COMBINATORIAL PROBLEMS IN ALGORITHMS AND COMPLEXITY THEORY

By

ADITYA POTUKUCHI

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Department of Computer Science

Written under the direction of

Swastik Kopparty

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2020

ABSTRACT OF THE DISSERTATION

Combinatorial problems in Algorithms and Complexity Theory

By ADITYA POTUKUCHI

Dissertation Director:

Swastik Kopparty

Theoretical Computer Science has connections to several areas of mathematics and one of the more prominent of these connections is to combinatorics. Indeed, many problems in this subject are often very combinatorial in nature. These problems have either used existing techniques from combinatorics or have given rise to new combinatorial techniques. This dissertation is a collection of the study of some such problems.

- We recover a result by Abbe, Shpilka, and Wigderson which states that a Reed-Muller code of rate $1 - \Theta\left(\frac{\log^r n}{n}\right)$ can be recovered from $o\left(\log^{\lfloor \frac{r-1}{2} \rfloor} n\right)$ randomly chosen errors in a stronger way. Namely, we show that the set of corrupted locations in the message can be recovered just from the *syndrome* of the message. Among the techniques are the study of tensor decomposition over finite fields and an algorithm to find the roots of a space of low degree polynomials.

- A hypergraph is r-rainbow colorable if the vertices can be colored with r colors such that every hyperegde has at least one vertex of each color. We show that it is **NP**-hard to properly 2-color a k-uniform $(k - O(\sqrt{k}))$-rainbow colorable hypergraph. In particular, we show that it is **NP**-hard to properly 2-color a 4-uniform 3-rainbow colorable hypergraph. We further extend this using a notion of *almost* rainbow colorability. We show that given a k-uniform hypergraph where there is a $(k - \sqrt{ck})$-coloring of the vertices such that every edge gets $(k - 3\sqrt{ck})$ colors, it is **NP**-hard to properly c-color it. Among the techniques are topological methods to

lower bound the chromatic number of certain hypergