that

$$kn^3 \cdot D^4 \leq D^4 \cdot \text{val}(f_{M,b}) \leq \|A - B\|_{\infty\to 1} + N^4\|B\|_2 \leq o(N^4) + N^4 O(\Delta\sqrt{k\ell\log n}) \ .$$

Substituting the value for Δ and rearranging, we conclude that $k \leq \ell \cdot \text{polylog}(n) \leq \tilde{O}(n^{1/4})$.

We remark that Sections 12.3.1 and 12.3.2 are fairly mechanical, and they justify the use of the heuristic calculation. The place where we had "freedom" is in the choice of constraints to use in the initial XOR instance, which we chose to be the 2-chains $\mathcal{H}_i^{(2)}$. It thus remains to bound the number of bad rows $\mathcal{B}$. This "row pruning" step is key to converting the heuristic into a full proof.

12.3.3 Step 3: row pruning, the key technical step

We want to understand if, after dropping a $1/\text{poly}(n)$ fraction of the rows, every Kikuchi graph $A_{i,j}$ satisfies approximate regularity. This is equivalent to showing that for every matrix $A_{i,j}$, with probability at least $1 - 1/\text{poly}(n)$ a uniformly random row (S_0, S_1, S_0', S_1'), has at most Δ nonzero entries in $A_{i,j}$ for $\Delta = \ell^4 \cdot \text{polylog}(n)/n = \Delta_{avg}\,\text{polylog}(n)$.

The heavy pair degree. We now make a key observation. Whether the above approximate regularity property holds for a given collection of matchings $H_1, H_2, \ldots, H_n$ is governed by a single parameter that we call the *heavy pair degree* d. This is the maximum, over all pairs $\{v, v'\} \subseteq [n]$, of the number of hyperedges across the H_i's that contain $\{v, v'\}$. We will prove that if d is small enough then approximate regularity holds for every $A_{i,j}$ after dropping a $1/\text{poly}(n)$-fraction of rows. When d is large, this property will not hold for the $A_{i,j}$'s from Definition 12.3.3. Instead, we will define a *different* collection of Kikuchi matrices that have high density and for which row pruning succeeds.

Lemma 12.3.4 (Row pruning for 2-chains with no heavy pairs)**.** *Let $H_1, \ldots, H_n$ be 3-uniform hypergraph matchings of size δn, and let d be the maximum, over all pairs $\{v, v'\}$ of vertices, of the number of pairs (u, C) with $u \in [n]$ and $C \in H_u$ where $\{v, v'\} \subseteq C$. Fix $i \neq j \in [k]$, and let $A_{i,j}$ be the matrix defined in Definition 12.3.3 at level $\ell \in \mathbb{N}$.*

Suppose that $d \leq \ell^2$. Then, the number of rows (S_0, S_1, S_0', S_1') of $A_{i,j}$ with more than $\Delta = \ell^4 \cdot \text{polylog}(n)/n$ nonzero entries is at most $N^4/\text{poly}(n)$.

We note that if the matchings $H_1, \dots, H_n$ are *random*, then we have $d \le \text{polylog}(n)$ with high probability, and so random matchings satisfy the "small heavy-pair degree" assumption with high probability. We can thus think of $d \le \text{polylog}(n)$ as a pseudorandom property of a collection $H_1, \dots, H_n$ of matchings. We now sketch a proof of Lemma 12.3.4.

The degree polynomial and its partial derivatives. As the first step in the proof of Lemma 12.3.4, we define a degree 4 polynomial $\mathsf{Deg}_{i,j} \colon \{0,1\}^{4n} \to \mathbb{N}$, where we think of the $4n$ variables as split into 4 groups of n variables $s^{(0)}, s^{(1)}, s'^{(0)}, s'^{(1)}$, which are indicator variables of the 4 sets S_0, S_1, S'_0, S'_1, respectively. This polynomial $\mathsf{Deg}_{i,j}(s^{(0)}, s^{(1)}, s'^{(0)}, s'^{(1)})$ upper bounds the number of nonzero entries in the (S_0, S_1, S'_0, S'_1)-th row in the matrix $A_{i,j}$ in Definition 12.3.3.

Formally, let $\mathcal{T}_{i,j}$ denote the (multi)-set of 4-tuples (u_0, u_1, v_0, v_1) such that there exists $\vec{C} = (i, C_0, w_0, C_1, w_1) \in \mathcal{H}_i^{(2)}$ and $\vec{C}' = (j, C'_0, w'_0, C'_1, w'_1) \in \mathcal{H}_j^{(2)}$ with $w_1 = w'_1$ such that $u_0 \in C_0, u_1 \in C_1, v_0 \in C'_0, v_1 \in C'_1$; if there are multiple such pairs $(\vec{C}, \vec{C}')$ that produce the same (u_0, u_1, v_0, v_1), then we add this tuple multiple times. Then, we set

$$\mathsf{Deg}_{i,j}(s^{(0)}, s^{(1)}, s'^{(0)}, s'^{(1)}) := \sum_{(u_0,u_1,v_0,v_1)\in\mathcal{T}_{i,j}} s^{(0)}_{u_0} s^{(1)}_{u_1} s'^{(0)}_{v_0} s'^{(1)}_{v_1} \quad .$$

Note that $\mathsf{Deg}_{i,j}$ is a polynomial with non-negative coefficients. We are interested in the probability that $\mathsf{Deg}_{i,j}$, on uniform draws of 4-tuples of ℓ-size sets, takes a value that deviates from its expectation μ by some multiplicative factor. It is not too difficult to show that we can pass on to independent p-biased product distribution on $\{0,1\}^{4n}$ for $p \sim \ell/n$ without much loss. This is helpful because the tail behavior of low-degree polynomials with non-negative coefficients on product distributions is determined by a bound on its expected partial derivatives. Namely, variants of the Kim-Vu inequality (see Fact 3.4.3) show the following: *if the expectation of every partial derivative of* $\mathsf{Deg}_{i,j}$ *is at most* μ*, then* $\mathsf{Deg}_{i,j}(S_0, S_1, S'_0, S'_1) \le O(\mu \log n)$ *with probability at least* $1 - 1/\text{poly}(n)$.

Let us now examine the expected partial derivatives of $\mathsf{Deg}_{i,j}(s)$. We start by introducing notation to refer to them. Let $Z = (z_0, z_1, z'_0, z'_1) \in ([n] \cup \{\star\})^4$ be an ordered tuple of length 4, with entries either in n or set to $\star$, which we think of as an "unfixed" value. Then, Z encodes partial derivatives with respect to any subset of variables that use at most one variable in each of the groups $s^{(0)}$, $s^{(1)}$, $s'^{(0)}$, $s'^{(1)}$. All other partial derivatives of $\mathsf{Deg}_{i,j}$ are 0 since $\mathsf{Deg}_{i,j}$ has degree 1 in each of the 4 groups of variables (i.e., $\mathsf{Deg}_{i,j}$ is 4-partite). We know that $\mathbb{E}[\mathsf{Deg}_{i,j}(s)] = \mu_{(\star,\star,\star,\star)} \le 2^4(\ell/n)^4 \cdot n^3 = O(1) \cdot \ell^4/n$; the factor of 2^4 comes from the fact that each pair $(\vec{C}, \vec{C}')$ adds 2^4 different tuples to $\mathcal{T}_{i,j}$. Now, Fact 3.4.3 implies that the chance that $\mathsf{Deg}_{i,j}$ takes a value larger than $\mu \cdot \text{polylog}(n)$ is at most $1/\text{poly}(n)$ if $\mu_Z \le \mu$ for all Z.

Computing expected partial derivatives. To help bound the expected partial derivatives μ_Z, let us relate these parameters to combinatorial quantities of the hypergraphs $H_1, H_2, \dots, H_n$. Notice that when we take partial derivatives with respect to some Z, the only monomials that "survive" are ones that "contain" Z, and furthermore the expectation of the partial derivative is simply $(\ell/n)^{\#\text{ of }\star\text{ entries in }Z}$ times the number of such monomials. Formally, let $\deg_{i,j}(Z)$ be the number of pairs $(\vec{C}, \vec{C}') \in \mathcal{H}_i^{(2)} \times \mathcal{H}_j^{(2)}$ where $w_1 = w'_1$ and $z_0 \in C_0, z_1 \in C_1, z'_0 \in C'_0, z'_1 \in C'_1$, where for the symbol $\star$, we say that $\star \in C$ always holds — we say that such a pair $(\vec{C}, \vec{C}')$ *contains* Z. Then, the expected partial derivative at Z is $\mu_Z = 2^{4-|Z|}(\ell/n)^{4-|Z|} \deg_{i,j}(Z)$, where $|Z|$ is the number

of non-$\star$ entries in Z.[8] For example, $Z = (\star, \star, \star, \star)$ is contained in all such pairs of 2-chains, and so $\deg_{i,j}(\star, \star, \star, \star) = m_{i,j} \le O(n^3)$ and $\mu_Z = \mu = 16(\ell/n)^4 m_{i,j}$. Let us use the shorthand $\mu_t = \max_{Z:|Z|=t} \mu_Z$.

Let Z be an arbitrary 4-tuple with at least one non-$\star$ entry. As explained above, estimating μ_Z is, up to scaling, equivalent to counting $\deg_{i,j}(Z)$, the number of pairs $(\vec{C}, \vec{C}')$ that contain Z. We next observe that if Z has no $\star$ entries, then the number of 2-chains $(\vec{C}, \vec{C}')$ containing Z is an absolute constant. This is because there is at most one constraint $C_0 \cup \{w_0\}$ that contains z_0 in H_i. Given this constraint, there are 2 choices for w_0, as $w_0 \in C_0 \cup \{w_0\} \setminus \{z_0\}$. Given w_0, there is at most one constraint $C_1 \cup \{w_1\}$ in H_1 that contains z_1, and then at most 2 choices for w_1. We can similarly use the knowledge of (z_0', z_1') to bound the number of choices for C_0', C_1'. All in all, we have at most $16 = O(1)$ choices for the pair $(\vec{C}, \vec{C}')$ given Z with no $\star$ entries. This immediately shows that for Z such that $|Z| = 4$, $\mu_Z \le O(1) \le \mu$.

Let us now deal with Z's with at least one $\star$ entry by breaking up into cases depending on $|Z|$. We will view the counting of $\deg_{i,j}(Z)$ as a procedure that makes a bounded number of choices to decode the pair $(\vec{C}, \vec{C}')$.

Let us deal with the case when $|Z| = 1$. By swapping the roles of i and j if needed, without loss of generality we can assume that one of z_0 or z_1 is non-$\star$, and all other entries in Z are $\star$. There are at most n choices for z_0 (if $z_1 \neq \star$) or z_1 (if $z_0 \neq \star$). We now have n choices for z_0', which again determines C_0' and w_0' up to 2 choices. We now observe that (C_1', w_1') is uniquely determined. Indeed, this is because we know w_1', as it equals w_1 (the two 2-chains must have matching tails), and therefore this determines the hyperedge $C_1' \cup \{w_1'\} \in H_{w_0'}$ uniquely. We have thus shown that for Z with $|Z| = 1$, we have $\deg_{i,j}(Z) \le O(n^2)$, and so $\mu_Z \le (\ell/n)^3 \cdot O(n^2) \le O(\ell^3/n) \le O(\ell^4/n)$.

Let us now handle the case when $|Z| = 2$. Similar arguments as above show that $\mathsf{Deg}_{i,j}(Z) \le O(n)$ holds for all Z except when the non-$\star$ entries of Z look like $Z = (\star, z_1, \star, z_1')$ where $z_1, z_1' \neq \star$, and thus $\mu_Z \le (\ell/n)^2 \cdot O(n) \le O(\ell^4/n)$ for these Z's. To count $\deg_{i,j}(Z)$ for $Z = (\star, z_1, \star, z_1')$ where $z_1, z_1' \neq \star$, we pay a factor of n to determine z_0, and then this determines (up to an $O(1)$ factor) C_0 and C_1 as well. Now, we know w_1' (because it is equal to w_1) and z_1' which is in C_1'. Thus, the hyperedge $C_1' \cup \{w_1'\}$ must contain the pair $\{z_1', w_1'\}$. Using the heavy pair degree, there are at most d choices for the pair $(w_0', C_1' \cup \{w_1'\})$, and after learning w_0' we also know C_0'. Hence, we have paid a total of $O(nd)$ choices, which implies that $\mu_2 \le (\ell/n)^2 \cdot O(nd) = O(\ell^2 d/n)$. For $|Z| = 3$, a similar issue arises and gives a bound of $\mu_3 \le O(\ell d/n)$.

We can now finish the proof of Lemma 12.3.4.

Proof of Lemma 12.3.4. Notice that if $d \le \ell^2$ then $\mu_t \le \mu$ for every t. Applying Fact 3.4.3 now yields that the probability that $\mathsf{Deg}_{i,j} > \mu \cdot \mathrm{polylog}(n)$ is at most $1/\mathrm{poly}(n)$. Taking a union bound on $k < n$ yields that the fraction of bad rows $|\mathcal{B}|/N$ is at most $1/\mathrm{poly}(n)$, as desired. □

12.3.4 Step 4: hypergraph decomposition to handle large heavy pair degree

We will handle the case when the heavy pair degree is high by designing a *different* Kikuchi matrix. To do this, we will construct the cross term polynomial (obtained by applying the Cauchy–Schwarz inequality) slightly differently. Our current Kikuchi matrix is built from the

[8] The extra factor of $2^{4-|Z|}$ comes from the fact that for every Z and pair $(\vec{C}, \vec{C}')$ containing Z, the pair $(\vec{C}, \vec{C}')$ produces $2^{4-|Z|}$ tuples (u_0, u_1, v_0, v_1) in $\mathcal{T}_{i,j}$ that contain Z. In this case, this is just a constant factor, so we can ignore it.

XOR instance obtained by pairing up chains that agree on their tails and thus "cancel" (i.e., square out) one variable. When the heavy pair degree is large, we will build chains by cancelling a pair of variables instead. The number of pairs of chains that agree in a pair of variables instead of just their tails, i.e., the new number of "Cauchy–Schwarzed" constraints, will of course be smaller than before. On the other hand, since we cancel a pair of variables instead of just the tail, the arity of the resulting XOR instance will be smaller: 6 instead of 8. The punchline is that the density vs. arity trade-off (i.e., our key heuristic discussed in Section 12.1.2) breaks in our favor, *provided that there are many "heavy pairs"*.

To formally implement this argument, we *decompose* the set of chains by "labeling" each chain by the heavy pair contained within, if one exists. Intuitively, this is the pair of variables in the chain that we intend to cancel in the Cauchy–Schwarz trick. If the chain does not contain any heavy pair, then we label it by its tail variable w, which we will cancel in the Cauchy–Schwarz trick as done before in Section 12.3.1. We let $\mathcal{H}^{(Q)}$ denote the set of chains labeled by the heavy pair Q, and $\mathcal{H}^{(1,w)}$ denote the set of chains labeled by the tail variable w.

Formally, our hypergraph decomposition is as follows. Given the collection $\mathcal{H}^{(1)} = \{(u, C, w) : u \in [n], C \cup \{w\} \in H_u\}$ of 1-chains, we perform the following greedy algorithm: if there exists an ordered pair $Q = (Q_1, Q_2)$ such that there are more than $d := \ell^2$ 1-chains (u, C, w) in $\mathcal{H}^{(1)}$ with $Q_1 \in C$ and $Q_2 = w$, i.e., Q is a heavy pair contained in the chain (u, C, w), then we choose an arbitrary set of *exactly* d such 1-chains, remove them from $\mathcal{H}^{(1)}$, and place them in a new "partition" $\mathcal{H}^{(1,Q)}$.[9] Finally, if there is no such heavy pair Q, then we create partitions $\mathcal{H}^{(1,w)}$ for each $w \in [n]$, and add all remaining 1-chains with "tail w", i.e., 1-chains of the form (u, C, w), to $\mathcal{H}^{(1,w)}$.

This decomposition has the following properties:

(1) $\mathcal{H}^{(1)} = (\cup_w \mathcal{H}^{(1,w)}) \bigcup (\cup_Q \mathcal{H}^{(1,Q)})$ is a disjoint partition of $\mathcal{H}^{(1)}$;

(2) For each $Q = (Q_1, Q_2)$, $\mathcal{H}^{(1,Q)}$ is a set of 1-chains that "contain" the tuple Q, i.e., each (u, C, w) in $\mathcal{H}^{(1,Q)}$ has $w = Q_2$ and $C \ni Q_1$;

(3) For each Q, $|\mathcal{H}^{(1,Q)}| = d$;

(4) For each $w \in [n]$, there is only one partition $\mathcal{H}^{(1,w)}$;

(5) The total number of partitions $\mathcal{H}^{(1,Q)}$ is at most $O(n^2/d)$, as there are at most $O(n^2)$ 1-chains, and each $\mathcal{H}^{(1,Q)}$ has exactly d 1-chains.

We stress that the decomposition is only on 1-chains, *not* the set of 2-chains $\cup_{i\in[k]} \mathcal{H}_i^{(2)}$ that are the constraints in the XOR instance! At a high level, this is because, e.g., the 2-chains in $\mathcal{H}_i^{(2)}$ (or $\mathcal{H}_j^{(2)}$) are formed by taking a 1-chain and *prepending* it with a hyperedge in H_i (or H_j), and so "first link" in each 2-chain is specific to the choice of $i \in [k]$, but the "second link" is an arbitrary 1-chain, and so it is "shared" across the $\mathcal{H}_i^{(2)}$'s in some informal sense.[10] This property turns out to be important when it comes time to bound the expected partial derivatives.

[9] There may be more than d such chains, in which case we may produce multiple *different* partitions that have the same pair Q. Thus, the Q's form a multiset, and we will use Q to refer to a particular partition $\mathcal{H}^{(1,Q)}$. In Section 12.5, we handle this issue by reweighting the chains instead.

[10] For this reason, in Section 12.5, the length of the chains defining the XOR constraints is $r + 1$, but we only decompose length r chains.

Now, we define $\mathcal{H}_i^{(2,Q)}$ to be the set of 2-chains (i, C_0, w_0, C_1, w_1) where the "second link" (w_0, C_1, w_1) is in $\mathcal{H}^{(1,Q)}$. Using the decomposition, we now define the following polynomials:

$$\Psi_b(x) \coloneqq \sum_{i=1}^{k} b_i \sum_{\vec{C}=(i,C_0,w_0,C_1,w_1)\in\mathcal{H}_i^{(2)}} x_{C_0} x_{C_1} x_{w_1} \ ,$$

$$\Psi_{i,w}(x) \coloneqq \sum_{C_0,w_0 : C_0\cup\{w_0\}\in H_i} \ \sum_{(w_0,C_1,w_1)\in\mathcal{H}^{(1,w)}} x_{C_0} x_{C_1} \ ,$$

$$\Psi_{i,Q}(x) \coloneqq \sum_{(i,C_0,w_0,C_1,w_1)\in\mathcal{H}_i^{(2,Q)}} x_{C_0} x_{C_1\setminus Q_1} \ ,$$

$$\Psi_b^{(0)}(x,y) \coloneqq \sum_{i=1}^{k} \sum_{w\in[n]} b_i y_w \Psi_{i,w}(x) \ ,$$

$$\Psi_b^{(1)}(x,y) \coloneqq \sum_{i=1}^{k} \sum_{Q} b_i y_Q \Psi_{i,Q}(x) \ ,$$

where above y_Q and y_w are new variables. By definition, if we set $y_w = x_w$ and $y_Q = x_{Q_1} x_{Q_2}$, then we have that $\Psi_b(x) = \Psi^{(0)}(x,y) + \Psi^{(1)}(x,y)$. Indeed, all we have done is partitioned the constraints into these two polynomials and removed the "$x_{Q_1}x_{Q_2}$ term" from each monomial, replacing it with the new variable y_Q.

We now refute the two polynomials $\Psi^{(0)}(x,y)$ and $\Psi^{(1)}(x,y)$ separately using the machinery in Sections 12.3.1 to 12.3.3. In fact, Sections 12.3.1 to 12.3.3 immediately show that we can successfully refute the polynomial $\Psi^{(0)}(x,y)$. Indeed, the only issue that we encountered was in Section 12.3.3, where the row pruning failed if there was a pair $\{v, v'\}$ that appeared in more than ℓ^2 1-chains in $\mathcal{H}^{(1)}$. However, this cannot happen, as otherwise our decomposition algorithm would not have terminated.

It thus remains to handle the second polynomial, $\Psi^{(1)}(x,y)$. Applying the "Cauchy–Schwarz trick" of Section 12.3.1, we can reduce this to the case of bounding the polynomial:

$$f_{M,b}(x) = \sum_{(i,j)\in M} b_i b_j \sum_{Q} \Psi_{i,Q}(x)\Psi_{j,Q}(x) \ ,$$

where M is a maximum matching, as before. Notice that the constraints in $f_{M,b}$ have arity 6 (see Fig. 12.2). Following the blueprint of Section 12.3.2, we define the following Kikuchi matrices.

Definition 12.3.5. For $i \neq j \in [k]$, Q, and $\vec{C} = (i, C_0, w_0, C_1, w_1) \in \mathcal{H}_i^{(2,Q)}$, $\vec{C}' = (j, C_0', w_0', C_1', w_1') \in \mathcal{H}_j^{(2,Q)}$, we define the matrix $A_{i,j}^{(\vec{C},\vec{C}',Q)}$ as follows. The matrix $A_{i,j}^{(\vec{C},\vec{C}',Q)}$ is indexed by a 3-tuple of sets (S_0, R, S_0'), each in $\binom{[n]}{\ell}$, and the (S_0, R, S_0'), (T_0, W, T_0')-th entry is 1 if $S_0 \oplus T_0 = C_0$, $S_0' \oplus T_0' = C_0'$, and $R = \{u\} \cup U$, $W = \{v\} \cup V$, where $C_1 = \{u, Q_1\}$, $C_1' = \{v, Q_1\}$, and $U \subseteq [n]$ is a set of size $\ell - 1$ where $u, v \notin U$.

We let $A_{i,j} = \sum_Q \sum_{\vec{C}\in\mathcal{H}_i^{(2,Q)}, \vec{C}'\in\mathcal{H}_j^{(2,Q)}} A_{i,j}^{(\vec{C},\vec{C}',Q)}$ and $A = \sum_{(i,j)\in M} b_i b_j A_{i,j}$.

Notice that for $\vec{C} = (i, C_0, w_0, C_1, w_1) \in \mathcal{H}_{i,Q,p}^{(2)}$ and $\vec{C}' = (j, C_0', w_0', C_1', w_1') \in \mathcal{H}_{j,Q,p}^{(2)}$, the split of the elements in the constraint across the row (S_0, R, S_0') and the column (T_0, W, T_0') is asymmetric: see Fig. 12.2.

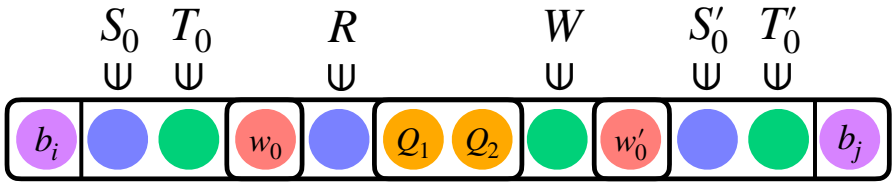

Figure 12.2: A pair of 2-chains $\vec{C} = (i, C_0, w_0, C_1, w_1) \in \mathcal{H}_i^{(2,Q)}$, $\vec{C}' = (j, C'_0, w'_0, C'_1, w'_1) \in \mathcal{H}_j^{(2,Q)}$. The blue vertices appear in the sets (S_0, R, S'_0) for the rows of the matrix $A_{i,j}^{(\vec{C},\vec{C}',Q)}$, and the green vertices appear in the columns. The orange vertices are the elements of Q that are canceled via the Cauchy–Schwarz operation. The purple vertices are the independent random bits that we "disconnect" from the chain and use for the right-hand sides.

Applying the same machinery in Section 12.3.2 to the matrices in Definition 12.3.5 will yield the correct lower bound provided that the row pruning step succeeds. It thus remains to bound the number of rows in $A_{i,j}$ for a fixed pair (i, j) with a number of nonzero entries exceeding the average by a polylog(n) factor.

We now apply Fact 3.4.3. As before, we define a similar degree polynomial $\mathsf{Deg}_{i,j}$, and the tail bound boils down to computing the expected partial derivatives μ_Z, where $Z = (z_0, r, z'_0) \in ([n] \cup \{\star\})^3$ is now a tuple of length 3, and $\mu_Z = (\ell/n)^{3-|Z|} \deg_{i,j}(Z)$, as the constraints have arity 3. We observe that $\deg_{i,j}(\star, \star, \star) \le O(n^2 d)$, as we have $O(n^2)$ choices for $\vec{C} = (i, C_0, w_0, C_1, w_1) \in \mathcal{H}_i^{(2)}$ (which then determines Q), followed by $O(d)$ choices for (w'_0, C'_1, w'_1) (because this must be in $\mathcal{H}^{(1,Q)}$, which has size d), and then a unique choice for C_0. Therefore, $\mu_0 \le (\ell/n)^3 \cdot O(n^2 d) = O(\ell^3 d/n)$.

Bounding μ_1 is straightforward, and we omit the calculations. We obtain a bound of $\mu_1 \le (\ell/n)^2 \cdot O(nd) = O(\ell^2 d/n)$. Bounding μ_2 can be done with a trivial bound of $\deg_{i,j}(Z) \le O(n)$, yielding $\mu_2 \le (\ell/n) \cdot O(n) = O(\ell)$. Finally, it is simple to bound $\deg_{i,j}(Z) \le O(1)$ when $|Z| = 3$, and so we obtain $\mu_3 \le O(1)$.

We notice that $\mu_0 \ge \mu_1$ and $\mu_2 \ge \mu_3$ always hold. So, either μ_0 or μ_2 must be the maximum. Because $d = \ell^2$, we have $\mu_0 = O(\ell^3 d/n) \sim \ell^5/n \gg \ell \sim \mu_2$ because $\ell^4 \gg n$, by choice of ℓ. Thus, $\mu_0 \gg \mu_2$, and so the row pruning argument, etc., will all succeed. This, combined with the refutation argument for $\Psi_b^{(0)}(x)$, implies that our heuristic calculation succeeds and we get a bound of $k \le \tilde{O}(\ell)$, where ℓ is chosen to be $\tilde{O}(n^{1/4})$. Thus, we obtain a lower bound of $k \le \tilde{O}(n^{1/4})$. □

12.3.5 Preview: extending the warmup to a proof of Theorem 8

We now give a brief overview of how we shall extend the ideas used in this warmup to prove Theorem 8. First, we observe that in the argument we presented in Sections 12.3.1 to 12.3.4, there were only two crucial moments in the proof where we had a lot of freedom: (1) the choice of the constraints in the initial XOR instance (in this warmup, we chose the set of 2-chains with head $i \in [k]$), and (2) the choice of the hypergraph decomposition in Section 12.3.4 — the rest of the proof was fairly mechanical, and boiled down to computing the expected partial derivatives

μ_Z. Namely, if we can choose the constraints and the decomposition so that the row pruning succeeds for all the resulting Kikuchi matrices, i.e., the expected partial derivatives of the degree polynomials are appropriately bounded, then the general machinery in Sections 12.3.1 to 12.3.3 succeeds in proving the lower bound predicted by the heuristic calculation in Section 12.1.2 (up to a small loss, see Remark 12.3.2).

As discussed in Section 12.1.2, we shall define the XOR instance using $(r+1)$-chains for a parameter $r = O(\log n)$, and the heuristic calculation predicts that this will yield an exponential lower bound. Thus, the key technical component of the proof is to choose the decomposition of the $(r+1)$-chains so that the degree polynomials of the resulting Kikuchi matrices all satisfy the bounded expected partial derivatives condition. In Section 12.3.4, we showed how to do this for the case when $r = 1$.

We now wish to point out the following crucial observation: the decomposition in Section 12.3.4 is "informed" by the row pruning calculation for the *undecomposed chains* done in Section 12.3.3. Specifically, in Section 12.3.3, we argued that if there is a violating partial derivative for the undecomposed chains, then there is some combinatorial structure in the chains (namely, a heavy pair) that is the "cause" of the large expected partial derivative, and this combinatorial structure is exactly the criteria that we use to decompose the hypergraph. In some sense, the hypergraph decomposition (along with the modified Cauchy–Schwarz trick and Kikuchi matrices) can be thought of as a precise way to "fix" this high expected partial derivative. For longer chains, there is once again an intimate relationship between the existence of a violating expected partial derivative and a certain "denser-than-anticipated" combinatorial structure (analogous to heavy pairs) being present in the chains we construct. For larger chains, this structure is a more complicated to describe, but an analogous chain decomposition for this structure accomplishes the same job.

More precisely, we generalize the decomposition of Section 12.3.4 as follows. As done in Section 12.3.4, we shall think of an $(r+1)$-chain in $\mathcal{H}_i^{(r+1)}$ as being split into two subchains, the "first link" in H_i and then the rest of the chain, which is an r-chain. As before, our decomposition shall decompose the r-chain part only, and this induces a decomposition of the $(r+1)$-chains in $\mathcal{H}_i^{(r+1)}$. Recall that in Section 12.3.4, we decomposed a 1-chain (u, C, w) by picking a Q where $Q_1 \in C$ and $Q_2 = w$. Notice that Q only contains one element of the hyperedge C; there was no need to do a further decomposition to handle, e.g., heavy triples $Q = (Q_1, Q_1', Q_2)$ where $\{Q_1, Q_1'\} = C$ and $Q_2 = w$.

Now, we have r-chains $(u, C_1, w_1, \ldots, C_r, w_r)$, and we shall decompose if there is a $Q = (Q_1, \ldots, Q_{r+1}) \in ([n] \cup \{\star\})^r \times [n]$ such that (1) Q is heavy, i.e., is contained in many r-chains, meaning that (a) $Q_{h+1} = w_r$, and so in particular $Q_{h+1} \neq \star$, and (b) $Q_h \in C_h$ for $h = 1, \ldots, r$; and (2) Q is *contiguous*, meaning that if $h \in [r+1]$ is the minimal h such that $Q_h \neq \star$, then $Q_{h'} \neq \star$ for all $h' \geq h$, i.e., Q has $\star$'s followed by only non-$\star$ entries.

Condition (1) above is a somewhat natural extension of the decomposition method in Section 12.3.4, but condition (2) is trickier. It turns out (in a somewhat subtle way) that because the H_i's are matchings, if there is a violating expected partial derivative, then not only is there a heavy Q, but there must be a heavy *contiguous* Q. In a sense (that can be made precise), the contiguous Q's are *irreducible* violations and thus it is enough to only handle them.

12.4 Proof of Theorem 8: from LCCs to XOR formulas

We now present the proof of Theorem 8 for the case of $\mathbb{F} = \mathbb{F}_2$. The proof is spread over Sections 12.4 to 12.7 and follows the steps in the warmup. In the current section, we define r-chains and the family of XOR instances associated to the LCC that we wish to refute. Then, in Section 12.5, we decompose the r-chains, and thereby decompose the $(r+1)$-chains forming the constraints in the XOR instance. Then, in Section 12.6, we define the Kikuchi matrices and finish the argument up to the proof of the row pruning lemma, Lemma 12.6.4, an analogue of Lemma 12.3.4 that is the key technical lemma. Finally, in Section 12.7, we prove Lemma 12.6.4.

Let $\mathcal{L}\colon \mathbb{F}_2^k \to \mathbb{F}_2^n$ be $(3,\delta,\varepsilon)$-locally correctable. Without loss of generality, by Fact 3.3.10 we can assume that $\mathcal{L}$ is $(3,\delta')$-normally decodable, where $\delta' \geq \delta/6$ and $n' = 2n$. For the remainder of the proof, we will redefine δ to be δ', and n to be $2n$. We shall also think of the code $\mathcal{L}\colon \mathbb{F}_2^k \to \mathbb{F}_2^n$ as a map $\mathcal{L}\colon \{-1,1\}^k \to \{-1,1\}^n$.

We will now define satisfiable XOR formulas Φ associated with the linear code $\mathcal{L}$. Let $\mathcal{L} : \{-1,1\}^k \to \{-1,1\}^n$ be a linear $(3,\delta)$-normally correctable code. Recall that without loss of generality, $\mathcal{L}$ is systematic, meaning that the first k bits of $\mathcal{L}$ are the message bits. In particular, for every $b \in \{-1,1\}^k$, there is a unique $x \in \mathcal{L}$ such that $x|_{[k]} = b$. We can thus generate $x \leftarrow \mathcal{L}$ uniformly at random by first choosing $b \leftarrow \{-1,1\}^k$ uniformly at random, and then setting x to be the unique extension of b.

Since $\mathcal{L}$ is a linear $(3,\delta)$-normally correctable code, there exist 3-uniform hypergraph matchings $H_1,\dots,H_n$, each of size exactly δn, such that every $x \in \mathcal{L}$ satisfies the following system of 4-XOR constraints, i.e., each constraint has arity 4:

$$\forall u \in [n], C \in H_u,\ x_C x_u = 1\,. \tag{12.5}$$

In the proof, we will think of each H_u as being a *directed* and *weighted* 3-uniform hypergraph (Definition 3.2.2). Namely, for each hyperedge $\{v_1,v_2,v_3\} \in H_u$, we define the weight of the *ordered tuple* (v_1,v_2,v_3) to be $\mathrm{wt}_{H_u}(v_1,v_2,v_3) := \frac{1}{6\delta n}$. For (v_1,v_2,v_3) with $\{v_1,v_2,v_3\} \notin H_u$, we additionally define $\mathrm{wt}_{H_u}(v_1,v_2,v_3) = 0$, and so $\sum_{(v_1,v_2,v_3)} \mathrm{wt}_{H_u}(v_1,v_2,v_3) = 1$. Directed and weighted hypergraphs will become important when proving Theorem 10, and so we will state our definitions and intermediate lemmas in terms of directed and weighted hypergraphs so that we may reuse them when we prove Theorem 10.

We will construct an XOR formula by *long chain* derivations. Intuitively, a long chain derivation starts from the natural XOR constraints (12.5) and derives new ones by chaining together t constraints with an appropriate combinatorial structure. Below, we formalize the set of constraints in this formula as a family of hypergraphs built from the H_u's.

Definition 12.4.1 (t-chain hypergraph $\mathcal{H}_u^{(t)}$)**.** Let $t \geq 1$ be an integer. For any $u \in [n]$, let $\mathcal{H}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{H}_u^{(t)}}\colon [n]^{3t+1} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t+1$ tuples of the form $C = (u_0, v_1, v_2, u_1, \dots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2}, u_t)$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \prod_{h=0}^{t-1} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1})\,.$$

For a t-chain C, we call u_0 the head, the u_h's the *pivots* for $1 \leq h \leq t-1$, and u_t the *tail* of the chain C. The monomial associated to C, which we denote by g_C, is defined to be $x_{u_t} \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$.

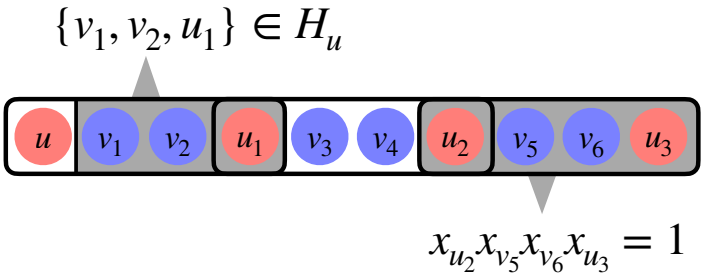

Figure 12.3: A 3-chain. The pairs of blue vertices are the "uncanceled vertices", and the red vertices are the "pivots". Note that for any $x \in \mathcal{L}$, we have $x_{u_h} x_{v_{2h+1}} x_{v_{2h+2}} x_{u_{h+1}} = 1$.

We note that for any $u \in [n]$, $\mathcal{H}_u^{(1)}$ is equivalent to H_u, i.e., $\mathcal{H}_u^{(1)} = \{u\} \times H_u$.

The following simple observation helps us understand the combinatorial structure in the chains.

Observation 12.4.2. Let $x = \mathcal{L}(b)$ for a linear LCC over $\mathbb{F}_2$ with $\{H_u\}_{u\in[n]}$ being the associated matchings. Then, for any t-chain C with head u, x satisfies $x_u g_C = 1$.

Proof. We know that x satisfies $x_{u_h} x_{v_{2h+1}} x_{v_{2h+2}} x_{u_{h+1}} = 1$ for every $0 \le h \le t-1$. Taking products of the left-hand sides of each of these t equations, we observe that for every $1 \le h \le t-1$, x_{u_h} is "squared out" (since $x_v^2 = 1$ for every $v \in [n]$), and this finishes the proof. □

Building chains iteratively. It is useful to think of t-chains as being built by extending smaller chains by iteratively adding hyperedges to the head (i.e., to the left). The following notation and observation formalizes this.

Definition 12.4.3 (Extending Chains)**.** For the t-chain hypergraph $\mathcal{H}^{(t)}$ built from 3-uniform matchings $H_1, H_2, \ldots, H_n$ on $[n]$, we define $H_u \circ \mathcal{H}^{(t+1)}$ as:

$$H_u \circ \mathcal{H}^{(t)} = \cup_{w_0\in[n]} \left\{(u, v_1, v_2, C) \mid C \in \mathcal{H}_{u_1}^{(t)}, \{v_1, v_2, u_1\} \in H_u\right\} .$$

We extend this definition to weighted hypergraphs in the analogous way.

Observation 12.4.4. For $t \ge 1$, let $\mathcal{H}^{(t)}$ be the t-chain hypergraph built from 3-matchings $H_1, H_2, \ldots, H_n$ on $[n]$. Then, $\mathcal{H}^{(t+1)} = \cup_{u\in[n]} H_u \circ \mathcal{H}^{(t)} = \cup_{u\in[n]} \mathcal{H}_u^{(t')} \circ \mathcal{H}^{(t-t')}$ for any $0 < t' < t$.

Chains that fix some positions. We will often refer to the set of chains where some of the links, i.e., pairs (v_{2h+1}, v_{2h+2}) are forced to contain some $v \in [n]$. Towards this, we introduce the following terminology.

Definition 12.4.5 (Chains containing Q)**.** Let t, r be integers with $t \le r$. For any $Q = (Q_1, \ldots, Q_t, Q_{t+1}) \in \{[n] \cup \star\}^{t+1}$, we say that a length $3r+1$ tuple $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, u_{t-1}, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r)$ contains Q, denoted by $Q \subseteq C$, if $Q_{t+1} \in \{\star, u_r\}$ and for $1 \le h \le t$, if $Q_h \ne \star$, then either $Q_h = v_{2(r-1-t+h)+1}$ or $Q_h = v_{2(r-1-t+h)+2}$.

We say that a Q is *contiguous* if there exists $s \le t$ such that $Q_h \ne \star$ for every $h \ge s+1$ and $Q_h = \star$ for every $1 \le h \le s$, i.e., the first s entries are $\star$, and the remaining entries are non-$\star$. We note that by definition, $Q_{t+1} \ne \star$ always.

We say that Q is *complete* if Q does not contain any $\star$. We say that $Q' \supseteq Q$ if whenever $Q_h \ne \star$, $Q'_h = Q_h$. We define the size $|Q|$ to be the number of coordinates in Q that do not equal $\star$.

We also prove a simple bound on the total weight of the hyperedges in $\mathcal{H}_u^{(t)}$.

Observation 12.4.6. For any $t \geq 1$ and $u \in [n]$, it holds that $\sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 1$.

Proof. The proof is by induction. The base case of $t = 1$ is simple, as by definition we have

$$\sum_{C \in [n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(C) = \sum_{(u,C) \in [n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(u, C) = \sum_{C \in [n]^3} \mathrm{wt}_{H_u}(C) = 1 \,.$$

We now show the induction step. Let $C \in [n]^{3t+1}$ have tail u_t. Let S denote the set of tuples in $[n]^{3t+4}$ that extend C, i.e., the first $3t+1$ coordinates are C. We observe that $S = C \times [n]^3$. Moreover, we have

$$\sum_{C' \in S} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C') = \sum_{C' \in [n]^3} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \mathrm{wt}_{H_{u_t}}(C') \leq \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \,.$$

Summing over C and applying the induction hypothesis proves the claim. □

XOR Formulas from r-chains. Next, we define XOR formulas associated with $\mathcal{H}^{(r+1)}$ that are guaranteed to be satisfiable. The length of the chain depends on a parameter r, which we shall set later.

We are now ready to define the chain XOR instances.

Definition 12.4.7 (The chain XOR instance Ψ_b)**.** Let $H_1, \ldots, H_n$ be weighted 3-uniform hypergraphs. Let $k \leq n$ and $r \geq 0$ be an integer. For each $1 \leq t \leq r+1$, we define the polynomial

$$\Psi_b(x) = \sum_{i=1}^{k} \sum_{C \in [n]^{3(r+1)+1}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \cdot b_i g_C \,.$$

Note that in the above sum, if the first entry of the tuple C is not i, then $\mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) = 0$. We will omit the subscript b from Ψ_b when it is clear from context. Above, each g_C is the monomial associated with the chain C, as defined in Definition 12.4.1.

We now observe that $\Psi_b(x)$ is satisfiable and thus has a high value.

Lemma 12.4.8. *For every $b \in \{-1,1\}^k$, Ψ_b is satisfied by $x = \mathcal{L}(b)$ and thus* $\mathrm{val}(\Phi_b) = k$.

Proof. Observe that each monomial in Ψ_b is corresponds to an $(r+1)$-chain, each of which is satisfied by $x = \mathcal{L}(b)$ by Observation 12.4.2. Thus, $\mathrm{val}(\Psi_b)$ equals the total weight of chains of length $r+1$ with head in $[k]$. By Observation 12.4.6, we have that for every $i \in [k]$, the total weight of chains in $\mathcal{H}_i^{(r+1)}$ is 1. As there are k choices of i, the total weight is k. □

To finish the proof of Theorem 8, we need to argue that $\mathbb{E}_b[\mathrm{val}(\Psi_b)]$ is low when $k \geq O(\log^4 n)$. We will argue this using the following lemma, which gives a spectral certificate to bound $\mathbb{E}_b[\mathrm{val}(\Psi_b)]$.

Lemma 12.4.9 (Refuting the chain XOR instances)**.** *Let $H_1, \ldots, H_n$ be 3-uniform hypergraph matchings of size δn, and let $k \leq n$. Let $\ell, d, r \geq 1$ be parameters such that $d^{r+1} \geq n$, $\ell \geq 6d(r+1)/\delta$, and $\ell r = o(n)$. Furthermore, suppose that $k \geq 1/\delta$. Then, it holds that*

$$\mathbb{E}_{b \leftarrow \{-1,1\}^k} [\mathrm{val}(\Psi_b)] \leq \left(\frac{k(r+1)}{\delta} O(\sqrt{k \ell r \log n}) \right)^{1/2} .$$

We observe that Lemma 12.4.9 immediately proves that $k \le O_\delta(\log^5 n)$, which is a single $\log n$ factor off of the bound we wish to show to prove Theorem 8 Indeed, we set $r = O(\log n)$, $d = 2$, and $\ell = O(dr/\delta) = \delta^{-1}O(\log n)$ and apply Lemmas 12.4.8 and 12.4.9. The conditions of Lemma 12.4.9 are all satisfied, and so we have

$$k = \mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(\Psi_b)] \le \left(\frac{k(r+1)}{\delta} O(\sqrt{k\ell r \log n})\right)^{1/2} \le O\left(\frac{k^{3/4}\log^{5/4} n}{\delta^{3/4}}\right)$$
$$\implies k \le O(\log^5 n/\delta^3)\,.$$

We will prove the stronger bound of $k \le O_\delta(\log^4 n)$ claimed in Theorem 8 using a simple trick. We will show this in Section 12.6.5 by using the technical lemmas that we need to prove Lemma 12.4.9. Even though Lemma 12.4.9 is not strictly needed to prove Theorem 8, we state it on its own because we will need it in the proof of Theorem 10 later.

12.5 Smooth partitions of chains

In this section, we begin the proof of Lemma 12.4.9.

For notation, we let $\mathcal{H}^{(t)}$ be the union, over u, of $\mathcal{H}_u^{(t)}$, and $\mathrm{wt}_{\mathcal{H}^{(t)}}(\cdot) = \sum_{u\in[n]} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(\cdot)$.

Lemma 12.5.1. *Let $t \ge 1$ and $d \ge 1$ be integers. There is a subset $P_t \subseteq [n]^{t+1}$ and disjoint sets $\mathcal{T}^{(Q)} \subseteq [n]^{3t+1}$ for $Q \in P_t$ such that (1) $Q \subseteq C$ for each $C \in \mathcal{T}^{(Q)}$, and (2) $\mathrm{wt}(Q) := \sum_{C\in\mathcal{T}^{(Q)}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) \ge nd^t \cdot (\delta n)^{-t-1}$.*

We say Q is heavy *if $Q \in P_t$. Note that if Q is heavy then Q is contiguous and complete by definition.*

Finally, as a trivial case, we let $P_0 = [n]$ and for $Q = (v) \in P_0$, we let $\mathcal{T}^{(Q)} = (v)$. Here, we let $\mathrm{wt}(Q) = 1$.

Proof. The proof follows by a simple greedy algorithm. Let $S = [n]^{3t+1}$. If there exists Q such that $\sum_{C\in S: Q\subseteq C} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) \ge nd^t \cdot (\delta n)^{-t-1}$, then we remove all such C from S and add them to $\mathcal{T}^{(Q)}$. We repeat until there is no such Q remaining. We note that Q cannot be used twice in this sequence, as when we pick a Q we remove all $C \in S$ containing Q. □

Definition 12.5.2 (Partitions of the chains)**.** Let $r \ge 1$ be an integer. For each $1 \le t \le r$ and heavy $Q \in P_t$, we let $\mathcal{H}^{(r,Q)}$ denote the set of tuples $C \in [n]^{3r+1}$ where:

1. C is extends a tuple in $\mathcal{T}^{(Q)}$ "backwards", i.e., $(C_{3(r-t)+1}, \ldots, C_{3r+1}) \in \mathcal{T}^{(Q)}$;
2. Q is maximal: for any $t' > t$ and $Q' \in P_{t'}$, $(C_{3(r-t')+1}, \ldots, C_{3r+1}) \notin \mathcal{T}^{(Q')}$.

Observation 12.5.3. We have that for each $t = 0, \ldots, r$, it holds that $\sum_{Q\in P_t} \mathrm{wt}(Q) \le n$, and so $\sum_{t=0}^{r} \sum_{Q\in P_t} \mathrm{wt}(Q) \le (r+1)n$.

Proof. We observe that for any $t = 0, \ldots, r$, it holds that

$$\sum_{Q\in P_t} \mathrm{wt}(Q) = \sum_{Q\in P_t} \sum_{C\in\mathcal{T}^{(Q)}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) \le \sum_{C\in[n]^{3t+1}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) = n\,. \quad \square$$

We note that Definition 12.5.2 gives a partition of the r-chains, but the polynomial $\Psi(x)$ uses a restricted set of $(r+1)$-chains, namely those that have their head in $[k]$. In the following definition, we use the partition of the r-chains to induce a partition of the special $(r+1)$-chains.

Definition 12.5.4 (Induced partition of $\mathcal{H}_i^{(r+1)}$)**.** Let $r \geq 1$ be an integer. For each $0 \leq t \leq r$ and each $Q \in P_t$, we let $\mathcal{H}_i^{(r+1,Q)}$ denote the set of length $3r+4$ tuples of the form (i, w_1, w_2, C) where $C \in \mathcal{H}^{(r,Q)}$.

Definition 12.5.5 (Bipartite XOR formulas from a smoothed partition)**.** Fix integers $r, d \geq 1$. For each $1 \leq t \leq r$ and $Q \in P_t$, we define $\Psi_{i,Q}$ as the following XOR formula with terms corresponding to $(r+1)$-chains in $\mathcal{H}^{(r+1,Q)}$ with x_Q "modded out" from the corresponding monomial.

$$\Psi_{i,Q}(x) = \sum_{C=(i,v_1,v_2,u_1,\ldots,u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \cdot x_{v_1} x_{v_2} \prod_{h=1}^{r} x_{\{v_{2h+1}, v_{2h+2}\} \setminus Q_h} \ .$$

Here, we use the convention that if $Q_h = \star$, then $\{v, v'\} \setminus Q_h \coloneqq \{v, v'\}$.

For each $0 \leq t \leq r$, let $\Psi^{(t)}(x, y) = \sum_{i=1}^{k} \sum_{Q \in P_t} b_i y_Q \Psi_{i,Q}(x)$. Finally, we let $\Psi(x, y) = \sum_{0 \leq t \leq r} \Psi^{(t)}(x, y)$; here, for every heavy $Q \in P_t$ for some $0 \leq t \leq r$ used in the smoothed partition, we introduce a new variable y_Q.

We next observe that $\Psi(x, y)$ is a relaxation of the polynomial $\Psi(x)$. Indeed, we have abused notation and labeled them both as "Ψ" for this reason. This follows from the observation is that $\Psi(x, y)$ is produced by simply replacing the monomial x_Q in $\Psi(x)$ with a new variable y_Q for each heavy Q. More formally, the following holds.

Lemma 12.5.6. *Fix $x \in \{-1, 1\}^n$. Then, there is a $y \in \{-1, 1\}^{\sum_{t=0}^{r} |P_t|}$ such that $\Psi(x, y) = \Psi(x)$.*

Proof. For each $0 \leq t \leq r$, set $y_Q = x_Q$ for every $Q \in P_t$, where $x_Q \coloneqq \prod_{h : Q_h \neq \star} x_{Q_h}$. □

We finish this section by proving the following statement, which intuitively shows that the partitions of the chains are smooth.

Lemma 12.5.7 (Smoothness of partitioned chains)**.** *Fix $i \in [k]$ and $t \in \{0, \ldots, r\}$. Let $Z \in ([n] \cup \{\star\})^{r+1} \times \{\star\}$ be a Z that has a $\star$ in the last entry. Then, $\sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \leq (\delta n)^{-|Z|}$.*

Let $Q \in P_t$ and $\mathcal{H}_i^{(r+1,Q)}$ be as defined in Definition 12.5.4. Let $Z \in ([n] \cup \{\star\})^{r+1} \times [n]$ be such that Z extends Q, i.e., $Z_{r-t+h} = Q_h$ for all $1 \leq h \leq t+1$. Then, $\sum_{C \in \mathcal{H}_i^{(r+1,Q)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)$ is at most $\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|}$ if $|Z| \leq r+1$, and at most $(\delta n)^{-r-1}$ if $|Z| = r+2$. Furthermore, if $d^{r+1} \geq n$, then $(\delta n)^{-r-1} \leq \mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|}$.

Remark 12.5.8. We remark that this is place where we need the assumption that $d^{r+1} \geq n$.

Proof. The first statement follows immediately by δ-smoothness of the original hypergraphs. Indeed, for any $u \in [n]$ and $v \in [n]$, we have that $\sum_{C \in [n]^3 : v \in C} \mathrm{wt}_{H_u}(C) \leq 1/\delta n$. We now have that

$$\begin{aligned}
&\sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \\
&\leq \sum_{\substack{(v_1, v_2, u_1) \\ Z_1 \in \{v_1, v_2\}}} \mathrm{wt}_{H_i}(v_1, v_2, u_1) \cdot \Bigg(\sum_{\substack{(v_3, v_4, u_2) \\ Z_2 \in \{v_3, v_4\}}} \mathrm{wt}_{H_{u_1}}(v_3, v_4, u_2) \Bigg(\cdots \Bigg(\sum_{\substack{(v_{2r+1}, v_{2r+2}, u_{r+1}) \\ Z_r \in \{v_{2r+1}, v_{2r+2}\}}} \mathrm{wt}_{H_{u_r}}(v_{2r+1}, v_{2r+2}, u_{r+1}) \Bigg) \cdots \Bigg) \Bigg).
\end{aligned}$$

We notice that the h-th term is at most $1/\delta n$ if $Z_h \neq \star$, and otherwise it is at most 1. So, in total, we get a bound of $(\delta n)^{-|Z|}$.

We now prove the second part of the statement. Let $|Q| = t+1$. We have two cases.

Case 1: Z does not contain a $\star$ entry. This means that $|Z| = r+2$. Let $Z' \in [n]^{r+1} \times \{\star\}$ be Z with the last entry replaced by a $\star$, i.e., $Z'_h = Z_h$ for all $1 \le h \le r+1$, and $Z'_{r+2} = \star$. We observe that

$$\sum_{C \in \mathcal{H}_i^{(r+1,Q)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le \sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le \sum_{C \in \mathcal{H}_i^{(r+1)} : Z' \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le (\delta n)^{-|Z'|} = (\delta n)^{-r-1},$$

where we use the first statement that we have already shown. To finish the argument in this case, we need to argue that $\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|} \ge (\delta n)^{-r-1}$. Indeed, we have by definition that $\mathrm{wt}(Q) \ge n d^{|Q|-1} (\delta n)^{-|Q|}$, and so

$$\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|} \ge \frac{1}{\delta} d^{|Z|-1} (\delta n)^{-|Z|} = (\delta n)^{-r-1} \cdot \frac{1}{\delta^2 n} d^{r+1}.$$

Thus, the desired inequality holds if $d^{r+1} \ge n$.

Case 2: Z contains a $\star$ entry. This means that $|Z| \le r+1$. Then, we have that $Z = (Z^{(1)}, \star, Z^{(2)}, Q)$, where $Z^{(2)}$ contains no $\star$ entries.

We observe that each $C \in \mathcal{H}_i^{(r+1,Q)}$ with $Z \subseteq C$ can be split into 3 parts: $C = (i, C^{(1)}, C^{(2)}, C^{(3)})$, where $C^{(3)} \in \mathcal{T}^{(Q)}$ is a length t chain, $(i, C^{(1)})$ is a length $|Z^{(1)}|$ chain with head i, and $C^{(2)}$ is a length $r - t - |Z^{(1)}|$ chain whose head is the tail of $C^{(1)}$ and whose tail is the head of $C^{(3)}$. By δ-smoothness, $\sum_{C^{(1)} : Z^{(1)} \subseteq C^{(1)}} \mathrm{wt}_{\mathcal{H}_i^{(|Z^{(1)}|)}}(i, C^{(1)}) \le (\delta n)^{-|Z^{(1)}|}$.

We either have that $(Z^{(2)}, Q)$ is Q, i.e., $Z^{(2)}$ is empty, or that $(Z^{(2)}, Q)$ is not Q. In the first case, $\sum_{C^{(3)} \in \mathcal{T}^{(Q)}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C^{(3)}) = \mathrm{wt}(Q)$ by definition (note that if $t = 0$, then $C^{(3)}$ is just the single vertex v where $Q = (v)$, and we have defined $\mathrm{wt}(Q) = 1$). In the second case, we observe that by Definitions 12.5.2 and 12.5.4, $(Z^{(2)}, Q)$ cannot be heavy. Indeed, if it was, then either $C^{(3)} \in \mathcal{T}^{(Z^{(2)},Q)}$, and so $C \in \mathcal{H}_i^{(r+1,(Z^{(2)},Q))}$, or else there is some other Q' with $|Q'| = |Z^{(2)}| + t + 1$ with $C^{(3)} \in \mathcal{T}^{(Q')}$, in which case we would have $C \in \mathcal{H}_i^{(r+1,Q')}$. We note that here we must use that Z contains at least one $\star$, so that $|Z^{(2)}| + |Q| \le r+1$. This is because all heavy Q' have $|Q'| \le r+1$, as they are defined for the length r-chains.

Thus, $(Z^{(2)}, Q)$ cannot be heavy. It then follows that $\sum_{C^{(3)} : C^{(3)} \notin \mathcal{T}^{(Q')} \ \forall Q' \in P_{t+|Z^{(2)}|}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C^{(3)}) \le n d^{|Z^{(2)}|+t} (\delta n)^{-|Z^{(2)}|-t-1} \le \mathrm{wt}(Q) d^{|Z^{(2)}|} (\delta n)^{-|Z^{(2)}|}$. We note that any $C \in \mathcal{H}_i^{(r+1,Q)}$ must have $C^{(3)} \notin \mathcal{T}^{(Q')} \ \forall Q' \in P_{t+|Z^{(2)}|}$, as otherwise we would violate Item (2) in Definition 12.5.2 since $|Z^{(2)}| \ge 1$.

To finish the proof, we observe that once $C^{(1)}$ and $C^{(3)}$ are chosen, the total weight of all "valid" $C^{(2)}$, i.e., $C^{(2)}$'s that could complete the chain to form $C \in \mathcal{H}_i^{(r+1,Q)}$, is at most $1/\delta n$. Indeed, this is because the head of $C^{(2)}$ is the tail of $C^{(1)}$ and its tail is the head of $C^{(3)}$, and the total weight of all length h chains, for any h, with a fixed head u and fixed tail v is at most $1/\delta n$ by δ-smoothness. Thus, in total, we have shown that $\sum_{C \in \mathcal{H}_i^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le (\delta n)^{-|Z^{(2)}|} \cdot (\delta n)^{-1} \cdot \mathrm{wt}(Q) d^{|Z^{(2)}|} (\delta n)^{-|Z^{(2)}|} = \mathrm{wt}(Q) \cdot d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|}$. □

12.6 Spectral refutation via Kikuchi matrices

In Section 12.5, we defined polynomials $\Psi^{(t)}(x, y)$ and a map from $x \mapsto y$ such that $\Psi(x) = \sum_{t=0}^{r} \Psi^{(t)}(x, y)$ when y is the image of x under this map. Thus, to prove Lemma 12.4.9, we need

to upper bound $\mathbb{E}_b[\mathrm{val}(\sum_{t=0}^{r}\Psi^{(t)}(x,y))]$. In this section, we will use the Kikuchi matrix method to bound this quantity, thus proving Lemma 12.4.9.

12.6.1 Step 1: the Cauchy–Schwarz trick

First, we show that we can relate $\sum_{t=0}^{r}\Psi^{(t)}(x,y)$ to a certain "cross-term" polynomial obtained via applying the Cauchy–Schwarz inequality.

Lemma 12.6.1 (Cauchy–Schwarz trick)**.** *Let M be a maximum directed matching[11][12] of $[k]$ and let f_M be the cross-term polynomial defined as*

$$f_M^{(t)} = \sum_{\{i,j\}\in M} b_i b_j \sum_{Q\in P_t} \frac{1}{\mathrm{wt}(Q)}\Psi_{i,Q}(x)\Psi_{j,Q}(x)\,,$$
$$f_M = \sum_{t=0}^{r} f_M^{(t)}\,.$$

Then for every x, y with ± 1 values, it holds that

$$\left(\sum_{t=0}^{r}\Psi^{(t)}(x,y)\right)^2 \le n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2k\mathbb{E}_M[f_M]\right),$$

where the expectation $\mathbb{E}_M$ is over a uniformly random maximum directed matching M.

Proof. We will first apply the Cauchy–Schwarz inequality to eliminate the y variables:

$$\left(\sum_{t=0}^{r}\Psi^{(t)}(x,y)\right)^2 = \left(\sum_{t=0}^{r}\sum_{Q\in P_t} y_Q\cdot\sqrt{\mathrm{wt}(Q)}\left(\sum_{i=1}^{k} b_i\frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)\right)^2$$
$$\le \left(\sum_{t=0}^{r}\sum_{Q\in P_t} y_Q^2\mathrm{wt}(Q)\right)\left(\sum_{t=0}^{r}\sum_{Q\in P_t}\left(\sum_{i=1}^{k} b_i\frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)^2\right).$$

By Observation 12.5.3, this is at most

$$\le n(r+1)\left(\sum_{t=0}^{r}\sum_{Q\in P_t}\left(\sum_{i=1}^{k} b_i\frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)^2\right)$$
$$\le n(r+1)\left(\sum_{t=0}^{r}\sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{i,j=1}^{k} b_i b_j\Psi_{i,Q}\Psi_{j,Q}\right)$$
$$\le n(r+1)\left(\sum_{t=0}^{r}\sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{i=1}^{k}\Psi_{i,Q}^2 + \sum_{t=0}^{r}\sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{i\neq j\in[k]} b_i b_j\Psi_{i,Q}\Psi_{j,Q}\right).$$

[11] A directed matching is a matching, only the edges are additionally directed

[12] This is a perfect matching if k is even, and will leave one element of $[k]$ unmatched if k is odd.

By Lemma 12.5.7, we have that

$$|\Psi_{i,Q}(x)| \le \sum_{C \in \mathcal{H}_i^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le \mathrm{wt}(Q) \cdot (\delta n)^{-1},$$

Hence, $\sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{i=1}^{k} \Psi_{i,Q}^2 \le \frac{k}{\delta^2 n^2} \sum_{t=0}^{r} \sum_{Q \in P_t} \mathrm{wt}(Q) \le \frac{k(r+1)}{\delta^2 n}$.

To finish the proof, we observe that the probability that a pair (i,j) is contained in a directed matching M is at least $\frac{1}{2k}$. □

12.6.2 Step 2: defining the Kikuchi matrices

It thus remains to bound $\mathbb{E}_b[\mathrm{val}(f_M)]$ for an arbitrary directed maximum matching M.

We define the Kikuchi matrices that we consider below.

Definition 12.6.2. Let $i, j \in [k]$ and $t \in \{0, \dots, r\}$. Let $Q \in P_t$.

Let $C = (i, v_1, v_2, u_1, v_3, v_4, \dots, u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}$ and $C' = (j, v_1', v_2', u_1, v_3', v_4', \dots, u_{r+1}) \in \mathcal{H}_j^{(r+1,Q)}$. We let $A_{i,j}^{(C,C',Q)} \in \{0,1\}^{\binom{[n]}{\ell}^{2r+2}}$ be the matrix with rows and columns by indexed by $(2r+2)$-tuples of sets $(S_0, \dots, S_r, S_0', \dots, S_r')$ of size exactly ℓ defined as follows.

We set $A_{i,j}^{(C,C',Q)}((S_0, \dots, S_r, S_0', \dots, S_r'), (T_0, \dots, T_r, T_0', \dots T_r'))$ equal to 1 if the following holds, and otherwise we set this entry to be 0. In what follows, we let $C_h = \{v_{2h+1}, v_{2h+2}\}$, and we note that $|C_h| = 2$ for any chain with nonzero weight, by Definition 3.2.1.

1. For $h = 0, \dots, r-t$, we have $S_h \oplus T_h = C_h$ and $v_{2h+1} \in S_h$, $v_{2h+2} \in T_h$.
2. For $h = 0, \dots, r-t$, we have $S_h' \oplus T_h' = C_h'$ and $v_{2h+1}' \in S_h'$, $v_{2h+2}' \in T_h'$,
3. For $h = 1, \dots, t$, the following holds. Let $w_h = C_{r-t+h} \setminus Q_h$, and $w_h' = C_{r-t+h}' \setminus Q_h$. We have $S_{r-t+h} = R \cup \{w_h\}$, $T_{r-t+h} = R \cup \{w_h'\}$, and $S_{r-t+h}' = T_{r-t+h}'$.[13]

We let $A_{i,j}^{(t)} = \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \cdot A_{i,j}^{(C,C',Q)}$ and $A_{i,j} = \sum_{t=0}^{r} \frac{1}{D_t} A_{i,j}^{(t)}$, where $D_t = \binom{n-2}{\ell-1}^{2r+2-t} \cdot \binom{n}{\ell}^{t}$. For any matching M on $[k]$, let $A_M = \sum_{(i,j) \in M} b_i b_j A_{i,j}$. We will abuse notation and let $A \coloneqq A_M$.

The following lemma shows that we can express $f_M(x)$ as a (scaling of a) quadratic form on the matrix $A^{(t)}$.

Lemma 12.6.3. *Let $x \in \{-1,1\}^n$, and let $x' \in \{-1,1\}^N$, where $N = \binom{n}{\ell}^{2r+2}$, denote the vector where the $(S_0, S_1, \dots, S_r, S_0', S_1', \dots, S_r')$-th entry of x' is $\prod_{h=0}^{r} x_{S_h} x_{S_h'}$. Let $i, j \in [k]$ and $t \in \{0, \dots, r\}$. Let $Q \in P_t$, and let let $C = (i, v_1, v_2, u_1, v_3, v_4, \dots, u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}$ and $C' = (j, v_1', v_2', u_1, v_3', v_4', \dots, u_{r+1}) \in \mathcal{H}_j^{(r+1,Q)}$. Then,*

$$x'^{\top} A_{i,j}^{(C,C',Q)} x' = D_t x_{v_1} x_{v_2} \prod_{h=1}^{r} x_{\{v_{2h+1}, v_{2h+2}\} \setminus Q_h} \cdot x_{v_1'} x_{v_2'} \prod_{h=1}^{r} x_{\{v_{2h+1}', v_{2h+2}'\} \setminus Q_h},$$

[13] It is possible that one could have $w_h = w_h'$ here. In that case, we pick a canonical extra vertex v, and require that $v \notin R$ as well. This is to ensure that the number of choices here for S_{r-t+h} and S_{r-t+h}' is *exactly* $\binom{n-2}{\ell-1}\binom{n}{\ell}$; otherwise it would be $\binom{n-1}{\ell-1}\binom{n}{\ell}$. The difference in the two cases is immaterial but it is convenient to have an exact count.

i.e., the product of the monomials associated to C and C', modded out by Q_h, where $D_t = \binom{n-2}{\ell-1}^{2r+2-t} \cdot \binom{n}{\ell}^t$. Moreover, for any matrix $B_{i,j}^{(C,C',Q)}$ obtained by "zeroing out" exactly αD_t entries of $A_{i,j}^{(C,C',Q)}$, the equality holds with a factor of $1-\alpha$ on the right.

In particular, $x'^\top A x' = f_M(x)$.

Proof. Let $\vec{S} = (S_0, S_1, \dots, S_r, S'_0, S'_1, \dots, S'_r)$ and $\vec{T} = (T_0, \dots, T_r, T'_0, \dots T'_r)$ be such that $A_{i,j}^{(\vec{C},\vec{C}',Q)}(\vec{S},\vec{T}) = 1$. Then, we have that

$$\begin{aligned} x'_{\vec{S}} x'_{\vec{T}} &= \prod_{h=0}^{r} x_{S_h} x_{T_h} x_{S'_h} x_{T'_h} = \prod_{h=0}^{r-t} x_{S_h \oplus T_h} x_{S'_h \oplus T'_h} \prod_{h=1}^{t} x_{S_{r-t+h} \oplus T_{r-t+h}} x_{S'_{r-t+h} \oplus T'_{r-t+h}} \\ &= \prod_{h=0}^{r-t} x_{C_h} x_{C'_h} \prod_{h=1}^{t} x_{C_{r-t+h} \setminus Q_h} x_{C'_{r-t+h} \setminus Q_h} \ , \end{aligned}$$

which is equal to the product of monomials on the right-hand side of the equation we wish to show.

It thus remains to argue that $A_{i,j}^{(\vec{C},\vec{C}',Q)}$ has exactly D_t nonzero entries. We observe that, for each $h = 0, \dots, r-t$, there are exactly $\binom{n-2}{\ell-1}$ pairs (S_h, T_h) such that $S_h \oplus T_h = C_h$ with $v_{2h+1} \in S_h$ and $v_{2h+2} \in T_h$. Indeed, this is because we must simply choose a set of size $\ell - 1$ that does not contain either of v_{2h+1} and v_{2h+2}, and then this determines S_h and T_h.

For $h = 1, \dots, t$, there are exactly $\binom{n-2}{\ell-1}$ choices of (S_{r-t+h}, T_{r-t+h}). Indeed, this is because S_{r-t+h} must contain w_h and T_{r-t+h} must contain w'_h. Note that if $w_h = w'_h$, then there are actually $\binom{n-1}{\ell-1}$ choices! However, using the slightly modified definition of the matrix in the footnote in Definition 12.6.2, we can again force there to be exactly $\binom{n-2}{\ell-1}$ choices. Finally, there are $\binom{n}{\ell}$ choices for (S'_{r-t+h}, T'_{r-t+h}), as we must have $S'_{r-t+h} = T'_{r-t+h}$.

Combining, we see that $D_t = \binom{n-2}{\ell-1}^{2(r-t+1)} \cdot (\binom{n-2}{\ell-1}\binom{n}{\ell})^t = \binom{n-2}{\ell-1}^{2r+2-t} \binom{n}{\ell}^t$, as required. □

12.6.3 Step 3: finding a regular submatrix of the Kikuchi matrix

By Lemma 12.6.3, in order to upper bound $\mathbb{E}_b[\mathrm{val}(f_M)]$, it suffices to bound $\mathbb{E}_b[\|A\|_{\infty\to 1}] \le N \mathbb{E}_b[\|A\|_2]$, where $N = \binom{n}{\ell}^{2r+2}$; here, we use that $\|A\|_{\infty\to 1} \le N\|A\|_2$ always holds.

To bound $\|A\|_2$, we will write $A = \sum_{(i,j)\in M} b_i b_j A_{i,j}$ and apply Fact 3.4.2. To do this, we need to bound $\|A_{i,j}\|_2$, which we shall do by upper bounding the maximum ℓ_1-norm of any row/column of the matrix. In turns out there are some rows that indeed have a large ℓ_1-norm. To handle this issue, we shall zero out the "bad rows", as follows. To do this, we will need to use the following technical lemma, proven in Section 12.7, that bounds the expected ℓ_1-norm of a row and the conditional expectation given that the row has a nonzero entry in a specific matrix $A_{i,j}^{(C,C',Q)}$.

Lemma 12.6.4 (First and conditional moment bounds)**.** *Fix $r \ge 1$, $i, j \in [k]$, and let $\mathcal{H}_i^{(r+1)}$ and $\mathcal{H}_j^{(r+1)}$ denote the $(r+1)$-chain hypergraph with heads in i and j respectively. Let $\cup_{t=0}^{r} \cup_{Q\in P_t} \mathcal{H}_i^{(r+1,Q)}$ be a smooth partition of $\mathcal{H}_i^{(r+1)}$, as defined in Definitions 12.5.2 and 12.5.4. Let $A_{i,j}$ be the Kikuchi matrix defined in Definition 12.6.2, which depends on r, i, j, and the pieces $\cup_{Q\in P_t} \mathcal{H}^{(r+1,Q)}$ of the refinement, and the matching M.*

Let $\vec{S} = (S_0, \dots, S_r, S'_0, \dots, S'_r) \in \binom{[n]}{\ell}^{2r+2}$ *be a row of the matrix, and let* $\deg_{i,j}(\vec{S})$ *denote the* ℓ_1*-norm of the* $\vec{S}$*-th row of* $A_{i,j}$*. Then,*

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \le \frac{1}{N \cdot \delta n},$$

where $N = \binom{n}{\ell}^{2r+2}$.

Furthermore, let $t \in \{0, \dots, r\}$, $Q \in P_t$, *and* $C \in \mathcal{H}_i^{(r+1,Q)}$ *and* $C' \in \mathcal{H}_j^{(r+1,Q)}$. *Let* $\mathcal{D}_{C,C',Q}$ *denote the uniform distribution over rows of* $A_{i,j}^{(C,C',Q)}$ *that contain a nonzero entry. Then, if* $d^{r+1} \ge n$ *and* $\ell \ge 2d(r+1)/\delta$, *it holds that*

$$\mathbb{E}_{\vec{S} \sim \mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \le \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{4}{N \delta n}.$$

Let us now use Lemma 12.6.4 to argue the following. For a sufficiently large constant Γ, there exist submatrices $B_{i,j}^{(C,C',Q)}$, i.e., a $\{0,1\}$-matrix where $B_{i,j}^{(C,C',Q)}(\vec{S},\vec{T}) = 1$ implies $A_{i,j}^{(C,C',Q)}(\vec{S},\vec{T}) = 1$, such that (1) each $B_{i,j}^{(C,C',Q)}$ contains exactly $D_t/2$ nonzero entries, and (2) the ℓ_1-norm of any row/column of $B_{i,j}$ (defined analogously to $A_{i,j}$) is at most $\frac{\Gamma}{N \cdot \delta n}$.

We do this as follows. First, we observe that $A_{i,j}^{(C,C',Q)}(\vec{S},\vec{T}) = A_{j,i}^{(C',C,Q)}(\vec{R},\vec{W})$, where $\vec{R} = (S'_0, \dots, S'_r, S_0, \dots, S_r)$ and $\vec{W} = (T'_0, \dots, T'_r, T_0, \dots, T_r)$. In particular, this symmetry implies that the bounds on the moments for rows in Lemma 12.6.4 hold for columns as well.

Let $\mathcal{B}_1 = \{\vec{S} : \deg_{i,j}(\vec{S}) \ge \frac{\Gamma}{N \cdot \delta n}\}$ denote the set of bad rows with ℓ_1-norm at least $\frac{\Gamma}{N \cdot \delta n}$, and similarly let $\mathcal{B}_2$ be the same but for the columns. Applying Markov's inequality and the conditional degree bound, we see that $\mathcal{B}_1$ contains at most $O(1/\Gamma)$-fraction of the rows where $A_{i,j}^{(C,C',Q)}$ is nonzero, and similarly $\mathcal{B}_2$ contains at most $O(1/\Gamma)$-fraction of the columns where $A_{i,j}^{(C,C',Q)}$ is nonzero. Thus, after removing these rows, we still have at least $(1 - O(1/\Gamma))D_t$ nonzero entries in $A_{i,j}^{(C,C',Q)}$. When Γ is a sufficiently large constant, this is at least $1/2$, and so we can choose an arbitrary subset of *exactly* $D_t/2$ nonzero entries. We let $B_{i,j}^{(C,C',Q)}$ be the matrix with those nonzero entries.

The first property is clearly satisfied by construction. The second property is satisfied because the ℓ_1-norm of any row/column of $B_{i,j}$ is clearly at most $\frac{\Gamma}{N \cdot \delta n}$, again by construction.

12.6.4 Step 4: finishing the proof

Let $B_{i,j}^{(C,C',Q)}$ be the matrix produced in Section 12.6.3.

We let $B_{i,j}^{(t)} = \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \cdot B_{i,j}^{(C,C',Q)}$ and $B_{i,j} = \sum_{t=0}^{r} \frac{1}{D_t} B_{i,j}^{(t)}$. For any matching M on $[k]$, let $B_M = \sum_{(i,j) \in M} b_i b_j B_{i,j}$. We will abuse notation and let $B := B_M$.

By Lemma 12.6.3 and the fact that $B_{i,j}^{(C,C',Q)}$ has exactly $D_t/2$ nonzero entries of $A_{i,j}^{(C,C',Q)}$ in it, we see that for every $x \in \{-1,1\}^n$, there exists $x' \in \{-1,1\}^N$ such that $x'^{\top} B x' = \frac{1}{2} f_M(x)$. We also have that $\|B_{i,j}\|_2 \le \frac{\Gamma}{N \cdot \delta n}$, by construction in Section 12.6.3.

By Fact 3.4.2, it therefore follows that

$$\mathbb{E}_b[\mathrm{val}(f_M(x))] \le 2\mathbb{E}_b[N \|B\|_2] \le N \cdot \frac{\Gamma}{N \cdot \delta n} \cdot O(\sqrt{k \log N}) = O(\sqrt{k \ell r \log n}) \cdot \frac{1}{\delta n}$$

Hence,

$$\mathbb{E}_b[\mathrm{val}(\Psi(x,y))]^2 \le \mathbb{E}_b[\mathrm{val}(\Psi(x,y)^2)] \le n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2k\mathbb{E}_{b,M}[\mathrm{val}(f_M)]\right)$$
$$\le n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2kO(\sqrt{k\ell r\log n})\cdot\frac{1}{\delta n}\right) = \frac{k(r+1)}{\delta}\left(\frac{r+1}{\delta} + 2O(\sqrt{k\ell r\log n})\right)$$
$$\le \frac{k(r+1)}{\delta}O(\sqrt{k\ell r\log n}),$$

as $\ell \ge O(r/\delta)$ and we can assume that $k \ge 1/\delta$ (as otherwise we are already done).

12.6.5 Step 5: optimizing the $\log n$ factor and proving Theorem 8

We will now prove Theorem 8 using the tools that we have developed in Sections 12.4 to 12.6. Let $\mathcal{L}$ be a $(3,\delta)$-linear LCC in normal form (Definition 3.3.9), with 3-uniform hypergraph matchings $H_1,\dots,H_n$ each of size δn. Similar to the analysis in Section 12.2, we will use the 3-LCC $\mathcal{L}$ to construct a 2-LDC, and then we apply the lower bound of [GKST06] (Fact 3.3.4).

The reason this approach saves a single $\log n$ factor is that, in the case of 2-query linear codes, Fact 3.3.4 shows a lower bound of $\delta k \le 2\log_2 n$, which saves a factor of δ over the lower bound from spectral refutation of $\delta^2 k \le O(\log n)$ for nonlinear 2-LDCs. In our reduction, we shall produce a 2-LDC with $\delta' \sim \delta/\log n$, so this optimization saves us a $O(\log n)$ factor. As a result, we get a final lower bound of $k \le O(\log^4 n)$, as opposed to the lower bound of $k \le O(\log^5 n)$ that we obtained earlier.

We set $r = O(\log n)$, $d = 2$, and $\ell = O(dr/\delta) = \delta^{-1}O(\log n)$ and follow the steps above. We construct the polynomial Ψ (Definition 12.4.7) and then decompose Ψ into $\Psi^{(0)},\dots,\Psi^{(r)}$ (Definition 12.5.5). By Lemmas 12.4.8 and 12.6.1, we have

$$k^2 = \mathbb{E}_b[\mathrm{val}(\Psi)]^2 \le \mathbb{E}_b[\mathrm{val}(\Psi(x,y))]^2 \le \mathbb{E}_b[\mathrm{val}(\Psi(x,y)^2)] \le n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2k\mathbb{E}_{b,M}[\mathrm{val}(f_M)]\right).$$

Hence, either $k \le (r+1)^2/\delta^2 = O(\log^2 n/\delta^2)$ and we are done, or else $\mathbb{E}_{b,M}[\mathrm{val}(f_M)] \ge \frac{k}{2n(r+1)}$, and hence there exists a directed matching M such that $\mathrm{val}(f_M) \ge \frac{k}{2n(r+1)}$. Let us proceed assuming that we are in the second case.

Let us now construct the new code and argue that it is a 2-LDC. We define a map $\mathcal{L}'\colon \{0,1\}^n \to \{0,1\}^{2N}$, where $N = \binom{n}{\ell}^{2r+2}$, in an analogous way to Section 12.2. Namely, there are $2N$ entries of $\mathcal{L}'(x)$, corresponding to the rows and columns of the Kikuchi matrices in Definition 12.6.2. For each row $\vec{S} = (S_0,\dots,S_r,S'_0,\dots,S'_r)$, we let $\mathcal{L}'(x)_{\vec{S}}$ be $x_{\vec{S}} \coloneqq \prod_{t=0}^r x_{S_t}x_{S'_t}$, and similarly for the columns $\vec{T}$.

Let $L = \{i : (i,j) \in M\}$ denote the "left halves" of the edges in the matching M. Without loss of generality, we can assume that $k' \coloneqq |L| \ge \frac{k-1}{2}$, as otherwise we can swap the left and right halves of M. Let $\mathcal{L}''\colon \{-1,1\}^L \to \{-1,1\}^{2N}$ be the linear code defined from $\mathcal{L}$ as follows. For each $b \in \{-1,1\}^L$, we first extend b to be in $\{-1,1\}^k$ by setting $b_j = 1$ for all $j \notin L$ (for $b \in \{-1,1\}^L$, we shall abuse notation and think of b as in $\{-1,1\}^k$ using this trivial extension). Then, we let $x = \mathcal{L}(b)$, and finally we let $x' \coloneqq \mathcal{L}'(x)$.

We now argue that $\mathcal{L}''$ is a $(2,\delta')$-linear LDC with $\delta' = \Omega(\delta/\log n)$. Let $B_{i,j}^{(C,C',Q)}$ be the matrix produced in Section 12.6.3. Similar to the proof of Lemma 12.4.8, we observe that for any nonzero

entry $(\vec{S}, \vec{T})$ of $B_{i,j}$ and any $x = \mathcal{L}(b)$, it holds that $x_{\vec{S}} x_{\vec{T}} = b_i b_j$. Hence, for any codeword $x' \in \mathcal{L}''$, it holds that $x'_{\vec{S}} x'_{\vec{T}} = b_i$, as $b_j = 1$ since $j \notin L$. We can thus use the $B_{i,j}^{(C,C',Q)}$ matrices to decode the message bits.

Hence, it remains to argue that each $i \in L$ admits a large matching on $[2N]$. As before, let $B_{i,j}^{(t)} = \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \cdot B_{i,j}^{(C,C',Q)}$ and $B_{i,j} = \sum_{t=0}^{r} \frac{1}{D_t} B_{i,j}^{(t)}$. We also let $B = \sum_{(i,j) \in M} b_i B_{i,j}$. We will view each $B_{i,j}$ as a weighted graph, where $m_{i,j}$ denotes the total weight of the edges in $B_{i,j}$. By the observation in the previous paragraph, if we let x' be a codeword of $\mathcal{L}''$, we have that $x'^{\top} B x' = m$, where $m = \sum_{(i,j) \in M} m_{i,j}$. Moreover, $m = \frac{1}{2} \mathrm{val}(f_M) \geq \frac{k}{4n(r+1)}$.

By construction, the ℓ_1-norm of any row/column of $B_{i,j}$ is at most $\Delta = \frac{\Gamma}{N \cdot \delta n}$. We can thus find an *unweighted* matching $G_{i,j}$ of size $m_{i,j}/2\Delta$ where $G_{i,j}$ is a subgraph of $B_{i,j}$. Indeed, this follows by a simple greedy algorithm, where we pick an arbitrary edge in $B_{i,j}$ to add to $G_{i,j}$ and then remove all the neighboring edges. At each step, the total weight of all the edges we remove is at most 2Δ, and therefore we construct a matching of size at least $m_{i,j}/2\Delta$.

We are now ready to apply Fact 3.3.4. We have

$$\sum_{(i,j) \in M} |G_{i,j}| \geq \frac{1}{2\Delta} \sum_{(i,j) \in M} m_{i,j} = \frac{m}{2\Delta} \geq \frac{k}{8n(r+1)\Delta} = \frac{kN \cdot \delta}{8\Gamma(r+1)} \geq \Omega\left(\frac{\delta k N}{\log n}\right),$$

where we use that $\Gamma = O(1)$ is a constant and $r = O(\log n)$. Hence, by Fact 3.3.4, we have that

$$\frac{k-1}{2} \leq |L| \leq O\left(\frac{\log n}{\delta} \cdot \log N\right) \leq O\left(\frac{\log n}{\delta} \cdot r\ell \log n\right) = O\left(\frac{\log^4 n}{\delta^2}\right).$$

Hence, $k \leq O(\log^4 n / \delta^2)$, which finishes the proof of Theorem 8 for the case of $\mathbb{F} = \mathbb{F}_2$, up to the proof of Lemma 12.6.4.

12.7 Row pruning: proof of Lemma 12.6.4

In this section, we prove Lemma 12.6.4, restated below.

Lemma 12.7.1 (First and conditional moment bounds). *Fix $r \geq 1$, $i, j \in [k]$, and let $\mathcal{H}_i^{(r+1)}$ and $\mathcal{H}_j^{(r+1)}$ denote the $(r+1)$-chain hypergraph with heads in i and j respectively. Let $\cup_{t=0}^{r} \cup_{Q \in P_t} \mathcal{H}_i^{(r+1,Q)}$ be a smooth partition of $\mathcal{H}_i^{(r+1)}$, as defined in Definitions 12.5.2 and 12.5.4. Let $A_{i,j}$ be the Kikuchi matrix defined in Definition 12.6.2, which depends on r, i, j, and the pieces $\cup_{Q \in P_t} \mathcal{H}^{(r+1,Q)}$ of the refinement, and the matching M.*

Let $\vec{S} = (S_0, \ldots, S_r, S'_0, \ldots, S'_r) \in \binom{[n]}{\ell}^{2r+2}$ be a row of the matrix, and let $\deg_{i,j}(\vec{S})$ denote the ℓ_1-norm of the $\vec{S}$-th row of $A_{i,j}$. Then,

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{N \cdot \delta n},$$

where $N = \binom{n}{\ell}^{2r+2}$.

Furthermore, let $t \in \{0, \ldots, r\}$, $Q \in P_t$, and $C \in \mathcal{H}_i^{(r+1,Q)}$ and $C' \in \mathcal{H}_j^{(r+1,Q)}$. Let $\mathcal{D}_{C,C',Q}$ denote the uniform distribution over rows of $A_{i,j}^{(C,C',Q)}$ that contain a nonzero entry. Then, if $d^{r+1} \geq n$ and

$\ell \geq 2d(r+1)/\delta$, *it holds that*

$$\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \leq \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{4}{N\delta n}\,.$$

Proof. We begin by estimating the first moment, i.e., $\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})]$. By definition, we have that

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] = \frac{1}{N}\sum_{t=0}^{r}\frac{1}{D_t}\sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{C\in\mathcal{H}_i^{(r+1,Q)},C'\in\mathcal{H}_j^{(r+1,Q)}}\mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)\mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C')\cdot D_t$$

$$= \frac{1}{N}\sum_{t=0}^{r}\sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{C\in\mathcal{H}_i^{(r+1,Q)},C'\in\mathcal{H}_j^{(r+1,Q)}}\mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)\mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C')\,.$$

We note that the latter quantity is simply equal to $\frac{1}{N}\sum_{C\in\mathcal{H}_i^{(r+1)}}\mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)\sum_{C'\in\mathcal{H}_j^{(r+1,Q)}:C\in\mathcal{H}_i^{(r+1,Q)}}\frac{1}{\mathrm{wt}(Q)}\cdot \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C')$, where the second sum is over $C'\in\mathcal{H}_j^{(r+1,Q)}$ where Q is determined by the choice of C. We note that for any Q, $\sum_{C'\in\mathcal{H}_j^{(r+1,Q)}}\mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \leq \frac{\mathrm{wt}(Q)}{\delta n}$, and hence we conclude that

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{N}\sum_{C\in\mathcal{H}_i^{(r+1)}}\mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)\frac{1}{\delta n} \leq \frac{1}{N\cdot\delta n}\,.$$

Next, we estimate the conditional first moment. Fix a $Q\in P_t$ for some $0\leq t\leq r$, and let $C\in\mathcal{H}_i^{(r+1,Q)}, C'\in\mathcal{H}_j^{(r+1,Q)}$. We now bound $\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})]$, where $\mathcal{D}_{C,C',Q}$ is the uniform distribution over all rows $\vec{S}$ such that $A_{i,j}^{(C,C',Q)}$ has a nonzero entry. We note that there are exactly D_t such rows.

We shall proceed in two steps. First, we consider a fixed (D,D',Q') with $D\in\mathcal{H}_i^{(r+1,Q')}, D'\in\mathcal{H}_j^{(r+1,Q')}$. Let $|Q'| = t'+1$. We will upper bound the number of rows $\vec{S}$ where $A_{i,j}^{(C,C',Q)}$ and $A_{i,j}^{(D,D',Q')}$, normalized by the factor of $1/D_{t'}$. This will depend on the number of shared vertices z between these two pairs of chains, for an appropriate definition of shared vertices. Then, we will, for each choice of z, bound the total weight of the number of chains (D,D',Q') have "intersection z" with (C,C',Q), which will conclude the argument.

Step 1: bounding the normalized number of entries for a fixed (D,D',Q'). To begin, we will define the number of "shared vertices" between two pairs of chains (C,C',Q) and (D,D',Q').

Definition 12.7.2 (Left vertices)**.** Let (C,C',Q) be such that $Q\in P_t$ and $C\in\mathcal{H}_i^{(r+1,Q)}, C'\in\mathcal{H}_j^{(r+1,Q)}$. Let $C=(i,v_1,v_2,u_1,\ldots,u_{r+1})$ and $C'=(j,v_1',v_2',u_1',\ldots,u_{r+1}')$. The tuple of *left vertices* of (C,C',Q) is the sequence $(v_1,v_3,v_5,\ldots,v_{2(r-t)+1},w_1,\ldots,w_t,v_1',v_3',\ldots,v_{2(r-t)+1}')$, where $C_h = \{v_{2h+1},v_{2h+2}\} = \{w_h,Q_h\}$. We denote this sequence by $L(C,C',Q)$.

Remark 12.7.3. The reason for the above definition is the following. If $\vec{S}$ is a row where the matrix $A_{i,j}^{(C,C',Q)}$ has a nonzero entry, then the entries of $L(C,C',Q)$ (in order) are contained in the sets $(S_0,\ldots,S_{r-t},S_{r-t+1},\ldots,S_r,S_0',\ldots,S_{r-t}')$, e.g., $v_1\in S_0$, $v_3\in S_1$, $w_1\in S_{r-t+1}$, etc.

Definition 12.7.4 (Intersection patterns)**.** Let (C, C', Q) and (D, D', Q') be such that $C \in \mathcal{H}_i^{(r+1,Q)}$, $C' \in \mathcal{H}_j^{(r+1,Q)}$ and $D \in \mathcal{H}_i^{(r+1,Q')}$, $D' \in \mathcal{H}_j^{(r+1,Q')}$.

The *intersection pattern* of (C, C', Q) and (D, D', Q'), given by $Z \in \{0,1\}^{2r+2-t}$, is defined as $Z_h = 1$ if $L(C, C', Q)_h = L(D, D', Q')_h$, and it is 0 otherwise. Note that the sequences $L(C, C', Q)$ and $L(D, D', Q')$ may not have the same length; if h is "out of bounds" for $L(D, D', Q')$, then we set $Z_h = 0$.

We now fix (D, D', Q') and count the number of rows as a function of the intersection pattern Z. Let $t' = |Q'| - 1$. We have two cases. In the first case, $t \geq t'$, which implies that $|L(C, C', Q)| \leq |L(D, D', Q)|$. We observe that in order for a row $\vec{S}$ to have a nonzero entry for both pairs of chains, the following must hold:

1. for $h = 1, \ldots, r+2$ (the first $r+1$ sets), we have $\{L(C, C', Q)_h, L(D, D', Q)_h\} \subseteq S_h$,
2. for $h = r+2, \ldots, 2r+3-t$ (the next $r+1-t$ sets), we have $\{L(C, C', Q)_h, L(D, D', Q)_h\} \subseteq S'_{h-(r+2)'}$
3. for $h = 2r+3-t, \ldots, 2r+2-t'$ (the next $t-t'$ sets), we have $L(D, D', Q)_h \in S'_{h-(r+2)'}$
4. for $h = 2r+2-t'+1, \ldots, 2r+2$ (the final t' sets), we have $S'_{h-(r+2)}$ is arbitrary.

We observe that for each intersection point, i.e., an h such that $L(C, C', Q)_h = L(D, D', Q)_h$, there are $\binom{n}{\ell-1}$ choices for the corresponding set, as it needs to only contain one vertex. For each nonintersection point, i.e., an $h \in \{1, \ldots, 2r+2-t\}$ where $L(C, C', Q)_h \neq L(D, D', Q)_h$, we have $\binom{n}{\ell-2}$ choices, because the set needs to contain both vertices. Finally, we have $\binom{n}{\ell-1}$ choices for each of the $t-t'$ sets in the third case, and $\binom{n}{\ell}$ choices for the last t sets in the final case. In total, we have $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{2r+2-t-z} \binom{n}{\ell-1}^{t-t'} \binom{n}{\ell}^{t'}$.

In the second case, $t \leq t'$. We observe that by swapping the roles of t and t' above, we get a bound of $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{2r+2-t'-z} \binom{n}{\ell-1}^{t'-t} \binom{n}{\ell}^{t}$.

Now, although the above counts are different, we observe that they are within constant factors of each other. Indeed, we have

$$\frac{\binom{n}{\ell-2}^{-t'} \binom{n}{\ell-1}^{t'-t} \binom{n}{\ell}^{t}}{\binom{n}{\ell-2}^{-t} \binom{n}{\ell-1}^{t-t'} \binom{n}{\ell}^{t'}} = \left(\binom{n}{\ell-2}^{-1} \binom{n}{\ell-1}^{2} \binom{n}{\ell}^{-1} \right)^{t'-t}$$
$$= \left(\frac{\ell(n-\ell+2)}{(\ell-1)(n-\ell+1)} \right)^{t'-t} = \left(1 + \frac{n-1}{(\ell-1)(n-\ell+1)} \right)^{t'-t},$$

and this ratio is between $\frac{1}{2}$ and 2 since $|t'-t| \leq r$ and $\frac{n-1}{(\ell-1)(n-\ell+1)} \geq \frac{2}{\ell} \geq \frac{1}{\Gamma r}$ for a sufficiently large constant Γ.

Next, we observe that while we have an upper bound of $2 \cdot \binom{n}{\ell-1}^z \binom{n}{\ell-2}^{2r+2-t} \binom{n}{\ell-1}^{t-t'} \binom{n}{\ell}^{t'}$ on the number of rows, which depends on t', each entry has a scaling factor of $\frac{1}{D_{t'}}$. We now give an

upper bound on the *normalized* number of entries that does not depend on t'. We have

$$2\frac{\binom{n}{\ell-1}^{z}\binom{n}{\ell-2}^{2r+2-t-z}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}}{D_{t'}} = 2\frac{\binom{n}{\ell-1}^{z}\binom{n}{\ell-2}^{2r+2-t-z}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}}{\binom{n-2}{\ell-1}^{2r+2-t'}\cdot\binom{n}{\ell}^{t'}} = 2\left(\frac{\binom{n}{\ell-2}}{\binom{n}{\ell-1}}\right)^{2r+2-t-z}\cdot\left(\frac{\binom{n}{\ell-1}}{\binom{n-2}{\ell-1}}\right)^{2r+2-t'}$$
$$= 2\left(\frac{\ell-1}{n-\ell+2}\right)^{2r+2-t-z}\cdot\left(\frac{n(n-1)}{(n-\ell+1)(n-\ell)}\right)^{2r+2-t'}$$
$$\leq 2\left(\frac{\ell}{n}\right)^{2r+2-t-z}\cdot\left(1+\frac{O(\ell r)}{n}\right).$$

Step 2: bounding the weight of (D, D', Q') with a fixed intersection pattern Z. Let us fix the intersection pattern Z and then determine the total weight of all (D, D', Q') with $D \in \mathcal{H}_i^{(r+1,Q')}$, $D' \in \mathcal{H}_j^{(r+1,Q')}$ with these intersection points. To do this, we will apply Lemma 12.5.7.

First, we observe that fixing an intersection pattern induces a $Z^{(1)} \in \{[n] \cup \{\star\}\}^{r+1} \times \{\star\}$, simply by filling in $Z^{(1)}$'s non-$\star$ entries with the appropriate vertices of $L(C, C', Q)$. We note that such a $Z^{(1)}$ never has the tail filled in, as the tail is not a potential intersection point. By Lemma 12.5.7, this implies that the total weight of D that contain $Z^{(1)}$ is at most $(\delta n)^{-|Z^{(1)}|}$.

Next, we bound the total weight of all D' that are valid for a fixed D. We observe that $D \in \mathcal{H}_i^{(r+1,Q')}$ for some i, and hence D' must be in $\mathcal{H}_j^{(r+1,Q')}$. We note that Z induces an intersection pattern $Z^{(2)}$ on D', and moreover $Z^{(2)}$ does not intersect with the "Q'-part" of the chain D', namely the links that contain vertices from Q'. So, it follows that D' contains $(Z^{(2)}, Q')$.

By Lemma 12.5.7, we have that the total weight of all D' is at most $\mathrm{wt}(Q)d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$. As each entry in $A_{i,j}^{(D,D',Q')}$ is scaled down by a factor of $\mathrm{wt}(Q')$, the normalized weight is therefore at most $d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$.

In total, we get a bound of $(\delta n)^{-|Z^{(1)}|}\cdot d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$, which is at most $d^{|Z|}(\delta n)^{-|Z|-1}$. Here, we use that $|Z| = |Z^{(1)}| + |Z^{(2)}|$.

Putting it all together. By combining steps (1) and (2) (and paying an additional $\binom{2r+2-t}{z}$ factor to choose the nonzero entries of Z), we thus obtain the final bound of

$$\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{D_t}\sum_{z=0}^{2r+2-t}\binom{2r+2-t}{z}\cdot\left(1+\frac{O(\ell r)}{n}\right)\cdot 2\left(\frac{\ell}{n}\right)^{2r+2-t-z}\cdot d^z(\delta n)^{-z-1}$$
$$\leq \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t}\left(\frac{\ell}{n}\right)^{2r+2-t}\cdot\sum_{z=0}^{2r+2-t}(2r+2-t)^z\cdot\left(\frac{\ell}{n}\right)^{-z}\cdot d^z(\delta n)^{-z-1}$$
$$= \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t\cdot\delta n}\left(\frac{\ell}{n}\right)^{2r+2-t}\cdot\sum_{z=0}^{2r+2-t}\left(\frac{(2r+2-t)\cdot d}{\delta\ell}\right)^z$$
$$\leq \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{D_t\cdot\delta n}\left(\frac{\ell}{n}\right)^{2r+2-t},$$

where we use that $\ell \geq 2d(2r+2)/\delta$.

To finish the proof, we need to compute $\frac{D_t}{N}$. We have that

$$\frac{D_t}{N} = \frac{\binom{n-2}{\ell-1}^{2r+2-t} \cdot \binom{n}{\ell}^t}{\binom{n}{\ell}^{2r+2}} = \left(\frac{\binom{n-2}{\ell-1}}{\binom{n}{\ell}}\right)^{2r+2-t} = \left(\frac{\ell(n-\ell)}{n(n-1)}\right)^{2r+2-t}$$
$$\geq \left(\frac{\ell}{n}\right)^{2r+2-t} \cdot \left(1 - \frac{\ell-1}{n-1}\right)^{2r+2-t} \geq \left(\frac{\ell}{n}\right)^{2r+2-t} \left(1 - \frac{(\ell-1)(2r+2)}{n-1}\right) = \left(\frac{\ell}{n}\right)^{2r+2-t} \left(1 - \frac{O(\ell r)}{n}\right),$$

Thus,

$$\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \leq \left(1 + \frac{O(\ell r)}{n}\right) \frac{4}{D_t \cdot \delta n} \left(\frac{\ell}{n}\right)^{2r+2-t} \leq \left(1 + \frac{O(\ell r)}{n}\right) \frac{4}{N \cdot \delta n},$$

which finishes the proof. □

12.8 From adaptive decoders to chain XOR polynomials

In this section, we begin the proof of Theorem 10. We start with a (possibly nonlinear) smooth 3-LCC with a (possibly adaptive) decoder. First, we define an abstract notion of a (smooth) 3-LCC hypergraph collection, which captures the fact that, unlike the linear case (Theorem 8), our hyperedges might now have size 2 in addition to being of size 3. Then, we will use the hypergraph collection to define a family of chain XOR instances similar to Definition 12.4.7. Finally, we will use Kikuchi matrices to argue (Lemma 12.8.6) that any "chain XOR instance" from a 3-LCC hypergraph collection must have small value. Finally, we will show that, given a 3-LCC, we can extract a 3-LCC hypergraph collection such that the resulting chain XOR instance has high value, which finishes the proof.

We begin by defining a (δ-smooth) 3-LCC hypergraph collection. One should view this as a generalization of the standard "combinatorial" definition of (linear) 3-LCCs (Definition 3.3.9). In the below definition, the hypergraph H_u is 3-uniform and intuitively captures decoding constraints that make 3 queries; the hypergraph G_u is 2-uniform, i.e., it is a graph, and it intuitively captures decoding constraints that only make at most 2 queries.

Definition 12.8.1 (3-LCC hypergraph collection)**.** A 3-LCC hypergraph collection on $[n]$ vertices is a collection of pairs (H_u, G_u), one for each $u \in [n]$, where G_u is a (weighted and directed) 2-uniform hypergraph and H_u is a (weighted and directed) 3-uniform hypergraph[14] such that for every $u \in [n]$, $\sum_{C\in[n]^2} \mathrm{wt}_{G_u}(C) + \sum_{C\in[n]^3} \mathrm{wt}_{H_u}(C) \leq 4$ and $\sum_{C\in[n]^3} \mathrm{wt}_{H_u}(C) \leq 1$.

For each $u \in [n]$, we define the polynomial $f_u(x) = \phi_u(x) + \psi_u(x)$, where $\phi_u(x) = \sum_{C\in[n]^2} \mathrm{wt}_{G_u}(C) x_C$ is the homogeneous degree-2 component of f_u and $\psi_u(x) = \sum_{C\in[n]^3} \mathrm{wt}_{H_u}(C) x_C$ is the homogeneous degree-3 component of f_u.

We furthermore say that the hypergraph collection is δ-smooth if for every $u, v \in [n]$, $\sum_{C\in[n]^2: v\in C} \mathrm{wt}_{G_u}(C) + \sum_{C\in[n]^3: v\in C} \mathrm{wt}_{H_u}(C) \leq \frac{1}{\delta n}$

We now use the above collection of polynomials to construct chain XOR polynomials. To define these polynomials, we first define the t-chain hypergraphs $\mathcal{H}_u^{(t)}$ and $\mathcal{G}_u^{(t)}$.

[14] Note that Definition 3.2.1 requires that each tuple with nonzero weight has *distinct* vertices.

Definition 12.8.2 (t-chain hypergraph $\mathcal{H}_u^{(t)}$, Definition 12.4.1)**.** Let $t \geq 1$ be an integer, and let $(G_u, H_u)_{u\in[n]}$ denote a 3-LCC hypergraph collection. For any $u \in [n]$, let $\mathcal{H}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{H}_u^{(t)}} \colon [n]^{3t+1} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t+1$ tuples of the form $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2}, u_t)$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \prod_{h=0}^{t-1} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1})\,.$$

For a t-chain C, we call u_0 the head, the u_h's the *pivots* for $1 \leq h \leq t-1$, and u_t the *tail* of the chain C. The monomial associated to C, which we denote by g_C, is defined to be $x_{u_t} \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$. We call the t-chain hypergraph $\mathcal{H}_u^{(t)}$ "hypergraph-tailed", as the last link uses one of the hypergraphs H_v.

We note that for any $u \in [n]$, $\mathcal{H}_u^{(1)}$ is equivalent to H_u, i.e., $\mathcal{H}_u^{(1)} = \{u\} \times H_u$.

Definition 12.8.3 (t-chain hypergraph $\mathcal{G}_u^{(t)}$)**.** Let $t \geq 1$ be an integer, and let $(G_u, H_u)_{u\in[n]}$ denote a 3-LCC hypergraph collection. For any $u \in [n]$, let $\mathcal{G}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{G}_u^{(t)}} \colon [n]^{3t} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t$ tuples of the form $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2})$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \mathrm{wt}_{G_{u_{t-1}}}(v_{2(t-1)+1}, v_{2(t-1)+2}) \cdot \prod_{h=0}^{t-2} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1})\,.$$

Note that the chains in $\mathcal{G}^{(t)}$ have no tail vertex u_t. The monomial associated to C, which we denote by x_C, is defined to be $g_C = \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$. We call the t-chain hypergraph $\mathcal{G}_u^{(t)}$ "graph-tailed", as the last link uses one of the graphs G_v.

We note that for any $u \in [n]$, $\mathcal{G}_u^{(1)}$ is equivalent to G_u, i.e., $\mathcal{G}_u^{(1)} = \{u\} \times G_u$.

We are now ready to define the chain XOR instances.

Definition 12.8.4 (Chain XOR instance)**.** Let $(G_u, H_u)_{u\in[n]}$ denote a 3-LCC hypergraph collection. Let $k \leq n$ and $r \geq 0$ be an integer. For each $1 \leq t \leq r+1$, we define the "graph-tailed" polynomial

$$\Phi_b^{(t)}(x) = \sum_{i=1}^{k} \sum_{C\in[n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \cdot b_i g_C\,,$$

and we also define the "hypergraph-tailed" polynomial

$$\Psi_b(x) = \sum_{i=1}^{k} \sum_{C\in[n]^{3(r+1)+1}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \cdot b_i g_C\,.$$

We will omit the subscript b when it is clear from context. We note that in the above definitions, each g_C is the monomial associated with the chain C, as defined in Definitions 12.8.2 and 12.8.3.

Remark 12.8.5 (Iterative view of the chain construction). We can view the chains as being constructed iteratively in the following way. We start with a fixed u_0, and have 2 choices. We either pick a hyperedge $(v_1, a_2, v_2, a_2, u_1) \in H_{u_0}$, and then recurse onto u_1, or else we pick an edge $(v_1, a_2, v_2, a_2) \in G_{u_0}$, in which case the chain is in $\mathcal{G}_u^{(1)}$ and we stop.

With the above setup in hand, we can now state the main technical lemma.

Lemma 12.8.6 (Refuting the chain XOR instances)**.** *Let $(G_u, H_u)_{u\in[n]}$ denote a δ-smooth 3-LCC hypergraph collection and let $k \le n$. Let $\ell, d, r \ge 1$ be parameters such that $d^{r+1} \ge n$, $\ell \ge 6d(r+1)/\delta$, and $\ell r = o(n)$. Furthermore, suppose that $k \ge 1/\delta$. Then, for each $1 \le t \le r+1$, it holds that*

$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Phi_b^{(t)})] \le O(\sqrt{k\ell r \log n}),$$
$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Psi_b)] \le \left(\frac{k(r+1)}{\delta} O(\sqrt{k\ell r \log n})\right)^{1/2}.$$

We observe that we have already proven Lemma 12.4.9, which is the second inequality above. Thus, we only need to show the first inequality, i.e., we need to refute the graph-tailed instances, which we will do in Section 12.9.

Finally, to finish the proof of Theorem 10, it remains to argue that, given any $(3, \delta, \varepsilon)$-smooth LCC, one can extract a 3-LCC hypergraph collection such that the resulting chain XOR polynomials (Definition 12.8.4) have large value. This is captured by the following lemma.

Lemma 12.8.7. *Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a 3-LCC. Let $C'\colon \{-1,1\}^k \to \{-1,1\}^{4n}$ be defined as $C'(b) = (C(b), -C(b), 1^n, (-1)^n)$, i.e., C' is a "padded" version of C, and let $\mathrm{Dec}(\cdot)$ denote its (possibly adaptive) decoder.*

Then, there exists a 3-LCC hypergraph collection $(H_u, G_u)_{u\in[4n]}$ with the following properties.

1. *For every $1 \le u \le 4n$ and every codeword $x \in C'$, we have $f_u(x)x_u = \mathbb{E}[\mathrm{Dec}^{(x)}(u)x_u]$, where the expectation is taken over the randomness of the decoder. In particular, if C has completeness $1-\varepsilon$, then $f_u(x)x_u \ge 1-2\varepsilon$ for all $x \in C'$.*
2. *If C is systematic and has completeness $1-\varepsilon$, then for any r such that $1-2(r+1)\varepsilon > 0$, it holds that for every $b \in \{-1,1\}^k$ and $x = C'(b)$, $\Psi_b(x) + \sum_{t=1}^{r+1} \Phi_b^{(t)}(x) \ge k(1-2(r+1)\varepsilon)$.*
3. *If C is δ-smooth, then C' is $\delta/4$-smooth, and $(H_u, G_u)_{u\in[n']}$ is a (δ/c)-smooth hypergraph collection for some constant $c \ge 4$.*

We prove Lemma 12.8.7 in Section 12.8.1.

Let us now finish the proof of Theorem 10.

Proof of Theorem 10. Let C be a 3-LCC that is δ-smooth and has completeness $1-\varepsilon$. By Fact 3.3.8, by adjusting k by a factor of $\log(1/\delta)$, we can assume that C is additionally systematic. By Lemma 12.8.7, the padded code C' is $(\delta/4)$-smooth with completeness $1-\varepsilon$ and has a $(\delta/4)$-smooth uniform hypergraph collection $(H_u, G_u)_{u\in[4n]}$. Let r be such that $1-2(r+1)\varepsilon > 0$. We have that for every $b \in \{-1,1\}^k$ and $x = C'(b)$, $\Psi_b(x) + \sum_{t=1}^{r+1} \Phi_b^{(t)}(x) \ge k(1-2(r+1)\varepsilon)$.

On the other hand, by Lemma 12.8.6, it holds that

$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Phi_b^{(t)})] \le O(\sqrt{k\ell r \log n}),$$
$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Psi_b)] \le \left(\frac{k(r+1)}{\delta} O(\sqrt{k\ell r \log n})\right)^{1/2},$$

where d and ℓ are parameters chosen so that $d^r \ge n$, $\ell \ge 6dr/\delta$, and $\ell r = o(n)$.

First, let us handle the case in Theorem 10 when $\varepsilon = 0$. Here, we set $r = O(\log n)$, $d = 2$, and $\ell = O(dr/\delta) = \delta^{-1}O(\log n)$. We clearly have that all the conditions of Lemma 12.8.6 are satisfied.

Hence, we have that

$$k = k(1-2(r+1)\varepsilon) \leq \mathbb{E}_b[\Psi_b(C'(b)) + \sum_{t=1}^{r+1} \Phi_b^{(t)}(C'(b))]$$

$$\leq (r+1)\cdot O(\sqrt{k\ell r \log n}) + \left(\frac{k(r+1)}{\delta} O(\sqrt{k\ell r \log n})\right)^{1/2} \leq O\left(\sqrt{\frac{k\log^5 n}{\delta}} + \frac{k^{3/4}\log^{5/4} n}{\delta^{3/4}}\right)$$

$$\implies k \leq O(\log^5 n/\delta^3)\,,$$

which proves the statement when $\varepsilon = 0$.

Now, let us consider the case when $\varepsilon > 0$. When we apply Lemma 12.8.6, we now set parameters as follows. Let $\eta > 0$, and set r_0 be such that $r_0 + 1 = \lfloor \frac{1-\eta}{2\varepsilon} \rfloor$ and $r_1 = \log_2 n$. We then let $r = \min(r_0, r_1)$. Note that by choice of r, $\frac{1-\eta}{2\varepsilon} \geq r+1$, and so $1-2(r+1)\varepsilon \geq 2\eta$, and we also have $r \leq O(\log n)$.

Now, we set d to be such that $d^{r+1} \geq n$, so we have to set $d = n^{1/(r+1)}$. Finally, we set $\ell = dr/\delta$. We thus have that

$$2\eta k = k(1-2(r+1)\varepsilon) \leq \mathbb{E}_b[\Psi_b(C'(b)) + \sum_{t=1}^{r+1} \Phi_b^{(t)}(C'(b))]$$

$$\leq (r+1)\cdot O(\sqrt{k\ell r \log n}) + \left(\frac{k(r+1)}{\delta} O(\sqrt{k\ell r \log n})\right)^{1/2}$$

$$\leq (r+1)\cdot O(\sqrt{kn^{1/(r+1)} r^2 \log n/\delta}) + \left(\frac{k(r+1)}{\delta} O(\sqrt{kn^{1/(r+1)} r^2 \log n/\delta})\right)^{1/2}.$$

This implies that either

$$\eta^2 k \leq \frac{1}{\delta}\cdot O(n^{1/(r+1)}\log^5 n)\,,$$

or

$$\eta^4 k \leq \frac{1}{\delta^3} O(n^{1/(r+1)}\log^5 n)\,.$$

The second equation is always the dominant term, which finishes the proof. Note that the final $\log(1/\delta)$ loss comes from Fact 3.3.8. □

The remainder of this section is dedicated to proving Lemma 12.8.7, which we do in Section 12.8.1. We will prove Lemma 12.8.6 in Section 12.9, which will complete the proof of Theorem 10.

We also make the following observation, which bounds the total weight of the hyperedges in $\mathcal{H}_u^{(t)}$ and $\mathcal{G}_u^{(t)}$.

Observation 12.8.8. Let $(G_u, H_u)_{u\in[n]}$ denote a 3-LCC hypergraph collection. Then, for any $t \geq 1$ and $u \in [n]$, it holds that $\sum_{C\in[n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \leq 1$ and $\sum_{C\in[n]^{3t}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) \leq 4$.

Proof. Let us first prove the statement for $\mathcal{H}_u^{(t)}$. This follows by induction. The base case of $t = 1$ is simple, as by definition we have

$$\sum_{C\in[n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(C) = \sum_{(u,C)\in[n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(u,C) = \sum_{C\in[n]^3} \mathrm{wt}_{H_u}(C) \le 1\,.$$

We now show the induction step. Let $C \in [n]^{3t+1}$ have tail u_t. Let S denote the set of tuples in $[n]^{3t+4}$ that extend C, i.e., the first $3t+1$ coordinates are C. We observe that $S = C \times [n]^3$. Moreover, we have

$$\sum_{C'\in S} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C') = \sum_{C'\in[n]^3} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C)\mathrm{wt}_{H_{u_t}}(C') \le \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C)\,.$$

Summing over C and applying the induction hypothesis proves the claim.

Now, we prove the statement for $\mathcal{G}_u^{(t)}$. Let $C \in [n]^{3t+1}$ have tail u_t. Let S denote the set of tuples in $[n]^{3t+3}$ that extend C, i.e., the first $3t+1$ coordinates are C. We observe that $S = C \times [n]^2$. We have

$$\sum_{C'\in S} \mathrm{wt}_{\mathcal{G}_u^{(t+1)}}(C') = \sum_{C'\in[n]^2} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C)\mathrm{wt}_{G_{u_t}}(C') \le 4\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C)\,.$$

Summing over C and applying the claim for $\mathcal{H}_u^{(t)}$ then proves the claim for $\mathcal{G}_u^{(t)}$. □

12.8.1 Constructing polynomials from adaptive smoothed decoders

In this subsection, we prove Lemma 12.8.7. Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a 3-LCC with an adaptive decoder $\mathrm{Dec}(\cdot)$. The vast majority of the proof will be for proving Item (1), which will be done in two steps. First, we will prove the following lemma, which is an analogue of Item (1) in Lemma 12.8.7 but for the AND polynomial, which is defined below.

Definition 12.8.9 (AND polynomial)**.** Let $\mathrm{AND}\colon \{-1,1\}^2 \to \{0,1\}$ be the function where $\mathrm{AND}(\sigma,\sigma') = 1$ if $\sigma = \sigma' = 1$, and 0 otherwise. We note that $\mathrm{AND}(\sigma,\sigma') = \frac{1}{2}(1+\sigma)\cdot\frac{1}{2}(1+\sigma')$.

Lemma 12.8.10. *Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a 3-LCC with an adaptive decoder $\mathrm{Dec}(\cdot)$ that uses at most r bits of randomness. Then, for every $u \in [n]$, there are weight functions $\mathrm{wt}_{H_u}\colon [n]\times\{-1,1\}\times[n]\times\{-1,1\}\times[n]\times\{0,1\}^{\mathrm{r}} \to \mathbb{R}_{\ge 0}$ and $\mathrm{wt}_{G_u}\colon [n]\times\{-1,1\}\times[n]\times\{-1,1\}\times\{0,1\}^{\mathrm{r}} \to \mathbb{R}_{\ge 0}$ and bits $\sigma_{(u,v_1,a_1,v_2,a_2,v_3,\mathbf{r})} \in \{-1,1\}$, $\sigma_{(u,v_1,a_1,v_2,a_2,\mathbf{r})} \in \{-1,1\}$ such that for every $x \in C$,*

$$\sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\right) = 4\,, \tag{12.6}$$

$$\sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\right)\cdot \mathrm{AND}(a_1x_{v_1}, a_2x_{v_2}) = 1\,, \tag{12.7}$$

$$\sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C)\sigma_{(u,C)} + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\sigma_{(u,C,v_3)}x_{v_3}\right)\cdot \mathrm{AND}(a_1x_{v_1}, a_2x_{v_2}) = \mathbb{E}[\mathrm{Dec}^x(u)]\,, \tag{12.8}$$

where the expectation $\mathbb{E}[\mathrm{Dec}^x(u)]$ *is over the internal randomness of the decoder.*

Furthermore, if $\mathrm{Dec}(\cdot)$ *is* δ*-smooth, then for any* $v \in [n]$*, we have*

$$\sum_{\substack{(C,v_3)=(v_1,a_1,v_2,a_2,v_3)\\ v_1=v\vee v_2=v\vee v_3=v}} \mathrm{wt}_{H_u}(C,v_3) + \sum_{\substack{C=(v_1,a_1,v_2,a_2)\\ v_1=v\vee v_2=v}} \mathrm{wt}_{G_u}(C) \le \frac{4}{\delta n}\,.$$

We postpone the proof of Lemma 12.8.10 to Section 12.8.2, and now finish the rest of the proof of Lemma 12.8.7.

Let $C'\colon \{-1,1\}^k \to \{-1,1\}^{4n}$ be the "padded" version of C, i.e., for each $b \in \{-1,1\}^k$, $C'(b) = (C(b), -C(b), 1^n, (-1)^n)$. Note that if C is systematic, then so is C'.

Let us extend $\mathrm{Dec}(\cdot)$ to be a decoder $\mathrm{Dec}'(\cdot)$ for C' by defining its behavior on $u \in \{2n+1,\ldots,4n\}$ to be: (1) if u is a "1 bit", i.e., $u \in \{2n+1,\ldots,3n\}$, then $\mathrm{Dec}'(u)$ queries a random pair of the "padded" bits of the same sign (namely, it queries either two bits that are supposed to be 1 or two bits that are -1), and (2) if u is a "-1 bit", i.e., $u \in \{3n+1,\ldots,4n\}$, then $\mathrm{Dec}('u)$ queries a random pair of the "padded" bits that have opposite signs. We note that if the original decoder $\mathrm{Dec}(\cdot)$ has completeness $1-\varepsilon$, then so does the padded decoder $\mathrm{Dec}'(\cdot)$, and if the original decoder is δ-smooth, then the padded decoder $\mathrm{Dec}'(\cdot)$ is $(\delta/3)$-smooth.

Proof of Item (1). We are now ready to prove Item (1) in Lemma 12.8.7. Fix $u \in [3n]$. We will now construct the desired hypergraph pair (H'_u, G'_u) as follows.

First, if $u \in [4n] \setminus [2n]$ is one of the "constant" padded vertices, then this is simple. We let H'_u be empty, i.e., all weights are 0, and if $u \in \{n+1,\ldots,2n\}$ is one of the "1 bit" padded vertices, then we let G'_u denote the graph with weight $1/2n(n-1)$ on all ordered pairs of vertices (v_1,v_2) where $v_1,v_2 \in \{2n+1,\ldots,3n\}$ or $v_1,v_2 \in \{3n+1,\ldots,4n\}$. If $u \in \{2n+1,\ldots,3n\}$ is one of the "-1 bit" padded vertices, then we let G'_u denote the graph with weight $1/2n^2$ on all ordered pairs of vertices (v_1,v_2) where $v_1 \in \{2n+1,\ldots,3n\}$, $v_2 \in \{3n+1,\ldots,4n\}$ or vice-versa. It is straightforward to observe that this satisfies the desired condition, as for every $x \in C'$, $x_{v_1}x_{v_2} = 1$ if $v_1,v_2 \in \{n+1,\ldots,2n\}$ or $v_1,v_2 \in \{2n+1,\ldots,3n\}$, and in the other case $x_{v_1}x_{v_2} = -1$ holds for all codewords.

It remains to handle the case when $u \in [2n]$. We will do this for the case when $u \in [n]$, and then observe that we can handle the case of $u \in [2n] \setminus [n]$ by flipping the "sign" of the first query.

Let $u \in [n]$. We construct (H'_u, G'_u) from the pair (H_u, G_u) given to us in Lemma 12.8.10, as follows. Recall that each term $C = (v_1,a_1,v_2,a_2,\mathbf{r})$ with $\mathrm{wt}_{G_u}(C) > 0$ contributes the term $\mathrm{wt}_{G_u}(C)\sigma_C\mathrm{AND}(a_1x_{v_1}, a_2x_{v_2})$ in Eq. (12.8). We have that for any $x \in C'$,

$$\begin{aligned}\sigma_C\mathrm{AND}(a_1x_{v_1}, a_2x_{v_2}) &= \frac{1}{4}\sigma_C\left(1 + a_1x_{v_1} + a_2x_{v_2} + a_1a_2x_{v_1}x_{v_2}\right)\\ &= \frac{1}{4}\left(x_{\sigma_C^{(v_1)}}x_{1^{(v_2)}} + x_{a_1v_1}x_{\sigma_C^{(v_2)}} + x_{\sigma_C^{(v_1)}}x_{a_2v_2} + x_{\sigma_C a_1v_1}x_{a_2v_2}\right),\end{aligned}$$

where (1) for any $\sigma \in \{-1,1\}$, $x_{\sigma^{(v_1)}}$ refers to the v_1-th copy of σ, i.e., if $\sigma = 1$ then $x_{\sigma^{(v_1)}} = x_{2n+v_1}$ and if $\sigma = -1$ then $x_{\sigma^{(v_1)}} = x_{3n+v_1}$, and (2) $x_{a_1v_1}$ is x_{v_1} if $a_1 = 1$ and x_{n+v_1}, i.e., the copy of $-x_{v_1}$, if $a_1 = -1$, and similar notation is used for x_{v_2}.

Now, we add 4 edges to G'_u for each such edge in G_u. Namely, for any $C = (v_1,a_1,v_2,a_2,\mathbf{r})$ with $\mathrm{wt}_{G_u}(C) > 0$ and term $\mathrm{wt}_{G_u}(C)\sigma_C\mathrm{AND}(a_1x_{v_1}, a_2x_{v_2})$, we add the 4 edges $(\sigma_C^{(v_1)}, 1^{(v_2)})$, $(a_1v_1, \sigma_C^{(v_2)})$, $(\sigma_C^{(v_1)}, a_2v_2)$, $(\sigma_C a_1v_1, a_2v_2)$ to G_u, each with weight $\frac{1}{4}\mathrm{wt}_{G_u}(C)$. We note that it is possible to add

the same edge to G'_u multiple times: in this case, we "merge" the edges by adding their weights together.

We now process the edges in H_u to form H'_u in a similar way. The difference here is that we will add the "degree 3 term" to H'_u, and all other terms will again be added to G'_u. More formally, each term $C = (v_1, a_1, v_2, a_2, v_3, \mathbf{r})$ with $\mathrm{wt}_{H_u}(C) > 0$ contributes the term $\mathrm{wt}_{H_u}(C)\sigma_C \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{v_3}$ in Eq. (12.8). From this term, we add the edge $(\sigma_C a_1 v_1, a_2 v_2, v_3)$ to H'_u with weight $\frac{1}{4}\mathrm{wt}_{H_u}(C)$, and we add the 3 edges $(\sigma_C^{(v_1)}, v_3)$, $(\sigma_C a_1 v_1, v_3)$, $(\sigma_C a_2 v_2, v_3)$ to G'_u with weight $\frac{1}{4}\mathrm{wt}_{G_u}(C)$.

This defines the pair (H'_u, G'_u) for all $u \in [4n] \setminus \{n+1, \ldots, 2n\}$. To define the pair for $u \in \{n+1, \ldots, 2n\}$, we simply observe that $-u \in [n]$, and so we flip the "sign" of the first vertex in each edge in H'_{-u} or G'_{-u}, and this defines (H'_u, G'_u) for $u \in \{n+1, \ldots, 2n\}$.

We now need to show that the pair (H'_u, G'_u) satisfies the normalization conditions of Definition 12.8.1, namely that $\sum_{C \in [n]^2} \mathrm{wt}_{G'_u}(C) + \sum_{C \in [n]^3} \mathrm{wt}_{H'_u}(C) \le 4$ and $\sum_{C \in [n]^3} \mathrm{wt}_{H'_u}(C) \le 1$. We note that for $u \in [4n] \setminus [2n]$, this clearly holds, so it suffices to argue this for $u \in [n]$ (which then implies the statement for $u \in [2n]$). The first inequality follows from Eq. (12.6), as the total weight of all edges is preserved. The second inequality follows because the total weight in H'_u is at most $1/4$ of the total weight in H_u, which is at most 4.

Finally, to finish the proof of Item (1), let f_u be the polynomial defined in Definition 12.8.1 from (H'_u, G'_u). We clearly have that for any $x \in C'$, $f_u(x) = \mathbb{E}[\mathrm{Dec}'(u)]$, as by construction $f_u(x)$ is equal to the left hand side of Eq. (12.8).

Proof of Item (2). We are now ready to prove Item (2). For simplicity, we will replace $4n$ with n. Let $p_{\mathcal{G}_u^{(t)}}(x) = \sum_{C \in [n]^{3t}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) \cdot x_u g_C$ and $p_{\mathcal{H}_u^{(t)}} = \sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \cdot x_u g_C$ denote the "graph-tailed" and "hypergraph-tailed" polynomials *with head* u, defined in a similar manner to the polynomials in Definition 12.8.4. We will show by induction on r that if $1 - 2(r+1)\varepsilon > 0$, then $p_{\mathcal{H}_u^{(r+1)}}(x) + \sum_{t=1}^{r+1} p_{\mathcal{G}_u^{(r+1)}}(x) \ge 1 - 2(r+1)\varepsilon$.

For the base case, we observe that when $r = 0$, $p_{\mathcal{H}_u^{(1)}}(x) = x_u \psi_u(x)$ and $p_{\mathcal{G}_u^{(1)}}(x) = x_u \phi_u(x)$ (see Definition 12.8.1). Therefore, for any $x \in C'$, $p_{\mathcal{H}_u^{(1)}}(x) + p_{\mathcal{G}_u^{(1)}}(x) = x_u f_u(x) = \mathbb{E}[x_u \mathrm{Dec}^{(x)}(u)] \ge 1 - 2\varepsilon$, as C (and therefore C') has completeness $1 - \varepsilon$. Note that we also have $|1 - x_u f_u(x)| = |x_u - f_u(x)| \le 2\varepsilon$ for all $x \in C'$.

For the induction step, we have by definition (see Remark 12.8.5) that for any $x \in C'$,

$$
\begin{aligned}
p_{\mathcal{H}_u^{(r+1)}}(x) + p_{\mathcal{G}_u^{(r+1)}(x)} - p_{\mathcal{H}_u^{(r)}}(x) &= \sum_{v \in [n]} \sum_{C \in [n]^{3r+1} : \mathrm{tail}(C) = v} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) \cdot x_u g_C \cdot (x_v f_v(x) - 1) \\
&\le \left(\sum_{v \in [n]} \sum_{C \in [n]^{3r+1} : \mathrm{tail}(C) = v} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) \cdot |x_u g_C| \cdot |1 - (x_v f_v(x))| \right) \\
&\le \left(\sum_{v \in [n]} \sum_{C \in [n]^{3r+1} : \mathrm{tail}(C) = v} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) \cdot 1 \cdot 2\varepsilon \right) \\
&\le 2\varepsilon \,.
\end{aligned}
$$

The final inequality uses the fact that $\sum_{C \in [n]^{3r+1}} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) \le 1$, which follows by induction using that $\sum_{C \in [n]^3} \mathrm{wt}_{H'_u}(C) \le 1$.

Hence, we conclude that

$$p_{\mathcal{H}_u^{(r+1)}}(x) + \sum_{t=1}^{r+1} p_{\mathcal{G}_u^{(r+1)}}(x) = \left(p_{\mathcal{H}_u^{(r+1)}}(x) + p_{\mathcal{G}_u^{(r+1)}}(x) - p_{\mathcal{H}_u^{(r)}}(x)\right) + \left(p_{\mathcal{H}_u^{(r)}}(x) + \sum_{t=1}^{r} p_{\mathcal{G}_u^{(r+1)}}(x)\right) \geq -2\varepsilon + 1 - 2r\varepsilon\,,$$

which finishes the proof of Item (2).

Proof of Item (3). The proof of smoothness is straightforward. First, if $u \in [4n] \setminus [2n]$, then the condition immediately holds by construction. Let us now consider the interesting case of $u \in [n]$. By Lemma 12.8.10, the pair (H_u, G_u) satisfies the smoothness condition. Thus, it remains to verify that the pair (H'_u, G'_u) is δ/c)-smooth for some constant c. This follows immediately because, for each edge in H'_u or G'_u that contains some $v' \in [4n]$, we can uniquely identify $v \in [n]$ such that the "original edge" in (H_u, G_u) that the new edge "comes from" contains v. Hence, if we consider the total weight of all hyperedges in (H'_u, G'_u) containing some vertex $v' \in [4n]$, there is a $v \in [n]$ such that the weight is upper bounded by the total weight of all hyperedges in (H_u, G_u) containing v. The extra constant factor c comes from the fact that the number of vertices is now $4n$ and the constant factor loss in Lemma 12.8.10.

12.8.2 Proof of Lemma 12.8.10

In this subsection, we prove Lemma 12.8.10. Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a 3-LCC with an adaptive decoder. For each $u \in [n]$, we use the decoding algorithm $\mathrm{Dec}(u)$ to define weight functions wt_{H_u} and wt_{G_u}. In what follows, we consider a fixed $u \in [n]$.

First, without loss of generality, we may assume that the decoder $\mathrm{Dec}(u)$ makes *exactly* 3 queries. We can view the decoder as a decision tree: first, $\mathrm{Dec}(u)$ generates the first query v_1 from some distribution. Then, $\mathrm{Dec}(u)$ receives a bit $a_1 \in \{-1,1\}$, the answer to the query v_1. This answer selects the branch of the decision tree, which determines the distribution of the next query v_2. Then, the decoder receives another answer $a_2 \in \{-1,1\}$, which selects the branch of the decision tree, and gives the distribution of the final query v_3. Finally, the decoder first selects some randomness $\mathbf{r} \in \{0,1\}^{\mathrm{r}}$, receives an answer a_3, and then it computes a (deterministic) function $f_{(v_1,a_1,v_2,a_2,v_3,\mathbf{r})}$ of a_3 to produce its output. This function is deterministic because the randomness is handled in $\mathbf{r}$. We note that there are exactly 4 valid deterministic functions: 1, -1, a_3, and $-a_3$, so $f_{(v_1,a_1,v_2,a_2,v_3,\mathbf{r})}$ must be one of these.

For each choice of $C = (v_1, a_1, v_2, a_2, v_3, \mathbf{r}) \in ([n] \times \{-1,1\})^2 \times [n] \times \{0,1\}^{\mathrm{r}}$, we let $\mathrm{wt}_u(C)$ be the probability that the decoder makes the set of queries C (with the appropriate answers) when given oracle access to *any* x that is consistent with C, meaning that $x_{v_1} = a_1$ and $x_{v_2} = a_2$. Indeed, this does not depend on the choice of x, as there is some probability p_{v_1} that the decoder queries v_1 (which does not depend on x), and then given $x_{v_1} = a_1$, there is a probability p_{v_2} that the decoder queries v_2, etc.

We now partition the query sets into two types. If C is such that $f_{(v_1,a_1,v_2,a_2,v_3,\mathbf{r})}$ is a constant function $\sigma \in \{-1,1\}$ (so it does not depend on a_3), then we set $\mathrm{wt}_{G_u}(v_1, a_1, v_2, a_2, \mathbf{r}) = \mathrm{wt}_u(C)$ and $\sigma_{(v_1,a_1,v_2,a_2,\mathbf{r})} = \sigma$. Otherwise, we have that C is such that $f_{(v_1,a_1,v_2,a_2,v_3,\mathbf{r})} = \sigma a_3$, and then we set $\mathrm{wt}_{H_u}(v_1, a_1, v_2, a_2, v_3, \mathbf{r}) = \mathrm{wt}_u(C)$ and $\sigma_{(v_1,a_1,v_2,a_2,v_3,\mathbf{r})} = \sigma$.

We now show that this weight function has the desired properties. Indeed, we have essentially encoded the behavior of the arbitrary decoder as this system of polynomials.

First, let us show that

$$\sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3) \right) = 4 \,.$$

Consider the decoder $\mathrm{Dec}'(u)$ that simulates Dec_u by generating random bits as the answers to the queries of $\mathrm{Dec}(u)$. It follows that the probability that $\mathrm{Dec}'(u)$ queries a particular C is $\mathrm{wt}(C)/4$, and hence Eq. (12.6) holds.

Next, let us show that for any $x \in C$

$$\sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3) \right) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = 1 \,.$$

Indeed, we observe that for any $x \in C$ and any C, $\mathrm{wt}_{H_u}(C, v_3) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$ is 0 if C is inconsistent with x, and otherwise it is the probability that $\mathrm{Dec}^x(u)$ queries C, and the same statement holds for $\mathrm{wt}_{G_u}(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$. Hence, the sum must be 1.

Finally, we have

$$x_u \sum_{C=(v_1,a_1,v_2,a_2,\mathbf{r})} \left(\mathrm{wt}_{G_u}(C)\sigma_C + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3)\sigma_{(C,v_3)} x_{v_3} \right) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = \mathbb{E}[\mathrm{Dec}^x(u) x_u] \,.$$

Indeed, this is because for any $C = (v_1, a_1, v_2, a_2, v_3, \mathbf{r})$ and any $x \in C$, then the execution of $\mathrm{Dec}^{(x)}(u)$ queries C with probability $\mathrm{wt}_{H_u}(C, v_3)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$, and then the output of the decoder is the decoding function, which is $\sigma_{(C,v_3)} x_{v_3}$. A similar statement holds for $C = (v_1, a_1, v_2, a_2, \mathbf{r})$ as well, which finishes the proof.

12.9 Refuting the graph-tail instances

In this section, we prove the first equation of Lemma 12.8.6. Let $r \geq 1$ and let $1 \leq t \leq r+1$ be fixed. We begin by defining the Kikuchi matrices.

Definition 12.9.1. Let $r \geq 1$ and $1 \leq t \leq r+1$. Let $i \in [k]$. For a tuple $C = (i, v_1, v_2, u_1, \ldots, v_{2(t-1)+1}, v_{2(t-1)+2}) \in [n]^{3t}$, we define the matrix $A_i^{(C)} \in \{0,1\}^N$ where $N = \binom{n}{\ell}^t$, to be the matrix indexed by tuples of sets $\vec{S} = (S_0, \ldots, S_{t-1})$, where $A_i^{(C)}((S_0, \ldots, S_{t-1}), (T_0, \ldots, T_{t-1})) = 1$ if for all $h = 0, \ldots, t-1$, $S_h \oplus T_h = \{v_{2h+1}, v_{2h+2}\}$ with $v_{2h+1} \in S_h, v_{2h+2} \in T_h$. If this does not hold, then the entry of the matrix is 0.

We let $A_i = \frac{1}{D_t} \sum_{C \in [n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) A_i^{(C)}$ and $A = \sum_{i=1}^k b_i A_i$. Here, $D_t = \binom{n-2}{\ell-1}^t$.

Next, we relate $\Phi^{(t)}(x)$ to a quadratic form on the matrix A.

Lemma 12.9.2. *Let $x \in \{-1,1\}^n$, and let $x' \in \{-1,1\}^N$, where $N = \binom{n}{\ell}^t$, denote the vector where the $(S_0, S_1, \ldots, S_{t-1})$-th entry of x' is $\prod_{h=0}^{t-1} x_{S_h}$. Let $i \in [k]$ and $t \in \{0, \ldots, r\}$. Then, for any $C = (i, v_1, v_2, u_1, v_3, v_4, \ldots, v_{2(t-1)+1}, v_{2(t-1)+2}) \in [n]^{3t}$, it holds that*

$$x'^{\top} A_i^{(C)} x' = D_t \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}} \,,$$

i.e., the product of the monomials associated to C, where $D_t = \binom{n-2}{\ell-1}^t$. Moreover, for any matrix $B_i^{(C)}$ obtained by "zeroing out" exactly αD_t entries of $A_i^{(C)}$, the equality holds with a factor of $1-\alpha$ on the right.

In particular, $x'^\top A x' = \Phi^{(t)}(x)$.

Proof. Let $\vec{S} = (S_0, S_1, \ldots, S_{t-1})$ and $\vec{T} = (T_0, \ldots, T_{t-1})$ be such that $A_i^{(C)}(\vec{S}, \vec{T}) = 1$. Then, we have that

$$x'_{\vec{S}} x'_{\vec{T}} = \prod_{h=0}^{t-1} x_{S_h} x_{T_h} = \prod_{h=0}^{t-1} x_{S_h \oplus T_h} = \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}} \ ,$$

which is equal to the product of monomials on the right-hand side of the equation we wish to show.

It thus remains to argue that $A_i^{(C)}$ has exactly D_t nonzero entries. We observe that, for each $h = 0, \ldots, t-1$, there are exactly $\binom{n-2}{\ell-1}$ pairs (S_h, T_h) such that $S_h \oplus T_h = C_h$ with $v_{2h+1} \in S_h$ and $v_{2h+2} \in T_h$. Indeed, this is because by Definition 3.2.1, these vertices must be distinct, and then we must simply choose a set of size $\ell - 1$ that does not contain either of v_{2h+1} and v_{2h+2} and this determines S_h and T_h. Thus, $D_t = \binom{n-2}{\ell-1}^t$, as required. □

We would like to now apply matrix Khintchine (Fact 3.4.2) to bound $\mathbb{E}_b[\|A\|_2]$ and thus bound $\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)}(x))]$. However, to do this, we need good bounds on the $\|A_i\|_2$ of the individual matrices A_i. It turns out that the bounds we require for this approach to work are false, but one can find a submatrix B_i of A_i such that the bounds hold. To argue this, we will need the following first moment bounds.

Lemma 12.9.3 (First and conditional moment bounds)**.** *Fix $r \geq 1$, $1 \leq t \leq r+1$, and $i \in [k]$. Let A_i be the Kikuchi matrix defined in Definition 12.9.1.*

Let $\vec{S} = (S_0, \ldots, S_{t-1}) \in \binom{[n]}{\ell}^t$ be a row of the matrix, and let $\deg_i(\vec{S})$ denote the ℓ_1-norm of the $\vec{S}$-th row of A_i. Then,

$$\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})] \leq \frac{4}{N} \, ,$$

where $N = \binom{n}{\ell}^t$.

Furthermore, let $C \in [n]^{3t}$ be a chain with head i. Let $\mathcal{D}_C$ denote the uniform distribution over rows of $A_i^{(C)}$ that contain a nonzero entry. Then, it holds that

$$\mathbb{E}_{\vec{S} \sim \mathcal{D}_C}[\deg_i(\vec{S})] \leq \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{16}{N} \, .$$

Finally, the same bounds hold for the columns of the matrix.

With Lemma 12.9.3, we can now do the following. Let Γ be a sufficiently large constant, let $\mathcal{B}_1 = \{\vec{S} : \deg_i(\vec{S}) \geq \Gamma/N\}$ be the set of rows with ℓ_1-norm at least Γ/N, and similarly let $\mathcal{B}_2$ be defined for the columns. We observe that by the conditional moment bounds in Lemma 12.9.3 and Markov's inequality, each $A_i^{(C)}$ has at least $1 - O(1/\Gamma)$-fraction of its nonzero rows not in $\mathcal{B}_1$, and similarly for columns and $\mathcal{B}_2$. It thus follows that after setting all the rows in $\mathcal{B}_1$ and columns in $\mathcal{B}_2$ to 0, the resulting matrix still has at least $1 - O(1/\Gamma)$-fraction of its original nonzero entries. By taking Γ large enough, we can ensure that this fraction is at least $1/2$. Now, we let $B_i^{(C)}$

be the matrix where we have deleted all rows in $\mathcal{B}_1$ and columns in $\mathcal{B}_2$ from $A_i^{(C)}$, and we have additionally set more entries to 0 so that $B_i^{(C)}$ has *exactly* $D_t/2$ nonzero entries, where t is such that $C \in [n]^{3t}$.

Let us define: $B_i = \frac{1}{D_t} \sum_{C \in [n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) B_i^{(C)}$ and $B = \sum_{i=1}^{k} b_i B_i$. By Lemma 12.9.2 (and the "moreover" part), we have that for every $x \in \{-1,1\}^n$, there exists $x' \in \{-1,1\}^N$ such that $x'^\top B x' = \frac{1}{2}\Phi^{(t)}(x)$. By construction, we have that $\|B_i\|_2 \le \Gamma/N$, as this is an upper bound on the ℓ_1-norm of any row/column in B_i.

Thus, applying matrix Khintchine (Fact 3.4.2), we obtain

$$\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)})] \le \mathbb{E}_b[N\|B\|_2] \le N \cdot \frac{\Gamma}{N} O(\sqrt{k \log N}) = O(\sqrt{k\ell r \log n}),$$

where we use that Γ is constant. This finishes the proof of the first equation in Lemma 12.8.6, up to the proof of Lemma 12.9.3.

Proof of Lemma 12.9.3. We will only prove the statement for the rows. One can observe from the proof that it will immediately hold for the columns also.

We begin by estimating the first moment, i.e., $\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})]$. By definition, we have that

$$\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})] = \frac{1}{N}\frac{1}{D_t} \sum_{C \in [n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \cdot D_t \le \frac{4}{N},$$

as the sum of the weights of all chains is at most 4 by Observation 12.8.8.

We now fix $t \in \{1, \dots, r+1\}$, $C \in [n]^{3t}$ with head i. Let $\mathcal{D}_C$ denote the uniform distribution over rows of $A_i^{(C)}$ that contain a nonzero entry. We compute the conditional expectation as follows. First, we shall bound, for $C' \in [n]^{3t}$ with head i, the number of rows $\vec{S}$ such that $A_i^{(C)}$ and $A_i^{(C')}$ both have a nonzero entry in the $\vec{S}$-th row, *normalized* by the scaling factor $1/D_t$. This quantity will depend on some parameter z, which is the number of "shared vertices" between C and C'. Then, we will bound, for each z, the total weight of all $C' \in [n]^{3t}$ that has at least z "shared vertices" with C.

Step 1: bounding the normalized number of entries for a fixed C'. To begin, we define the number of "shared vertices" between two pairs of chains C and C'.

Definition 12.9.4 (Left vertices)**.** Let $C \in [n]^{3t}$. The tuple of *left vertices* of C is the sequence $L(C) = (v_1, v_3, v_5, \dots, v_{2(t-1)+1})$. We note that if $\vec{S}$ is a row such that $A_i^{(C)}$ has nonzero entry in the $\vec{S}$-th row, then $v_{2h+1} \in S_h$ for $h = 0, \dots, t-1$.

Definition 12.9.5 (Intersection patterns)**.** Let $C \in [n]^{3t}$ and $C' \in [n]^{3t}$.

The *intersection pattern* of C with C', given by $Z \in \{0,1\}^t$, is defined as $Z_h = 1$ if $L(C)_h = L(C')_h$, and it is 0 otherwise.

We now fix $C' \in [n]^{3t}$ and count the number of rows as a function of the intersection pattern Z. We observe that in order for a row $\vec{S}$ to have a nonzero entry for both pairs of chains, we must have $\{L(C)_h, L(C')_h\} \subseteq S_{h-1}$ for all $h = 1, \dots, t$.

We observe that for each intersection point, i.e., an h such that $L(C)_h = L(C')_h$, there are $\binom{n}{\ell-1}$ choices for the corresponding set, as it needs to only contain one vertex. For each nonintersection

point, i.e., an $h \in \{1, \dots, t\}$ where $L(C)_h \neq L(C')_h$, we have $\binom{n}{\ell-2}$ choices, because the set needs to contain both vertices. In total, we have $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}$.

Now, this implies an upper bound of $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}/D_t$ on the normalized number of entries, which we can compute as

$$\begin{aligned}
&\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}/D_t = \frac{\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}}{\binom{n-2}{\ell-1}^t} = 2\left(\frac{\binom{n}{\ell-2}}{\binom{n}{\ell-1}}\right)^{t-z} \cdot \left(\frac{\binom{n}{\ell-1}}{\binom{n-2}{\ell-1}}\right)^t \\
&\leq \left(\frac{\ell-1}{n-\ell+2}\right)^{t-z} \cdot \left(\frac{n(n-1)}{(n-\ell+1)(n-\ell)}\right)^t \leq \left(\frac{\ell}{n}\right)^{t-z} \cdot \left(1 + \frac{O(\ell r)}{n}\right).
\end{aligned}$$

Step 2: bounding the weight of C' with a fixed intersection pattern Z. Let us fix the intersection pattern Z. We observe that this determines a set of $|Z|$ vertices that must be contained in C'. We will abuse notation and let $Z \in [n] \cup \{\star\}^t$ denote this sequence of vertices (with $\star$'s for the unfixed entries). Let t'' denote the largest $h \in \{1, \dots, t\}$ for which $Z_{t''} \neq \star$. We then have

$$\begin{aligned}
&\sum_{C' \in [n]^{3t} : Z \subseteq C} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\sum_{C' \in [n]^{3(t-t'')}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C'', C') \right) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\sum_{(u, C') \in [n]^{3(t-t'')}} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \right) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \sum_{C' \in [n]^{3(t-t'')-1}} \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \right) \\
&\leq 4 \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \right).
\end{aligned}$$

Above, we use that $\sum_{C' \in [n]^{3(t-t'')-1}} \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \leq 4$, which follows by Observation 12.8.8.

We now clearly have that $\sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \right) \leq 4(\delta n)^{-|Z|}$. This follows by δ-smoothness, as when we sum over a link with no fixed vertex, it has weight 1 (unless it is the last link, where it has weight 4), and when we sum over a link where $Z_h \neq \star$, by δ-smoothness it must have weight at most $1/\delta n$. We thus have a bound of $4(\delta n)^{-|Z|}$.

Putting it all together. By combining steps (1) and (2) (and paying an additional $\binom{t}{z}$ factor to

choose the nonzero entries of Z), we thus obtain the final bound of

$$\begin{aligned}
\mathbb{E}_{\vec{S}\sim\mathcal{D}_C}[\deg_i(\vec{S})] &\le \frac{4}{D_t}\sum_{z=0}^{t}\binom{t}{z}\cdot 2\left(1+\frac{O(\ell r)}{n}\right)\cdot\left(\frac{\ell}{n}\right)^{t-z}\cdot(\delta n)^{-z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{8}{D_t}\left(\frac{\ell}{n}\right)^{t}\cdot\sum_{z=0}^{t}\cdot\left(\frac{t}{\delta\ell}\right)^{z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{8}{D_t}\left(\frac{\ell}{n}\right)^{t}\cdot\sum_{z=0}^{r}\cdot\left(\frac{r}{\delta\ell}\right)^{z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{16}{D_t}\left(\frac{\ell}{n}\right)^{t},
\end{aligned}$$

where we use that $\ell \ge 2r/\delta$.

Finally, we need to compute D_t/N. We have

$$\begin{aligned}
\frac{D_t}{N} &= \frac{\binom{n-2}{\ell-1}^t\cdot\binom{n}{\ell}^{r+1-t}}{\binom{n}{\ell}^{r+1}} = \left(\frac{\binom{n-2}{\ell-1}}{\binom{n}{\ell}}\right)^t \\
\left(\frac{\ell(n-\ell)}{n(n-1)}\right)^t &\ge \left(\frac{\ell}{n}\right)^t\left(1-\frac{O(\ell r)}{n}\right).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\mathbb{E}_{\vec{S}\sim\mathcal{D}_C}[\deg_i(\vec{S})] &\le \left(1+\frac{O(\ell r)}{n}\right)\frac{16}{D_t}\left(\frac{\ell}{n}\right)^{t} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{16}{N},
\end{aligned}$$

which finishes the proof. □

12.10 Linear 3-LCC lower bounds over larger fields

In this section, we prove Theorem 8 in the case where the finite field $\mathbb{F}$ is not $\mathbb{F}_2$. The proof will be nearly identical to the proof in Sections 12.4 to 12.7 for the case of $\mathbb{F} = \mathbb{F}_2$, and so we shall only give a proof sketch and mainly focus on the parts of the proof where modifications are required.

To begin, we recall that by Definition 3.3.9, there exist 3-uniform hypergraph matchings $H_1, \dots, H_n$, each of size at least δn, such that for each $u \in [n]$ and $C = \{v_1, v_2, v_3\} \in H_u$, there exists $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{F}\setminus\{0\}$ such that for every $x \in \mathcal{L}$, it holds that $\alpha_1 x_{v_1} + \alpha_2 x_{v_2} + \alpha_3 x_{v_3} = x_u$. Furthermore, without loss of generality we can assume that the code is systematic, i.e., for any $b \in \mathbb{F}^k$, $x = \mathcal{L}(b)$ satisfies $x_i = b_i$ for all $i \in [k]$.

Next, let us define a code $\mathcal{L}' \colon \{-1,1\}^k \to \{-1,1\}^{n(|\mathbb{F}|-1)}$ where, for each $u \in [n]$ and $\alpha \in \mathbb{F}\setminus\{0\}$, we set $\mathcal{L}'(b)_{(u,\alpha)} = \alpha\mathcal{L}(b)_u$. Let $n' = n(|\mathbb{F}|-1)$, and associate $[n']$ with the set $[n]\times(\mathbb{F}\setminus\{0\})$. We now observe that $\mathcal{L}'$ is a 3-LCC in normal form with the additional property that the coefficients of all constraints can be taken to be 1 without loss of generality. Formally, there exist 3-uniform

hypergraph matchings $H_1, \dots, H_{n'}$ such that (1) each H_u has $|H_u| \geq \delta n'/(|\mathbb{F}|-1)$, and (2) for each $u \in [n']$ and each $C = \{v_1, v_2, v_3\} \in H_{(u,\alpha)}$, every $x \in \mathcal{L}$ satisfies $x_u = x_{v_1} + x_{v_2} + x_{v_3}$.

Moreover, there is now a group action of $(\mathbb{F} \setminus \{0\}, \times)$ on the elements of $[n']$, namely for any $\alpha \in \mathbb{F} \setminus \{0\}$, this action maps $u \mapsto \alpha u$. We note that this action respects the constraints. Namely, for $C = \{v_1, v_2, v_3\} \in \binom{[n']}{3}$, if we define $\alpha C = \{\alpha v_1, \alpha v_2, \alpha v_3\}$, then we have that $H_{\alpha u} = \alpha H_u = \{\alpha C : C \in H_u\}$. For the proof, we will be using the fact that there is a negation action for $\alpha = -1$; this is because this transformation has made all coefficients in the constraints be equal to 1, so to cancel a variable x_u we shall only need x_{-u}.

We shall now abuse notation and redefine n' to be n, and we now simply assume that we have this group action on $[n]$. We have thus added this additional property to the code, and in doing so we have only decreased δ by a factor of $|\mathbb{F}| - 1$.

We now turn to the main part of the proof. Following Section 12.4, we define t-chains. The definition of t-chains now requires a small modification because in the original definition we formed longer chains by canceling a variable x_w via the operation $x_w + x_w = 0$, which was specific to the field $\mathbb{F}_2$. Now, we use the negation action on $[n]$ to cancel a variable.

Definition 12.10.1 (t-chain hypergraph $\mathcal{H}_u^{(t)}$)**.** Let $t \geq 1$ be an integer. For any $u \in [n]$, let $\mathcal{H}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{H}_u^{(t)}} \colon [n]^{3t+1} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t+1$ tuples of the form $C = (u_0, v_1, v_2, u_1, \dots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2}, u_t)$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \prod_{h=0}^{t-1} \mathrm{wt}_{H_{-u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1}) \,.$$

For a t-chain C, we call u_0 the head, the u_h's the *pivots* for $1 \leq h \leq t-1$, and u_t the *tail* of the chain C. The monomial associated to C, which we denote by g_C, is defined to be $x_{u_t} \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$.

Given any t-chain $C = (u_0, v_1, v_2, u_1, \dots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2}, u_t)$, we let the negation of the chain, denoted by $-C$, be the chain $(-u_0, -v_1, -v_2, -u_1, \dots, -u_{t-1}, -v_{2(t-1)+1}, -v_{2(t-1)+2}, -u_t)$.

As before, we note that the linear equation defined by a t-chain or its negation is satisfied by any $x \in \mathcal{L}$.

In Section 12.4, we defined an instance polynomial Φ_b related to the system of linear constraints. This was natural over $\mathbb{F}_2$ as there is a group isomorphism between $(\mathbb{F}_2, +)$ and $\{-1, 1\} \in (\mathbb{R}, \times)$. Here, we can make a similar definition by using a nontrivial group homomorphism π from $(\mathbb{F}, +)$ to $(\mathbb{C}, \times)$ where the image of π is contained in the unit circle $\{z \in \mathbb{C} : |z| = 1\}$. However, the instance polynomial Φ_b (and the "decomposed polynomials" $\Psi_{i,Q}$ defined later) were only formally needed to discuss sets of linear constraints that are satisfied by the subspace $\mathcal{L}$. Thus, to avoid using the group homomorphism π, here we shall simply use these polynomials to refer to the underlying sets of constraints.

We now perform the hypergraph decomposition step as in Section 12.5, which is unchanged (once we use the updated definition of chain).[15] This produces the subinstances $\Psi^{(t)}(x, y)$, as before.

To finish the proof, we follow Section 12.6.5 in Section 12.6. A near-identical calculation

[15] We note that the naive application of the decomposition step will produce partitions $\mathcal{H}^{(r,Q)}$ where $\mathcal{H}^{(r,-Q)}$ is not necessarily equal to $-\mathcal{H}^{(r,Q)}$. This turns out to not matter in the proof; as it turns out, we merely need that both decompositions $\cup_Q -\mathcal{H}^{(r,Q)}$ and $\cup_Q \mathcal{H}^{(r,Q)}$ are both smooth partitions of $\mathcal{H}^{(r)}$, which obviously holds. Nonetheless, we note that one could also easily modify the decomposition step to respect this negation action.

as before shows that $\mathcal{L}''$ is a $(2, \delta)$-LDC for $\delta' = \Omega(\delta/r)$ provided that $\delta^2 k \leq O(\log^2 n)$. We then apply Fact 3.3.4, and conclude that $k \leq O(\log^4 n/\delta^2)$. In either case, we thus have that $k \leq O(\log^4 n/\delta^2)$.

Now, we recall that we had redefined n to be $n(|\mathbb{F}|-1)$ and δ to be $\delta/(|\mathbb{F}|-1)$. Thus, we have that for the original code, $\frac{k\delta^2}{(|\mathbb{F}|-1)^2} \leq O(\log^4 n)$ provided that $|\mathbb{F}| \leq n$. Note that if $|\mathbb{F}| \geq k$, then Theorem 8 becomes trivial, and so we can assume that $|\mathbb{F}| \leq k \leq n$ (as we always have $k \leq n$). This finishes the proof of Theorem 8 for larger fields.

12.11 Design 3-LCCs over $\mathbb{F}_2$ from Reed–Muller codes

In this section, we give a simple folklore construction of a design 3-LCCs (Definition 3.3.11) using Reed–Muller codes.

Lemma 12.11.1 (Design 3-LCCs over $\mathbb{F}_2$ from Reed–Muller Codes)**.** *Let t be an integer, and let $k = 1 + t + \binom{t}{2}$. Then, there is a design 3-LCC with blocklength $n = 4^t$ of dimension k. In particular, $n \leq 2^{2\sqrt{2k}}$.*

To prove this lemma, we will need the following fact about polynomials over $\mathbb{F}_4$.

Fact 12.11.2. *Let $f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$ be a degree-2 polynomial over $\mathbb{F}_4$. Then, $\sum_{\beta \in \mathbb{F}_4} f(\beta) = 0$.*

Proof. Recall that the field $\mathbb{F}_4$ is equivalent to the polynomial ring $\mathbb{F}_2[\beta]$ modulo the equation $\beta^2 + \beta + 1 = 0$. We have

$$\begin{aligned}
&f(0) = \alpha_0 \\
&f(1) = \alpha_0 + \alpha_1 + \alpha_2 \\
&f(\beta) = \alpha_0 + \alpha_1\beta + \alpha_2\beta^2 \\
&f(1+\beta) = \alpha_0 + \alpha_1(1+\beta) + \alpha_2(1+\beta)^2 \\
&\quad \implies f(0) + f(1) + f(\beta) + f(1+\beta) = \alpha_0 \cdot 4 + \alpha_1 \cdot 2(1+\beta) + \alpha_2(1 + \beta^2 + (1 + 2\beta + \beta^2)) \\
&= 0\,,
\end{aligned}$$

as $2 = 0$ in $\mathbb{F}_4$. □

Proof of Lemma 12.11.1. We will define the code in two stages. First, we will define, via an encoding map, a code over $\mathbb{F}_4$ with the desired dimension argue that it is a design 3-LCC. Then, we will use this code to construct a code over $\mathbb{F}_2$.

Let $\mathcal{V}$ denote the vector space of degree ≤ 2 polynomials over $\mathbb{F}_4$ in t variables $x_1, \ldots, x_t$. We note that $\mathcal{V}$ has dimension k.

For each $b \in \mathbb{F}_4^k$, we encode b by (1) letting $f_b(x_1, \ldots, x_t)$ be the degree-2 polynomial with coefficients given by b, and (2) evaluating f_b over all $x \in \mathbb{F}_4^t$; this yields an output $Z \in \mathbb{F}_4{}^{4^t} = \mathbb{F}_4^n$, which is the encoding $\mathrm{Enc}(b)$. We note that Enc is clearly an $\mathbb{F}_4$ linear map.

We now argue that this encoding map is a design 3-LCC. Indeed, we need to define a system of constraints such that for every pair $x^{(0)}, x^{(1)} \in \mathbb{F}_4^t$, there is a unique constraint containing $x^{(0)}, x^{(1)}$. Let $x^{(\beta)} = x^{(0)} + \beta(x^{(1)} - x^{(0)})$ and $x^{(1+\beta)} = x^{(0)} + (1+\beta)(x^{(1)} - x^{(0)})$. We note that $x^{(0)}, x^{(1)}, x^{(\beta)}$ and $x^{(1+\beta)}$ is the line $L(t) = x^{(0)} + \lambda(x^{(1)} - x^{(0)})$ containing $x^{(0)}, x^{(1)}$. Fix $b \in \mathbb{F}_4^k$, and let f_b be the corresponding polynomial. We know that $g(\lambda) = f_b(L(\lambda))$ is a degree-2 univariate polynomial in λ. Hence, by Fact 12.11.2, it follows that $f_b(x^{(0)}) + f_b(x^{(1)}) + f_b(x^{(\beta)}) + f_b(x^{(1+\beta)}) = 0$. Hence, for

each pair $x^{(0)}, x^{(1)} \in \mathbb{F}_4^t$, there exists a constraint containing this pair, and moreover, because two points determine a line, any constraint containing this pair must be exactly this line. Thus, the code given by Enc is a design 3-LCC.

We now use the above code to construct a binary code. Let $\mathrm{Tr}\colon \mathbb{F}_4 \to \mathbb{F}_2$ be the trace map. We let $\mathcal{V}'$ be the image of $\mathcal{V}$ under Tr (applied element-wise to each vector in $\mathcal{V}$). We note that because $\mathcal{V}$ has dimension k over $\mathbb{F}_4$ is a linear code, it is systematic, meaning that there is a subset $S \subseteq \mathbb{F}_4^t$ such that $\mathcal{V}|_S = \mathbb{F}_4^k$. Therefore, because the trace map is identity on $\mathbb{F}_2$, it follows that $\mathcal{V}'|_S = \mathbb{F}_2^k$, i.e., that $\mathcal{V}'$ has dimension k also.

To finish the proof, we need to argue that $\mathcal{V}'$ is a design 3-LCC. Let $g \in \mathcal{V}'$. We will show that for each line $x^{(0)}, x^{(1)}, x^{(\beta)}, x^{(1+\beta)}$ in $\mathbb{F}_4^t$ as defined earlier, it holds that $g(x^{(0)}) + g(x^{(1)}) + g(x^{(\beta)}) + g(x^{(1+\beta)}) = 0$. Indeed, we have that $g = \mathrm{Tr}(f)$ for some $f \in \mathcal{V}$, and that $f(x^{(0)}) + f(x^{(1)}) + f(x^{(\beta)}) + f(x^{(1+\beta)}) = 0$. Because all the coefficients in the linear constraint are 1, i.e., they are in $\mathbb{F}_2$, the constraint still holds after applying $\mathrm{Tr}(\cdot)$, as this is an $\mathbb{F}_2$-linear map. Thus, the constraint holds, which finishes the proof. □

Part IV

Future Directions

Chapter 13

Kikuchi Matrices over Larger Alphabets

In the majority of this thesis, we have considered problems over $\mathbb{F}_2$ or binary alphabets. In this section, we will sketch how to extend our methods (specifically, the basic approach outlined in Section 2.1) to the case of arbitrary prime finite fields $\mathbb{F}_p$; we remark that this likely extends to arbitrary finite fields $\mathbb{F}_q$, rings $\mathbb{Z}_t$ where t is composite, and more generally any Abelian group G. It is likely not too difficult to prove generalizations of the main results of this thesis to larger alphabets/fields using the ideas in this chapter, though we will not formally do so in this thesis.

As an example, let us consider task of refuting a random k-XOR instance Ψ in n variables over the finite field $\mathbb{F}_p$, where p is prime and k is even. We represent Ψ, the random system of k-sparse linear equations over $\mathbb{F}_p$, as a set $H \subseteq \mathbb{F}_p^n$ where each $z \in H$ is k-sparse, i.e., has exactly k nonzero entries, along with "right-hand sides" $b_z \in \mathbb{F}_p$ for each $z \in H$. The equations are then given by $\sum_{i=1}^n x_i z_i = b_z$ for all $z \in H$, where we note that the left-hand side of the equation is k-sparse because z contains k nonzero coordinates.

In this section, we will sketch the proof of the following theorem.

Theorem 13.0.1 (Theorem 2.0.2 for random k-XOR over larger fields)**.** *Let k be even. For every integer $\ell \geq k/2$, there is an algorithm $\mathcal{A}$ that takes as input a k-XOR instance Ψ in n variables $x_1, \ldots, x_n$ over the field $\mathbb{F}_p$, specified by a set of k-sparse vectors $H \subseteq \mathbb{F}_p^n$ of $m = |H|$ vectors, along with "right-hand sides" $b_z \in \mathbb{F}_p$ for each $z \in H$. The algorithm $\mathcal{A}$ outputs in $((p-1)n)^{O(\ell)}$-time a value* alg-val$(\Psi) \in [0, m]$ *with the following two properties:*

(1) val$(\Psi) \leq$ alg-val(Ψ) *for all k-XOR instances Ψ over $\mathbb{F}_p$;*

(2) If $m \geq O\left(\frac{1}{\varepsilon^2}\left(\frac{n(p-1)}{\ell}\right)^{\frac{k}{2}-1} n \log n\right)$ and the input Ψ is a random k-XOR instance, i.e., H is a random collection of m k-sparse vectors z and each b_z is chosen from $\mathbb{F}_p$ uniformly at random, then with high probability over the draw of H and the b_z's, it holds that alg-val$(\Psi) \leq m(\frac{1}{p} + \varepsilon)$.

Above, val(Ψ) denotes the maximum number of constraints that one can simultaneously satisfy with a single assignment $x \in \mathbb{F}_p^n$.

Proof sketch of Theorem 13.0.1. We follow the basic approach outlined in Section 2.1. As a first step, we need to encode the instance Ψ as a polynomial. We do this by embedding $\mathbb{F}_p$ into $\mathbb{C}$ via the map $\alpha \in \mathbb{F}_p \mapsto \omega^\alpha$, where $\omega = e^{2\pi i/p} \in \mathbb{C}$ is a primitive p-th root of unity.

Namely, let $\Omega = \{\omega^\alpha : \alpha \in \mathbb{F}_p\}$ be the set of p-th roots of unity, and let Ω^n denote the "$\mathbb{F}_p$ hypercube" in $\mathbb{C}^n$. For an assignment $x \in \mathbb{F}_p^n$, let $y \in \mathbb{C}^n$ be given by $y_i = \omega^{x_i}$. We can then embed

a k-XOR constraint $\sum_{i=1}^n x_i z_i = b_z$ into $\mathbb{C}$ via the polynomial $\sum_{\alpha \in \mathbb{F}_p \setminus \{0\}} \left(\omega^{-b_z} \prod_{i=1}^n y_i^{z_i}\right)^\alpha$; here, we think of $z_i \in \mathbb{F}_p$ and $b_z \in \mathbb{F}_p$ as integers in $\{0, \dots, p-1\}$, and we note that this is well-defined since $\omega^p = 1$. Notice that this polynomial is $p-1$ if $\sum_{i=1}^n x_i z_i = b_z$, i.e., the constraint is satisfied, and otherwise it is -1. Indeed, we have that $y_i^{z_i} = \omega^{x_i z_i}$, so that $\omega^{-b_z} \prod_{i=1}^n y_i^{z_i} = \omega^{-b_z + \sum_{i=1}^n x_i z_i}$. Therefore, if the constraint is unsatisfied, then the polynomial is $\sum_{\alpha \neq 0} \omega^{\alpha\beta}$ for some $\beta \neq 0$, and so it is -1 (as for any $\beta \neq 0$, $\sum_{\alpha \in \mathbb{F}_p} \omega^{\alpha\beta} = 0$), and if the constraint is satisfied, then this sum is simply $p-1$.

Thus, we let $f(y) \coloneqq \sum_{z \in H} \sum_{\alpha \neq 0} \left(\omega^{-b_z} \prod_{i=1}^n y_i^{z_i}\right)^\alpha = \sum_{z \in H} \sum_{\alpha \neq 0} \omega^{-\alpha b_z} \prod_{i=1}^n y_i^{\alpha z_i}$, and we have argued that $\max_{y \in \Omega^n} f(y) = q \operatorname{val}(\Psi) - m$. Hence, we define $\operatorname{val}(f) \coloneqq \max_{y \in \Omega^n} f(y)$, and we have that $\operatorname{val}(\Psi) = \frac{1}{q}(\operatorname{val}(f) + m)$. It thus remains to give an algorithm to upper bound $\operatorname{val}(f)$, which we do by defining Kikuchi matrices over larger fields.

Definition 13.0.2 (Definition 2.1.1 for larger fields)**.** Let p be a prime. Let $z \in \mathbb{F}_p^n$ be a k-sparse vector where k is even, and let $\ell \geq k/2$ be an integer. We define the matrix $A_z \in \mathbb{C}^{N \times N}$ as follows. Let $N = \binom{n}{\ell}(p-1)^\ell$ and identify N with ℓ-sparse vectors $u \in \mathbb{F}_p^n$. We let $A_z(u, v) = 1$ if (1) $v - u = z$, and (2) $\operatorname{supp}(u) \cap \operatorname{supp}(z) = k/2$, $\operatorname{supp}(v) \cap \operatorname{supp}(z) = k/2$, and $\operatorname{supp}(u) \cap \operatorname{supp}(z) \cap \operatorname{supp}(v) = \emptyset$. Otherwise, we set $A_z(u, v) = 0$. Here, $\operatorname{supp}(u)$ is the support of u, i.e., the set $\operatorname{supp}(u) \coloneqq \{i \in [n] : u_i \neq 0\}$.

Proposition 13.0.3 (Proposition 2.1.2 for larger fields)**.** *Let $z \in \mathbb{F}_p^n$ be a k-sparse vector where k is even. Let $\ell \geq k/2$ be an integer, and let A_z be defined as in Definition 13.0.2. Then, the following hold:*

1. *A_z has at most one nonzero entry per row or column, and has* exactly *D nonzero entries, where $D = \binom{k}{k/2}\binom{n-k}{\ell-k/2}(p-1)^{\ell-k/2}$;*
2. *For any $y \in \Omega^n$, let $y^{\odot \ell} \in \Omega^N$ be the vector where the u-th entry is $y_u^{\odot \ell} = \prod_{i=1}^n y_i^{u_i}$. Then, $(y^{\odot \ell})^\dagger A_z y^{\odot \ell} = D \prod_{i=1}^n y_i^{z_i}$.*

Remark 13.0.4. We remark that one could define A_z in Definition 13.0.2 without condition (2), i.e., the restrictions on the support. This makes the quantity D in Proposition 13.0.3 more complicated (but only affects lower order terms) and also introduces some additional technical difficulties in the analysis.

Proof. The fact that A_z has at most one nonzero entry per row or column follows because for a fixed z and a fixed choice of the row u, there is at most one v such that $v - u = z$. One can compute the number of nonzero entries by counting the number of pairs (u, v) satisfying the two conditions. Indeed, the two conditions imply $A_z(u, v) = 1$ if and only if we can write $z = z_1 + z_2$ where z_1, z_2 are $k/2$-sparse vectors with disjoint support and $u = z_1 + w$, $v = z_2 + w$, where w is a $(\ell - k/2)$-sparse vector with $\operatorname{supp}(w) \cap \operatorname{supp}(z) = \emptyset$. There are $\binom{k}{k/2}$ ways to split z into z_1, z_2, and after that there are $\binom{n-k}{\ell-k/2}(p-1)^{\ell-k/2}$ ways to choose w. So, $D = \binom{k}{k/2}\binom{n-k}{\ell-k/2}(p-1)^{\ell-k/2}$. Notice that here we crucially need that k is even so that we can divide z into two halves of equal sparsity.

To prove Item (2), we observe that for any $y \in \Omega^n$,

$$(y^{\odot \ell})^\dagger A_z y^{\odot \ell} = \sum_{(u,v): A_z(u,v)=1} \prod_{i=1}^n y_i^{-u_i} y_i^{v_i} = \sum_{(u,v): A_z(u,v)=1} \prod_{i=1}^n y_i^{z_i} = D \prod_{i=1}^n y_i^{z_i}.$$

Note that in the above, we raise y_i to an element of $\mathbb{F}_p$, and such operations are well-defined since $y_i \in \Omega$ for each $i \in [n]$. □

We thus define $A = \sum_{z\in H}\sum_{\alpha\in\mathbb{F}_p\setminus\{0\}}\omega^{-\alpha b_z}A_{\alpha z}$. By Proposition 13.0.3, it remains to bound $\max_{y\in\Omega^N} y^\dagger Ay$, which we do using, e.g., the Matrix Bernstein inequality (Fact 3.4.1). To do this, we observe that for a fixed hypergraph H, $A = \sum_{z\in H} B_z$, where $B_z = \sum_{\alpha\neq 0}\omega^{-\alpha b_z}A_{\alpha z}$ is a mean 0 random matrix. Because each A_z has at most one nonzero entry per row/column and $|b_z| = 1$, we have that $\|A_z\|_2 \le 1$ holds, and so $\|B_z\|_2 \le p-1$ holds. In fact, we can show that $\|B_z\| \le 1$, crucially saving a factor of $p-1$, as follows. We observe that for any row u (column v) and any fixed z, there is at most one column v (row u) such that $v - u = \alpha z$ for some $\alpha \neq 0$. That is, if the row u is such that $v - u = \alpha z$, then there cannot exist a column v' such that $v' - u = \alpha' z$ where $\alpha' \neq \alpha$. This follows from the observation that $\mathrm{supp}(u)$ contains exactly half of $\mathrm{supp}(\alpha z) = \mathrm{supp}(z)$ for some $\alpha \neq 0$, and so $\mathrm{supp}(u)\cap\mathrm{supp}(\alpha z) = \mathrm{supp}(u)\cap\mathrm{supp}(\alpha' z)$ for all $\alpha' \neq 0$. Hence, given u and z, we can determine the unique choice of α (if one exists) by reading the coefficients of u on the variables in $\mathrm{supp}(u)\cap\mathrm{supp}(z)$. We note that this is the main reason we have the condition on the support in Definition 13.0.2 (see Remark 13.0.4).

By a similar calculation as done in Section 2.1, we observe that $\mathbb{E}[AA^\dagger] = \sum_{z\in H}\sum_{\alpha\neq 0} A_{\alpha z}A_{\alpha z}^\dagger$, and that $\mathbb{E}[A^\dagger A] = \sum_{z\in H}\sum_{\alpha\neq 0} A_{\alpha z}^\dagger A_{\alpha z}$. These matrices are equal, as $A_{\alpha z}^\dagger = A_{-\alpha z}$, and both these matrices are diagonal. Let $\Upsilon = \mathbb{E}[AA^\dagger] = \mathbb{E}[A^\dagger A]$. The u-th diagonal entry Υ_u is the total number of 1's occurring in the u-th row of $\sum_{z\in H}\sum_{\alpha\neq 0} A_{\alpha z}$, which is also is the total number of 1's occurring in the u-th column of $\sum_{z\in H}\sum_{\alpha\neq 0} A_{\alpha z}$.

To apply Matrix Bernstein, it remains to bound Υ_u for all k-sparse u. This now requires using that the set of k-sparse vectors H is random, and by a simple Chernoff bound[1] we conclude that the maximum is $O(mD(p-1)/N)$. Hence, we can apply Matrix Bernstein to conclude that $\|A\|_2 \le O\left(\log N + \sqrt{\frac{mD(p-1)\log N}{N}}\right)$ with high probability over the draw of H and the b_z's, where $|H| = m$. By Fact 3.6.1, we have that $D/N \sim (\frac{\ell}{(p-1)n})^{k/2}$, and so we can rephrase our assumption on m as $m \ge O\left(\frac{N\log N}{D(p-1)\varepsilon^2}\right)$.

Thus, the second term above dominates, and so we have shown that

$$D\,\mathrm{val}(f) \le N\|A\|_2 \le N\cdot O\left(\sqrt{\frac{mD(p-1)\log N}{N}}\right)$$
$$\implies \mathrm{val}(f) \le O(1)\cdot m\cdot\sqrt{\frac{N(p-1)\log N}{Dm}} \le m\cdot\varepsilon p\,.$$

As $\mathrm{val}(\Psi) = \frac{1}{p}(\mathrm{val}(f) + m)$, we have thus certified a bound of $m(\frac{1}{p}+\varepsilon)$, which finishes the proof. □

[1] See [WAM19, Section F.1.4] for a similar calculation in the $\mathbb{F}_2$ case.

Chapter 14

Improved Algorithms for Planted CSPs

14.1 Subexponential-time algorithms for planted CSPs

In Part I, we defined two different average-case problems for CSPs: refutation and search, and we discussed our results for each in Sections 4.1 and 4.2, respectively. For the task of semirandom/smoothed CSP refutation, we gave a family of algorithms that achieve a runtime vs. clause threshold trade-off. Namely, one can refute a semirandom k-ary CSP in $n^{O(\ell)}$-time when the instance has $m \geq \tilde{O}((n/\ell)^{\frac{k}{2}} \cdot \ell)$ constraints (Theorem 1). However, for the task of solving semirandom planted CSPs, we only gave a poly(n)-time algorithm to find an assignment that satisfies $1 - o(1)$ fraction of constraints when the instance has $m \geq \tilde{O}(n^{k/2})$ constraints, where k is the arity of the CSP (Theorem 3). That is, we only gave an algorithm for the "$\ell = O(1)$ case", and in particular we do not show any runtime vs. clause threshold trade-off. Thus, a lingering question is:

Question 14.1.1. *Can we give algorithms for solving semirandom planted CSPs that achieve the same runtime vs. clause threshold trade-off as in the case of refutation?*

In fact, this question is open also for the simpler case of *random* CSPs!

Below, we explain the key technical barrier that we need to overcome to extend the proof techniques of Theorem 3 to give such an algorithm. At a high level, the reason the algorithm in Theorem 3 does not generalize to the subexponential-time case is because some of the additional polylog(n) factors in m that are hidden in the $\tilde{O}(\cdot)$ notation are actually polylog(N), where N is the size of the matrix used in the spectral algorithm. In the polynomial-time case, $N = \binom{n}{k/2}$, so $\log N$ is simply $O(\log n)$, and we can afford to lose these extra factors. However, in the subexponential-time case, $N = \binom{n}{\ell}$, and so $\log N = O(\ell \log n)$, and losing factors of ℓ will not yield the "correct" threshold of $m = \tilde{O}((n/\ell)^{\frac{k}{2}} \cdot \ell)$.

The proof of Theorem 3 relies on two key ingredients: (1) expander decomposition, and (2) a spectral sparsification lemma. In order to extend this approach to achieve the subexponential-time trade-off, we expect that we need to prove a generalized spectral sparsification lemma, which we describe below. For simplicity, we will describe the lemma for the even k case only.

Definition 14.1.2 (Kikuchi Graph and Laplacian)**.** Let k be even. Given a parameter ℓ and a set $C \in \binom{[n]}{k}$, we let G_C be the graph with adjacency matrix A_C that is defined in Definition 2.1.1. Namely, G_C has vertex set $N = \binom{n}{\ell}$, where we have an edge $(S, T) \in E(G_C)$ if $S \oplus T = C$. For a k-uniform hypergraph H, we let G_H denote the union of the graphs G_C for $C \in H$. We let

$\mathcal{K} = \binom{[n]}{k}$ denote the complete hypergraph, and $G_{\mathcal{K}}$ denote the corresponding Kikuchi graph. We let L_C denote the Laplacian of G_C, and $L_H = \sum_{C \in H} L_C$ denote the Laplacian of H. We also let $L_{\mathcal{K}} = \sum_{C \in \binom{[n]}{k}} L_C$ denote the Laplacian of the complete graph.

The main spectral sparsification statement that we would like to show is the following: if H is a random k-uniform hypergraph with $|H| = m \geq \tilde{O}((n/\ell)^{\frac{k}{2}} \cdot \ell)$ hyperedges, then L_H is a good spectral approximation of $L_{\mathcal{K}}$.

Conjecture 14.1.3. *Let H be a random k-uniform hypergraph with $|H| = m \geq \tilde{O}((n/\ell)^{\frac{k}{2}} \cdot \ell)$ hyperedges. Then, with probability $\geq 1 - 1/\mathrm{poly}(n)$, it holds that $\eta(1 - o(1))L_{\mathcal{K}} \preceq L_H \preceq \eta(1 + o(1))L_{\mathcal{K}}$, where $\eta = m/\binom{n}{k}$ is the appropriate normalization factor.*

We now explain the difficulty in proving Conjecture 14.1.3 for the subexponential-time case, i.e., when ℓ is super-constant. To see this, let us first consider the case of $\ell = k/2$, which is the case that suffices to argue Theorem 3 for *random* CSPs. Here, the graph G_H is almost[1] the normalized adjacency matrix of a *random* graph on N vertices with $m = O(N \,\mathrm{polylog}(N))$ edges, and the graph $G_{\mathcal{K}}$ is the complete graph on $N = \binom{n}{k/2}$ vertices. One can easily show that G_H is a good expander graph with spectral gap $1/\sqrt{d}$, where $d = \tilde{O}(1)$ is the average degree in $\mathcal{H}$, and thus it follows that the smallest nonzero eigenvalue of $L_{\mathcal{H}}$ is $1 - 1/\sqrt{d} = 1 - o(1)$, and so Conjecture 14.1.3 holds.

However, the issue is that for larger ℓ, the graph G_H is *not* a good expander, even for *random* H or in the "complete" case $\mathcal{K}$. These non-expanding sets are easy to construct. For example, let $R \subseteq [n]$ be any set of size, say, $(1 - O(1/\ell))n$. Then, the set of all S of size ℓ with $S \subseteq R$ is a set of at least $\Omega(N)$ vertices in the Kikuchi graph that is typically non-expanding. This is because a hyperedge C can only cross this cut if $|C \cap R| \leq k - 1$, and a random hyperedge C will satisfy $C \subseteq R$ with probability $1 - O(k/\ell) = 1 - o(1)$ when ℓ is superconstant.

But, we can observe that such a non-expanding set is not just non-expanding for a typical hypergraph H, it is also non-expanding even for the complete hypergraph $\mathcal{K}$. In particular, this suggests that such vectors will have a correspondingly low quadratic form in $L_{\mathcal{K}}$ also, and so this example does not disprove Conjecture 14.1.3. However, it does give solid evidence that in order to prove Conjecture 14.1.3, one will need to do a more fined-grained analysis that uses that spectrum of $L_{\mathcal{K}}$ has eigenvalues of very different scales. This unlike the case of $\ell = k/2$, where $L_{\mathcal{K}}$ is an expander and so all its eigenvalues (except for the trivial 0 eigenvalue) are have approximately the same magnitude.

14.2 Smoothed models of planted CSPs

In Part I, we gave algorithms to refute semirandom and smoothed CSPs, whereas for the task of solving planted CSPs, we only gave an algorithm in the semirandom case. This is because in the case of planted CSPs, there is a natural way to define a semirandom planted model that is analogous to the semirandom refutation model, whereas defining a planted model analogous to the smoothed refutation model appears to be tricky. In this section, we propose a candidate smoothed model for planted CSPs. Unlike the semirandom planted model studied in Theorem 3,

[1] The gap is that a constraint C has $\binom{k}{k/2} = O(1)$ ways of being partitioned into two sets of size $k/2$. However, as this is constant, this is not a substantial difference.

where the instances generated are satisfiable, i.e., have value 1, in our proposed smoothed planted model, the CSPs generated will have *low but nontrivial value*.

A candidate smoothed model for planted CSPs. To generate the smoothed planted instance Ψ, we start from an arbitrary CSP instance Φ with predicate P, along with an initial assignment x^* that satisfies a μ_P-fraction of constraints in Φ. Here, μ_P is the fraction of constraints satisfied by a random assignment. For example, $\mu_P = 7/8$ for 3-SAT, or $1/2$ for 3-XOR. As a result, there always exists such an x^* regardless of the choice of Ψ.

We now produce a p-smoothed instance Ψ from Φ by doing the following. For each constraint in Φ, with probability p independently we rerandomize the literal negation pattern for that constraint according to the planting distribution Q and the assignment x^*. That is, for each constraint, with probability p we replace its literal negation pattern with one sampled as done in Definition 4.2.1. As a result, the assignment x^* satisfies each constraint that has been "rerandomized", and so x^* will, with high probability, satisfy at least $\mu_P(1-p) + p - o(1)$-fraction of the constraints in Ψ, which is larger than μ_P. The computational task is to now recover an assignment x that satisfies $\mu_P + \varepsilon$-fraction of constraints in Ψ, for some constant $\varepsilon := \varepsilon(p)$ that is a function of p.

Question 14.2.1. *Consider the smoothed planted CSP model defined above. Can we give an algorithm for this task, or can we prove hardness?*

To understand the above question, it is perhaps best to start with the case of k-XOR, as this case forms the backbone of the existing algorithms in the semirandom case. In the case of k-XOR, our proposed smoothed model is equivalent to the following process. First, we start with an arbitrary k-XOR instance $\Phi = (H, \{b_C\}_{C \in H})$, where H is a k-uniform hypergraph and $b_C \in \{-1, 1\}$ for each $C \in H$. We also let x^* denote an arbitrary assignment such that $\mathrm{val}_\Phi(x^*) \geq \frac{1}{2}$. To construct the smoothed instance Ψ, we do the following. For each $C \in H$, with probability p independently, we set b_C to be $\prod_{i \in C} x^*_i$, i.e., we change it to agree with x^*. Thus, with high probability, $\mathrm{val}_\Psi(x^*) \geq \frac{1}{2}(1-p) + p = \frac{1}{2}(1+p)$. The computational task is to then find an x such that $\mathrm{val}_\Psi(x) \geq \frac{1}{2} + \varepsilon$, where ε is a function of p.

Our proposed model is potentially hard. Indeed, one fundamental barrier is that the SDP value of the initial instance Φ may be very close to 1 (even if the true value is $\mu_P + o(1)$), and it is quite believable that this will remain the case after the smoothing process. If this happens, then intuitively it seems that the SDP is unable to "detect" that any smoothing has occurred, and from this it seems difficult to round the SDP to find an assignment x.

To sidestep this barrier, one could instead consider a similar model in which the initial instance Φ is furthermore guaranteed to (1) have value at most $\mu_P + o(1)$, and (2) be sampled from the smoothed model for refutation (Definition 4.1.2). Thus, by applying Theorem 1, we know that the SDP value of the initial instance Φ will be $\mu_P + o(1)$ with high probability over the initial instance Φ, whereas after the smoothing it must be at least $\mu_P(1-p) + p - o(1)$ with high probability. This means that the SDP is able to "detect" that some change has occurred to the instance Φ, which gives us more hope to find a rounding algorithm. Nonetheless, recovering an assignment x with value $\geq \mu_P + \varepsilon$ is still a nontrivial, intriguing rounding task, even in this potentially easier setting.

Chapter 15

Improved Lower Bounds for LDCs/LCCs

In Part III, we discussed in detail the problem of understanding the optimal blocklength n of a q-query locally decodable (or correctable) code with k message bits. After this thesis, the best-known upper and lower bounds on the blocklength n can be summarized as follows.

(1) When $q = 2$, the best 2-LCC (also LDC) is the Hadamard code with $n = 2^k$, and there is a matching lower bound of $n \geq 2^{\Omega(k)}$ due to [KW04, GKST06].

(2) For 3-LCCs, the best-known contruction is the degree 2 Reed–Muller code, which achieves $n = 2^{2\sqrt{2k}}$ (Section 12.11). The best-known lower bound is $2^{\Omega(k^{1/5})}$ (Theorem 10), with better lower bounds possible if one assumes that the code is linear (Theorem 8 and the follow-up works of [Yan24, AG24]) or a design LCC (Theorem 9).

(3) For 3-LDCs, the best-known construction is the matching vector code of [Yek08, Efr09], which achieves $n = 2^{2^{O(\sqrt{\log k \log\log k})}}$. The best-known lower bound is $n \geq \tilde{\Omega}(k^3)$ (Theorem 7).

(4) For $q \geq 4$, the best-known q-LCC is the degree $q-1$ Reed–Muller code, which achieves $n = 2^{O(k^{\frac{1}{q-1}})}$. The best-known q-LDC comes from matching vector codes [Yek08, Efr09], and achieves $n \leq 2^{k^{o(1)}}$. The best-known lower bound is $n \geq \tilde{\Omega}(k^{\frac{q}{q-2}})$ for even q, and $n \geq \tilde{\Omega}(k^{\frac{q+1}{q-1}})$ for odd q ([KW04] and Theorem 2.0.4).

As mentioned in Section 10.1, the contributions of this thesis are to (1) improve the lower bound for 3-LDCs from $\tilde{\Omega}(k^2)$ to $\tilde{\Omega}(k^3)$, achieving the bound of $\tilde{\Omega}(k^{\frac{q}{q-2}})$ (that we know for even q) for the odd value $q = 3$, and (2) improve the lower bound for 3-LCCs from $\tilde{\Omega}(k^2)$ to $2^{\Omega(k^{1/5})}$, with better lower bounds possible for linear and design LCCs. The obvious open question is to improve any of the above bounds (either constructions or lower bounds). In the following sections, we discuss certain concrete plausible improvements and discuss the technical barriers to proving them.

15.1 Better LDC lower bounds: barriers and a path forward

In this section, we discuss the barriers towards improving the current lower bounds for LDCs.

15.1.1 Improving odd q LDC lower bounds

The first open question, and possibly the easiest one to tackle, is to extend Theorem 7 to all odd q.

Question 15.1.1. *Theorem 7 achieves a lower bound of $\tilde{\Omega}(k^{\frac{q}{q-2}})$ (which we can prove for even q) for the odd value of $q = 3$. Can we prove a lower bound of $\tilde{\Omega}(k^{\frac{q}{q-2}})$ for all odd $q \geq 5$ as well?*

As we briefly mentioned at the end of the proof overvew in Chapter 11, the techniques used to prove Theorem 7 fail for $q \geq 5$ because of an issue with the hypergraph decomposition step that comes from the potential existence of "heavy pairs" in the hypergraphs $H_1, \ldots, H_k$. We can thus prove a lower bound of $\tilde{\Omega}(k^{\frac{q}{q-2}})$ for odd $q \geq 5$ under the assumption that the hypergraphs $H_1, \ldots, H_k$ have *no heavy pairs*, which is an analogous assumption to the design case for LCCs.

The main technical barrier to proving Question 15.1.1 is the following. When we have heavy pairs, the natural way to handle them is via a hypergraph decomposition step, as done in several other instances in this thesis, e.g., Sections 5.2 and 12.5. In the decomposition step, we set "cut-off thresholds" that determine when a set Q is "heavy"; these thresholds are determined by the regularity property that we need to enforce on the initial hypergraphs to make the row pruning step succeed for the "top level" q-XOR instance. Once we have set these thresholds, the decomposition step produces a family of "bipartite" instances, analogous to Section 5.4, and for each of these instances, we need the underlying hypergraph to satisfy a (possibly different) regularity condition so that the row pruning for each bipartite instance will also succeed. The hypergraph for each bipartite instance does inherit some regularity properties via the decomposition. In, e.g., Sections 5.2 and 5.4, the inherited regularity of each bipartite instance matches the required regularity that we need to refute the instance, and so the analysis works out. The technical issue to proving Question 15.1.1 is that if one uses the natural generalization of the decomposition in Section 5.2, the inherited regularity is weaker than required, and we are unable to refute the bipartite instances. The primary reason we are able to succeed for $q = 3$ is that the resulting bipartite instance is arity $3 - 1 = 2$, and refuting 2-XOR instances is substantially easier than larger arity XOR.

15.1.2 Improving even q LDC lower bounds

A perhaps more intriguing question is whether one can prove a q-LDC lower bound beyond $\tilde{\Omega}(k^{\frac{q}{q-2}})$ for any choice of q (in particular, $q = 4$, say). An affirmative answer to this question would be interesting even if one assumes that the code is a linear q-LDC, or even a design q-LDC (which has no heavy pairs).

Question 15.1.2. *Can we prove a q-LDC lower bound (even for linear or design LDCs) beyond $n \geq \tilde{\Omega}(k^{\frac{q}{q-2}})$ for some $q \geq 3$?*

To explain the technical barriers to answering Question 15.1.2, let us first recall how the $\tilde{\Omega}(k^{\frac{q}{q-2}})$ threshold arises. This bounds arises from the degree heuristic calculation done in Section 12.1.1, where we observe that the Kikuchi graph $A_i \coloneqq \sum_{C \in H_i} A_C$ has average degree $\sim \delta n(\ell/n)^{q/2}$. By applying Matrix Khintchine (Fact 3.4.2) and rearranging terms, we found that our lower bound has the form $k \leq \tilde{O}(\ell)$, and we need to choose ℓ to make the average degree of A_i be $\gtrsim 1$, i.e., we need $\ell \gtrsim n^{1-2/q}$. Thus, going beyond the $\tilde{\Omega}(k^{\frac{q}{q-2}})$ requires a new method to proving LDC lower bounds that goes beyond the "degree heuristic".

The trace moment method behaves poorly for large ℓ. Ideally, one would like to take $\ell = \Omega(n)$, as we expect the Kikuchi matrices to yield tighter bounds on the value of the q-LDC XOR instance Ψ_b as ℓ increases. However, a strange feature of the current analysis is that the bounds get weaker

as ℓ increases beyond $n^{1-2/q}$, which is counter to the expected behavior.

The reason this behavior appears[1] is that we bound $\mathbb{E}_b[\mathrm{val}(\Psi_b)]$ by $\mathbb{E}_b[\|A\|_2]$, which we then bound (implicitly via the trace moment method) by $\left(\mathbb{E}_b[\mathrm{tr}(A^{2r})]\right)^{1/2r}$ where $r = O(\log N)$. The issue is that with probability 2^{-k} over the draw of $b \leftarrow \{-1,1\}^k$, we have $b_i = 1$ for all $i \in [k]$, and when this happens we have $\mathrm{val}(\Psi_b) = 1$, and so $\|A\|_2$ is truly large. However, to bound $\mathbb{E}_b[\|A\|_2]$, we compute $\mathbb{E}_b[\mathrm{tr}(A^{2r})]$ and then take $2r$-th roots. Thus, the contribution of this "bad event" to $(\mathbb{E}_b[\mathrm{tr}(A^{2r})])^{1/2r}$ is quite large when $k \ll r$, as then $2^{-k/2r} = 1 - o(1)$, and this prevents us from obtaining a good lower bound when $r = O(\ell \log n)$ is much larger than k. A natural way to circumvent this issue is to separate out the "rank 1 component" by considering the matrix $\left(\sum_{i=1}^k b_i(A_i - J_N)\right) + \left(\sum_{i=1}^k b_i J_N\right)$, where J_N is the $N \times N$ all 1's matrix, but so far we have not been able to use this approach to prove a better lower bound.

The necessity of larger ℓ: rainbow even covers. On the other hand, we can argue formally that one cannot improve Theorem 2.0.4 or tighten the analysis in Section 2.3 without taking $\ell \gg n^{1-2/q}$. This argument goes by connecting the problem of LDC lower bounds to the problem of finding certain colored even covers in hypergraphs, as we now explain.

Definition 15.1.3 (Odd-colored and rainbow even covers)**.** Let $H_1, \ldots, H_k$ be q-uniform hypergraph matchings on n vertices, each of size δn. Let $H := \cup_{i=1}^k H_i$ be a collection of hyperedges *colored* by k different colors, where we view a hyperedge $C \in H_i$ as appearing in H with color i. An even cover (Definition 9.0.1) in $H := \cup_{i=1}^k H_i$ is a collection of hyperedges $C_1, \ldots, C_t$ in H such that $C_1 \oplus \cdots \oplus C_t = \emptyset$. We say that the even cover is *odd-colored* if some color $i \in [k]$ appears an odd number of times, and we say that the even cover is *rainbow* if every color appears at most once in the even cover.

The *odd-colored* even cover problem is to determine the extremal value of k (as a function of n, δ) such that any k q-uniform matchings $H_1, \ldots, H_k$ must contain an odd-colored even cover. The *rainbow* even cover problem is defined similarly, just for rainbow even covers.

The connection between Definition 15.1.3 and LDC lower bounds is the following. One can observe that a set of q-uniform matchings $H_1, \ldots, H_k$ form a valid linear q-LDC if and only if, for any choice of $b_1, \ldots, b_k$, there is a solution to the q-XOR instance corresponding to these matchings. One can also observe, via a simple linear-algebraic argument, that there is a solution for any $b \in \{-1,1\}^k$ if and only if $H_1, \ldots H_k$ do not contain an odd-colored even cover. Thus, the odd-colored even cover threshold k is exactly the maximum dimension of a (q, δ)-linear LDC in normal form!

Reinterpreting our proof in Section 2.3 through this lens, we see that it shows that for q even, if $k \geq O(n^{1-2/q} \log n)$, then there must exist an odd-colored even cover of length $\leq O(n^{1-2/q} \log n)$ In fact, if we use the connection between the trace moments for Kikuchi matrices and even covers that we discussed in Part II, we can observe that our proof shows the following stronger statement: not only is there an odd-colored even cover of length $\leq O(n^{1-2/q} \log n)$, the even cover uses each color $i \in [k]$ at most *once*. Namely, it is a *rainbow* even cover.

We can now explain why analysis in Section 2.3 is tight for $\ell = n^{1-2/q}$. The crux of the issue is that, if one chooses $H_1, \ldots H_k$ at random, then the above result for rainbow even covers is in fact *tight*: if $k \lesssim n^{1-2/q}$, then with high probability over $H_1, \ldots, H_k$ chosen randomly, there is no rainbow even cover. This can be shown via some simple concentration bounds. To get

[1] The following observations were made in joint discussions with Jun-Ting Hsieh and Pravesh K. Kothari.

some intuition for why this is the right threshold, we note that the hypergraph $H = \cup_{i=1}^{k} H_i$ is a q-uniform hypergraph with $m = \delta n k$ hyperedges, so following Conjecture 8.0.2, we expect the length of the shortest even cover to be $\sim \ell$, where $m \sim (n/\ell)^{\frac{q}{2}}\ell$, and so here $\ell \sim nk^{-\frac{2}{q-2}}$. Thus, if $k \ll n^{1-2/q}$, then we expect the shortest even cover to have length $\gg k$. In particular, there is no rainbow even cover, as any rainbow even cover has length $\leq k$. The key point is that, by using the connection between the trace moments for Kikuchi matrices and even covers that we discussed in Part II, the spectral norm $\|A\|_2$ only "contains information" about even covers of length at most $O(\ell \log n)$. When $k \ll n^{1-2/q}$, there are no violated[2] rainbow even covers (or any rainbow even covers at all!), and so the spectral norm $\|A\|_2$ cannot "see" any contradictions and therefore "thinks" that the instance is satisfiable.

The above shows that when $k \ll n^{1-2/q}$, we cannot prove better LDC lower bounds using only rainbow even covers. We must therefore make use of odd-colored even covers. The challenge with generalizing to odd-colored even covers comes from the bound on the spectral norm of $A = \sum_{i=1}^{k} b_i A_i$ that we obtain via the trace method. Currently, our techniques are not very good at bounding the number of walk terms that correspond to using each b_i at least 4 times (rather than the typical "at least 2 times"), and this is the dominant term when $k \ll n^{1-2/q}$.

As a final remark, we note that the recent work of [HKM+24] proves a linear 3-LDC lower bound of $k \leq O(n^{1/3} \log n)$, which has a better polylog(n) factor than Theorem 7, which achieves $k \leq O(n^{1/3} \log^2 n)$, or Corollary 11.3.3, which achieves $k \leq O(n^{1/3} \log^{4/3} n)$. They obtain this small improvement by working with an intermediate notion of odd-colored even covers, where they show the existence of an even cover that uses *some* color exactly once.

15.2 The "LDC barrier" for LCC lower bounds

In this thesis, we have proven exponential lower bounds for 3-LCCs. In the design 3-LCC case, we proved a tight lower bound, proving that Reed–Muller codes give optimal design 3-LCCs, and in the case of linear and nonlinear (smooth) LCCs, we proved that Reed–Muller codes are near-optimal. The major open question is now: can we extend these results to 4-query LCCs?

Question 15.2.1. *Can we prove better lower bounds for 4-LCCs? Namely, can we show:*

(1) A superpolynomial lower bound?

(2) An exponential lower bound?

Let us now discuss the main technical barriers that we will encounter when trying to answer Question 15.2.1 by using the methods in this thesis.

The "degree heuristic calculation" for $q \geq 4$. First, we observe that a naive application of the long chain derivation can likely improve the lower bounds for q-LCCs beyond those known for LDCs, even for $q \geq 4$, although this naive application of our approach will likely to only yield a polynomial factor improvement. Our explanation is rooted in the "degree heuristic" calculation based on the density of the Kikuchi matrices explained earlier in Section 12.1.1. For larger q, the number of length $(r+1)$-chains with head $i \in [k]$ is still $k(3\delta n)^{r+1}$. The arity of the derived constraints, however, is now $(q-1)(r+1)+1$. This means that the density (i.e., average degree of the natural Kikuchi matrix) at level ℓ is $(3\delta n)^{r+1}(\ell/n)^{\frac{(q-1)(r+1)+1}{2}} \to \left(n(\ell/n)^{\frac{q-1}{2}}\right)^{r+1}$ for large r.

[2] Even covers whose "right-hand sides" multiply to -1.

Thus, the optimal ℓ turns out to be $n^{1-\frac{2}{q-1}}$, and so we can only hope to achieve a lower bound of $k \leq \tilde{O}(n^{1-\frac{2}{q-1}})$. This nevertheless would yield an improvement on the current best-known lower bound[3] of $k \leq \tilde{O}(n^{1-\frac{2}{q}})$, inherited from q-LDCs, by a polynomial factor via long chains. In fact, for odd q, our methods generalize to this case in a fairly straightforward manner, and one can indeed prove this bound, though we will not do so in this thesis.[4]

A reduction from q-LCCs to $(q-1)$-LDCs via long chains. We observe that one obtains the threshold of $n^{1-\frac{2}{q-1}}$ by substituting in $q-1$ for q in the existing[5] q-LDC lower bounds of $n^{1-\frac{2}{q}}$, which potentially suggests a connection between q-LCCs and $(q-1)$-LDCs. In fact, as we showed in Sections 12.2 and 12.6.5, we can use our long chain derivation strategy to give a reduction from a 3-LCC of length n to a 2-LDC of length $n^{\text{polylog}(n)}$. More generally, one might hope to use long chain derivations to give a reduction from a q-LCC of length n to a $(q-1)$-LDC of length $n^{\text{polylog}(n)}$, and this does appear to be fairly straightforward to show using the techniques in this thesis.[6] However, because the reduction blows up the length of the code by a polylog(n) factor in the exponent, current $(q-1)$-LDC lower bounds are not strong enough to yield any improved q-LCC lower bounds via this route except for $q = 3$, which succeeded because for 2-LDCs we can prove exponential lower bounds.

The subexponential "LDC barrier" for long chains. The above discussion thus implies that if we could obtain (a large enough) superpolynomial 3-LDC lower bound, then we could prove a superpolynomial 4-LCC lower bound and thus answer Item (1) of Question 15.2.1 in the affirmative. On the other hand, one cannot use a reduction from 4-LCCs to 3-LDCs to prove an exponential lower bound for 4-LCCs (and thus answer Item (2) of Question 15.2.1), as there are constructions of 3-LDCs of subexponential length! Thus, even if we could obtain substantially better 3-LDC lower bounds, or even prove that the existing constructions of [Yek08, Efr09] are optimal, the reduction from 4-LCCs to 3-LDCs will at best only yield a subexponential lower bound for 4-LCCs.[7] On the other hand, one could wonder if this barrier is appearing not because of a defect of our techniques, but rather because it is the "truth". That is, perhaps the connection between 4-LCCs and 3-LDCs also goes in the reverse direction, and so we can ask:

Question 15.2.2. *Do there exist 4-LCCs of subexponential length?*

The above discussion could be viewed as suggesting that such a construction is plausible.

Finally, we remark that a very recent work of [AG24] proves, for LCCs over small fields of characteristic 2: (1) a lower bound of $k \leq O(\log^2 n \log\log n)$ for linear 3-LCCs, and (2) a lower bound of $k \leq \tilde{O}(n^{1-2/(q-1)})$ for linear q-LCCs where q is odd (this matches, up to polylog(n)

[3] At least, for even q. For odd q, the best-known lower bound is weaker.

[4] When q is even, there are additional technical challenges to overcome because each "link" in the chain has $q-1$ vertices, which is odd.

[5] At least for even q and $q = 3$, although the degree heuristic calculation predicts this threshold for all q.

[6] Let us spell out the reduction in a bit more detail in this footnote. Rather than construct a Kikuchi matrix with rows $S^{(1)}$ and columns $S^{(2)}$ (which corresponds to 2 queries), we now construct a $(q-1)$-tensor with modes indexed by $S^{(1)}, \ldots, S^{(q-1)}$. For each "link" in the chain, we split the $q-1$ uncanceled entries across the $q-1$ sets $S^{(1)}, \ldots, S^{(q-1)}$, which form the $q-1$ queries.

[7] There is still some hope that, if one were to prove such an LDC lower bound, combining the new proof strategy with the long chains might yield better LCC lower bounds than the ones predicted by the q-LCC to $(q-1)$-LDC reduction. This is not that inconceivable, as one can use long chains to prove a q-LCC lower bound, for odd $q \geq 5$, of $k \leq \tilde{O}(n^{1-2/(q-1)})$, whereas one cannot obtain any improvement using the q-LCC to $(q-1)$-LDC reduction because it increases the length from n to $n^{\text{polylog}(n)}$.

factors, the threshold predicted by our earlier heuristic calculation). Their proof goes via a reduction from q-LCCs to the rainbow even cover problem for $(q-1)$-uniform hypergraph matchings (Definition 15.1.3), and then applies (in Case (1)) the recent breakthrough of [ABS+23] on the rainbow cycle bound for graphs, and (in Case (2)) the rainbow cycle bound shown implicitly (via the analysis in Section 2.3) by existing $(q-1)$-LDC lower bounds where $q-1$ is even (as q is odd). Their approach is reminiscent of the q-LCC to $(q-1)$-LDC reduction discussed above, although their reduction is to the (stronger) problem of rainbow cycles. Although we can reduce LDC lower bounds to the rainbow cycle problem, the difference between LDCs and rainbow cycles turns out to be a major difference because, as we discussed in Section 15.1.2, we have a lower bound of $k \gtrsim n^{1-2/q}$ on the rainbow cycle threshold for q-uniform matchings. Thus, we know that one cannot obtain q-LCC lower bounds better than $k \leq O(n^{1-2/(q-1)} \operatorname{polylog}(n))$ (except for small $\operatorname{polylog}(n)$ factor improvements) for $q \geq 4$ via this reduction.

Nonetheless, the work of [AG24] does not obtain any improved lower bounds for 4-LCCs, and so the simplest open problem for $q = 4$ remains:

Question 15.2.3. *Can we prove a lower bound of $k \lesssim n^{\frac{1}{2}-\varepsilon}$ for binary 4-LCCs, for some constant $\varepsilon > 0$?*

Chapter 16

Improved Nondeterministic and Interactive Refutations

In this chapter, we give some open problems related to the work of [FKO06] and our extensions to semirandom/smoothed instances done in Chapter 6. Recall that in Chapter 6, we showed that there is a *nondeterministic* polynomial-time refutation algorithm to (weakly) refute semirandom/smoothed instances of 3-SAT with $m \geq \tilde{O}(n^{1.4})$ constraints, which is below the $\tilde{O}(n^{1.5})$ constraint threshold required for polynomial-time algorithms (Chapter 5). This extends the results of [FKO06], which was for fully random instances, to the case of semirandom/smoothed instances. Equivalently, this shows the existence of short, efficiently verifiable witnesses of unsatisfiability for semirandom/smoothed instances with $\tilde{O}(n^{1.4})$ constraints, whereas we can only *find* such witnesses efficiently when instances have $\tilde{O}(n^{1.5})$ constraints. We also showed analogous results for k-ary CSPs more generally, extending results of [FW15, Wit17] to the semirandom/smoothed setting.

The first immediate open question is the following.

Question 16.0.1. *Is there a nondeterministic polynomial-time algorithm to refute random* 3*-SAT instances with $m \lesssim n^{1.4-\varepsilon}$ constraints, for some constant $\varepsilon > 0$?*

We note that the FKO-style strategy used in Chapter 6 cannot go beyond the $n^{1.4}$ threshold. This is because Theorem 6 achieves the optimal girth vs. density trade-off for hypergraphs (up to polylog(n) factors), and so the weak refutation of the "top level" 3-XOR instance using the even covers cannot be improved. In fact, it cannot be improved even for random hypergraphs, as the proof of near-optimality for the trade-off in Theorem 6 argues near-optimality by showing that a random k-uniform hypergraph H with $m \lesssim (n/\ell)^{k/2}\ell$ hyperedges has no even cover of length $O(\ell \log n)$.

On the other hand, it is possible that $n^{1.4}$ is the optimal threshold for nondeterministic refutation. However, so far the only (rather weak) evidence we have to suggest that this is the case is the argument in the above paragraph. Of course, proving any lower bound requires making some complexity assumption (at the very least, e.g., $\mathsf{NP} \neq \mathsf{coNP}$), and so we can ask:

Question 16.0.2. *Can we prove, under plausible assumptions, that there is no nondeterministic polynomial-time algorithm to refute random* 3*-SAT instances with $m \lesssim n^{1.4-\varepsilon}$ constraints, for some constant $\varepsilon > 0$?*

One way to approach the above question is to try to prove lower bounds in restricted proof systems, such as sum-of-squares. As we have mentioned in Part I, there is a known lower bound,

due to [KMOW17], of $m \lesssim (n/\ell)^{k/2}\ell$ for SoS proofs of degree $O(\ell)$, i.e., if $m \lesssim (n/\ell)^{k/2}\ell$, then with high probability over the draw of the random k-SAT instance, there is no degree $O(\ell)$ SoS proof of unsatisfiability. In the lower bound of [KMOW17], the "measure of complexity" of the proof is the SoS degree, which is analogous to the runtime of the corresponding deterministic SoS-based algorithm. To capture the notion of *nondeterministic* algorithms, the correct measure of complexity is the *size* of the SoS proof.

Question 16.0.3. *Can we prove, for some constant $\varepsilon > 0$, that with high probability over the draw of a random 3-SAT instance with $m \lesssim n^{1.4-\varepsilon}$ constraints, there is no polynomial-sized SoS proof of unsatisfiability?*

This question of proving polynomial-size lower bounds for SoS, as well as in other proof systems, was also posed in Section 5.3.2 in [Wit17].

We note that there is known relationship between size and degree of SoS proofs due to [AH19]: if there is an SoS proof of size s, then there is a proof of degree $O(\sqrt{n \log s})$. Combining this with the SoS lower bound of [KMOW17] (and setting $\ell = O(\sqrt{n}\,\mathrm{polylog}(n))$), we can show that, e.g., for k-SAT, there is no $\mathrm{poly}(n)$-sized SoS proof of unsatisfiability when $m \lesssim n^{k/4+1/2}$. For 3-SAT, this narrows the range of possible m to between $n^{1.25}$ and $n^{1.4}$.

Subexponential-time nondeterministic refutations. We can also ask what we can achieve if we allow our nondeterministic refutation algorithm to run in subexponential time.

Question 16.0.4. *For what m, as a function of n, k, and ℓ, is there a nondeterministic $n^{O(\ell)}$-time algorithm to refute random k-SAT instances with m constraints?*

We note than one can trivially improve the thresholds shown in Chapter 6 by using the $n^{O(\ell)}$-time deterministic algorithm of Chapter 5 (instead of the $\mathrm{poly}(n)$-time algorithm) to better refute the t-XOR instances for $t \leq k-1$, and then, as before, using violated even covers to construct the polynomial-time verifiable certificate for the "top level" k-XOR instance. Thus, to obtain an interesting answer to the above question, one would need to use the extra allotted runtime in the "nondeterministic part" of the algorithm.

Nondeterministic interactive refutations. Another interesting question, posed by [Wit17], is to ask what we can achieve if we allow the refutation procedure to be *interactive*. That is, we can consider AM protocols instead of NP algorithms.

Question 16.0.5. *Is there an AM protocol (or constant round interactive proof) to refute random 3-SAT instances with $m \lesssim n^{1.4-\varepsilon}$ constraints, for some constant $\varepsilon > 0$?*

FKO for other refutation problems. Another final interesting question is to find other problems with "FKO-like" behavior. Namely, can we find other refutation problems where there is a nondeterministic refutation algorithm that outperforms the best-known deterministic algorithms? As a concrete example, we pose the following question for the well-studied planted clique problem for $G(n, 1/2)$.

Question 16.0.6. *Is there a nondeterministic polynomial-time algorithm to refute the existence of a clique of size $n^{\frac{1}{2}-\varepsilon}$ in a random graph $G \sim G(n, 1/2)$, for some constant $\varepsilon > 0$?*

We note that for planted clique, the classic spectral algorithm certifies that there is no clique of size larger than $\sqrt{n}$, and moreover there is an SoS lower bound ([BHK+16]) that gives evidence

that the $\sqrt{n}$ threshold is optimal for polynomial-time *deterministic* algorithms.

Finally, we remark that the recent work of [BR23] gives a nondeterministic *interactive* refutation protocol for the "nearest boolean vector" problem that beats known SoS lower bounds, which is a result in the spirit of Questions 16.0.5 and 16.0.6.

Bibliography

[Abb18] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[ABH16] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Trans. Inf. Theory*, 62(1):471–487, 2016.

[ABS+23] Noga Alon, Matija Bucić, Lisa Sauermann, Dmitrii Zakharov, and Or Zamir. Essentially tight bounds for rainbow cycles in proper edge-colourings. *arXiv preprint arXiv:2309.04460*, 2023.

[ACIM01] Dimitris Achlioptas, Arthur Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k SAT and NAE 3-SAT. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 721–722, 2001.

[AE98] Gunnar Andersson and Lars Engebretsen. Better approximation algorithms for Set splitting and Not-All-Equal SAT. *Information Processing Letters*, 65(6):305–311, 1998.

[AF09] Noga Alon and Uriel Feige. On the power of two, three and four probes. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 346–354. SIAM, Philadelphia, PA, 2009.

[AG24] Omar Alrabiah and Venkatesan Guruswami. Near-tight bounds for 3-query locally correctable binary linear codes via rainbow cycles. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024.

[AGK21] Jackson Abascal, Venkatesan Guruswami, and Pravesh K. Kothari. Strongly refuting all semi-random Boolean CSPs. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 454–472. SIAM, 2021.

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom CSP refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1438–1448. ACM, 2023.

[AH19] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for sums-of-squares and positivstellensatz proofs. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The moore bound for irregular

graphs. *Graphs Comb.*, 18(1):53–57, 2002.

[Ahn20] Kwangjun Ahn. A simpler strong refutation of random k-xor. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[AK92] E. F. Assmus and J. D. Key. *Designs and their Codes*. Cambridge Tracts in Mathematics. Cambridge University Press, 1992.

[AKK95] Sanjeev Arora, David R. Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of *NP*-hard problems. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 284–293. ACM, 1995.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 25-27 January 1998, San Francisco, California, USA*, pages 594–598. ACM/SIAM, 1998.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[ALWZ20] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 624–630. ACM, 2020.

[AOW15] Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to Refute a Random CSP. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 689–708. IEEE Computer Society, 2015.

[App16] Benny Applebaum. Cryptographic Hardness of Random Local Functions: Survey. *Computational complexity*, 25:667–722, 2016.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

[AS21] Vahid R Asadi and Igor Shinkar. Relaxed locally correctable codes with improved parameters. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[BBH23] Afonso S Bandeira, March T Boedihardjo, and Ramon van Handel. Matrix concentration inequalities and free probability. *Inventiones mathematicae*, pages 1–69, 2023.

[BCG20] Arnab Bhattacharyya, L Sunil Chandran, and Suprovat Ghoshal. Combinatorial lower bounds for 3-query ldcs. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, page 85. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

[BCK15] Boaz Barak, Siu On Chan, and Pravesh K. Kothari. Sum of Squares Lower Bounds from Pairwise Independence. In *Proceedings of the Forty-Seventh Annual ACM on*

Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015, pages 97–106. ACM, 2015.

[BDL13] Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New bounds for matching vector families. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 823–832. ACM, 2013.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.*, 3:307–318, 1993.

[BGH+04] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps and applications to coding. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 1–10. ACM, 2004.

[BGLR93] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 294–304, 1993.

[BGMT12] Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. SDP gaps from pairwise independence. *Theory of Computing*, 8(1):269–289, 2012.

[BGT17] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[Bha19] Vijay Bhattiprolu. *On the Approximability of Injective Tensor Norm*. Phd thesis, Carnegie Mellon University, June 2019.

[BHK+16] Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 428–437. IEEE Computer Society, 2016.

[BHL+02] Wolfgang Barthel, Alexander K Hartmann, Michele Leone, Federico Ricci-Tersenghi, Martin Weigt, and Riccardo Zecchina. Hiding solutions in random satisfiability problems: A statistical mechanics approach. *Physical review letters*, 88(18):188701, 2002.

[BIW10] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private pir. *Algorithmica*, 58(4):831–859, 2010.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *Journal of the ACM (JACM)*, 42(1):269–291, 1995.

[BKS22] Rares-Darius Buhai, Pravesh K Kothari, and David Steurer. Algorithms approaching the threshold for semi-random planted clique. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing*, 2022.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–

595, 1993.

[BM16] Boaz Barak and Ankur Moitra. Noisy Tensor Completion via the Sum-of-Squares Hierarchy. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 417–445. JMLR.org, 2016.

[BQ09] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 12th International Workshop, APPROX 2009*, pages 392–405. Springer, 2009.

[BR23] Andrej Bogdanov and Alon Rosen. Nondeterministic interactive refutations for nearest boolean vector. In *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 28:1–28:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[BS95] Avrim Blum and Joel Spencer. Coloring Random and Semi-Random k-Colorable Graphs. *J. Algorithms*, 19(2):204–234, 1995.

[BS14] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *CoRR*, abs/1404.5236, 2014.

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares, 2016. Lecture notes in preparation, available on http://sumofsquares.org.

[CCF10] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.

[CGL04] Amin Coja-Oghlan, Andreas Goerdt, and André Lanka. Strong refutation heuristics for random k-sat. In *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques*, volume 3122 of *Lecture Notes in Computer Science*, pages 310–321. Springer, 2004.

[CGS20] Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1395–1411. SIAM, 2020.

[CGW10] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Efficient and error-correcting data structures for membership and polynomial evaluation. In *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, volume 5 of *LIPIcs*, pages 203–214. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.

[CKL+22] Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 612–623. IEEE, 2022.

[CY23] Gil Cohen and Tal Yankovitz. Asymptotically-good rlccs with $(\log{n})^{2+o(1)}$ queries. *Electron. Colloquium Comput. Complex.*, TR23-110, 2023.

[DGGW19] Zeev Dvir, Sivakanth Gopi, Yuzhou Gu, and Avi Wigderson. Spanoids - an abstraction of spanning structures, and a barrier for lccs. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 32:1–32:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[DHV78] Jean Doyen, Xavier Hubaut, and Monique Vandensavel. Ranks of incidence matrices of steiner triple systems. *Mathematische Zeitschrift*, 163:251–259, 1978.

[DS05] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 592–601. ACM, 2005.

[DSS14] Jian Ding, Allan Sly, and Nike Sun. Satisfiability threshold for random regular NAE-SAT. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 814–822, 2014.

[DSW14] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Breaking the quadratic barrier for 3-lcc's over the reals. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 784–793. ACM, 2014.

[Dvi10] Zeev Dvir. On matrix rigidity and locally self-correctable codes. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 291–298. IEEE Computer Society, 2010.

[Dvi12] Zeev Dvir. Incidence theorems and their applications. *CoRR*, abs/1208.5073, 2012.

[Dvi16] Zeev Dvir. Lecture notes on linear locally decodable codes. https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC8.pdf, Fall 2016.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 39–44. ACM, 2009.

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 534–543, 2002.

[Fei07] Uriel Feige. Refuting Smoothed 3CNF Formulas. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 407–417. IEEE Computer Society, 2007.

[Fei08] Uriel Feige. Small linear dependencies for binary vectors of low weight. In *Building Bridges: Between Mathematics and Computer Science*, pages 283–307. Springer, 2008.

[FK01] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, 2001.

[FKO06] Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3cnf formulas. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 497–508. IEEE Computer Society, 2006.

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic Proofs and Efficient Algorithm Design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[FLP16] Dimitris Fotakis, Michael Lampis, and Vangelis Th. Paschos. Sub-exponential Approximation Schemes for CSPs: From Dense to Almost Sparse. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 37:1–37:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[FPV15] Vitaly Feldman, Will Perkins, and Santosh S. Vempala. Subsampled Power Iteration: a Unified Algorithm for Block Models and Planted CSP's. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2836–2844, 2015.

[FPV18] Vitaly Feldman, Will Perkins, and Santosh S. Vempala. On the Complexity of Random Satisfiability Problems with Planted Solutions. *SIAM Journal on Computing*, 47(4):1294–1338, 2018.

[FW15] Uriel Feige and David Witmer. Nondeterministic refutation of any csp beyond spectral methods. 2015.

[FW16] Uriel Feige and Tal Wagner. Generalized girth problems in graphs and hypergraphs, 2016.

[GHKM23] Venkatesan Guruswami, Jun-Ting Hsieh, Pravesh K. Kothari, and Peter Manohar. Efficient algorithms for semirandom planted csps at the refutation threshold. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 307–327. IEEE, 2023.

[GK01] Andreas Goerdt and Michael Krivelevich. Efficient recognition of random unsatisfiable k-sat instances by spectral methods. In *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*, volume 2010 of *Lecture Notes in Computer Science*, pages 294–304. Springer, 2001.

[GKM22] Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. Algorithms and certificates for Boolean CSP refutation: smoothed is no harder than random. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 678–689. ACM, 2022.

[GKST06] Oded Goldreich, Howard Karloff, Leonard J Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[GL03] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-sat instances efficiently. *Electron. Notes Discret. Math.*, 16:21–46, 2003.

[Gol00] Oded Goldreich. Candidate One-Way Functions Based on Expander Graphs. *Electron. Colloquium Comput. Complex.*, 2000.

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, 2001.

[GRR20] Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory of Computing*, 16(1):1–68, 2020.

[Ham73] Noboru Hamada. On the p-rank of the incidence matrix of a balanced or partially balanced incomplete block design and its applications to error correcting codes. *Hiroshima Mathematical Journal*, 3(1):153–226, 1973.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[HKM23] Jun-Ting Hsieh, Pravesh K. Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph Moore bound. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2324–2344. SIAM, 2023.

[HKM+24] Jun-Ting Hsieh, Pravesh K. Kothari, Sidhanth Mohanty, David Munhá Correia, and Benny Sudakov. Small even covers, locally decodable codes and restricted subgraphs of edge-colored kikuchi graphs. *CoRR*, abs/2401.11590, 2024.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HO75] N Hamada and H Ohmori. On the bib design having the minimum p-rank. *Journal of Combinatorial Theory, Series A*, 18(2):131–140, 1975.

[IK99] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 79–88. ACM, 1999.

[IK04] Yuval Ishai and Eyal Kushilevitz. On the hardness of information-theoretic multiparty computation. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 439–455. Springer, 2004.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[IS18] Eran Iceland and Alex Samorodnitsky. On coset leader graphs of structured linear codes. *Electron. Colloquium Comput. Complex.*, TR18-023, 2018.

[JHL+12] Domingos Dellamonica Jr., Penny E. Haxell, Tomasz Luczak, Dhruv Mubayi, Brendan Nagle, Yury Person, Vojtech Rödl, Mathias Schacht, and Jacques Verstraëte. On even-degree subgraphs of linear hypergraphs. *Comb. Probab. Comput.*, 21(1-2):113–127, 2012.

[JMS07] Haixia Jia, Cristopher Moore, and Doug Strain. Generating Hard Satisfiable Formulas by Hiding Solutions Deceptively. *Journal of Artificial Intelligence Research*, 28:107–118, 2007.

[JT09] Dieter Jungnickel and Vladimir D. Tonchev. Polarities, quasi-symmetric designs, and Hamada's conjecture. *Des. Codes Cryptogr.*, 51(2):131–140, 2009.

[Jun84] Dieter Jungnickel. The number of designs with classical parameters grows exponentially. *Geom. Dedicata*, 16(2):167–178, 1984.

[Jun11] Dieter Jungnickel. Recent results on designs with classical parameters. *J. Geom.*, 101(1-2):137–155, 2011.

[Kan94] William M. Kantor. Automorphisms and isomorphisms of symmetric and affine designs. *J. Algebraic Combin.*, 3(3):307–338, 1994.

[Kar94] David R Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 648–657, 1994.

[KM24a] Pravesh K. Kothari and Peter Manohar. An exponential lower bound for linear 3-query locally correctable codes. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 776–787. ACM, 2024.

[KM24b] Pravesh K. Kothari and Peter Manohar. Superpolynomial lower bounds for smooth 3-lccs and sharp bounds for designs. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024.

[KM24c] Vinayak M. Kumar and Geoffrey Mon. Relaxed local correctability from local testing. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1585–1593. ACM, 2024.

[KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 132–145. ACM, 2017.

[KMZ12] Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Reweighted Belief Propagation and Quiet Planting for Random k-SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3-4):149–171, 2012.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *Journal of the ACM (JACM)*, 61(5):1–20, 2014.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86, 2000.

[KV00] Jeong Han Kim and Van H Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[KW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69(3):395–420, 2004.

[KZ09] Florent Krzakala and Lenka Zdeborová. Hiding Quiet Solutions in Random Constraint Satisfaction Problems. *Physical review letters*, 102(23):238701, 2009.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 2–10. IEEE Computer Society, 1990.

[LLT00] Clement Lam, Sigmund Lam, and Vladimir D. Tonchev. Bounds on the number of affine, symmetric, and Hadamard designs and matrices. *J. Combin. Theory Ser. A*, 92(2):186–196, 2000.

[LLT01] Clement Lam, Sigmund Lam, and Vladimir D. Tonchev. Bounds on the number of Hadamard designs of even order. *J. Combin. Des.*, 9(5):363–378, 2001.

[LP91] Françoise Lust-Piquard and Gilles Pisier. Noncommutative Khintchine and Paley inequalities. *Ark. Mat.*, 29(2):241–260, 1991.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LT02] Clement Lam and Vladimir D. Tonchev. A new bound on the number of designs with classical affine parameters. volume 27, pages 111–117. 2002. Special issue in honour of Ronald C. Mullin, Part II.

[Mar88] G. A. Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 529–537. IEEE Computer Society, 2001.

[Mek14] Raghu Meka. Discrepancy and beating the union bound. https://windowsontheory.org/2014/02/07/discrepancy-and-beating-the-union-bound, February 2014.

[MNS15] Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for the planted bisection model. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 69–75. ACM, 2015.

[Mos15] Dana Moshkovitz. The Projection Games Conjecture and the NP-Hardness of $\ln n$-Approximating Set-Cover. *Theory Comput.*, 11:221–235, 2015.

[MR10] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5):29:1–29:29, 2010.

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On ε-biased generators in NC0. *Random Structures & Algorithms*, 29(1):56–81, 2006.

[MW16] Ryuhei Mori and David Witmer. Lower Bounds for CSP Refutation by SDP Hierarchies. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 41:1–41:30, 2016.

[NV08] Assaf Naor and Jacques Verstraëte. Parity check matrices and product representa-

tions of squares. *Combinatorica*, 28(2):163–185, 2008.

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 1–12. IEEE, 2014.

[Rao23] Anup Rao. Sunflowers: from soil to oil. *Bulletin of the American Mathematical Society*, 60(1):29–38, 2023.

[Rom06] Andrei E. Romashchenko. Reliable computations based on locally decodable codes. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, volume 3884 of *Lecture Notes in Computer Science*, pages 537–548. Springer, 2006.

[RRS17] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random CSPs below the spectral threshold. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 121–131. ACM, 2017.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[Sch08] Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 593–602. IEEE Computer Society, 2008.

[Sha90] Adi Shamir. Ip=pspace. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15. IEEE Computer Society, 1990.

[Spi19] Daniel Spielman. Spectral and algebraic graph theory. *Yale lecture notes, draft of December*, 4:47, 2019.

[SS94] Michael Sipser and Daniel A. Spielman. Expander codes. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 566–576. IEEE Computer Society, 1994.

[SS08] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 563–568. ACM, 2008.

[SS12] Warren Schudy and Maxim Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 437–446. SIAM, 2012.

[ST03] Daniel A. Spielman and Shang-Hua Teng. Smoothed Analysis (Motivation and Discrete Models). In *Algorithms and Data Structures, 8th International Workshop, WADS 2003, Ottawa, Ontario, Canada, July 30 - August 1, 2003, Proceedings*, volume 2748 of *Lecture Notes in Computer Science*, pages 256–270. Springer, 2003.

[ST11] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM*

Journal on Computing, 40(4):981–1025, 2011.

[SW19] Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2635. SIAM, 2019.

[Tei80] Luc Teirlinck. On projective and affine hyperplanes. *Journal of Combinatorial Theory, Series A*, 28(3):290–306, 1980.

[Ton99] Vladimir D Tonchev. Linear perfect codes and a characterization of the classical designs. *Designs, Codes and Cryptography*, 17:121–128, 1999.

[Ton11] Vladimir D. Tonchev. Finite geometry designs, codes, and Hamada's conjecture. In *Information security, coding theory and related combinatorics*, volume 29 of *NATO Sci. Peace Secur. Ser. D Inf. Commun. Secur.*, pages 437–448. IOS, Amsterdam, 2011.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *arXiv preprint cs/0409044*, 2004.

[Tre09] Luca Trevisan. Max cut and the smallest eigenvalue. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 263–272. ACM, 2009.

[Tro12] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, Aug 2012.

[Tro15] Joel A. Tropp. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1-2):1–230, 2015.

[WAM19] Alexander S. Wein, Ahmed El Alaoui, and Cristopher Moore. The Kikuchi Hierarchy and Tensor PCA. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1446–1468. IEEE Computer Society, 2019.

[Wit17] David Witmer. *Refutation of random constraint satisfaction problems using the sum of squares proof system*. PhD thesis, Carnegie Mellon University, 2017.

[Wol09] Ronald de Wolf. Error-correcting data structures. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 313–324. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

[Woo10] David P. Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, volume 6302 of *Lecture Notes in Computer Science*, pages 766–779. Springer, 2010.

[Wul17] Christian Wulff-Nilsen. Fully-dynamic minimum spanning forest with improved worst-case update time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1130–1143, 2017.

[Yan24] Tal Yankovitz. A stronger bound for linear 3-lcc. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

[Zou12] Anastasios Zouzias. A matrix hyperbolic cosine algorithm and applications. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2012.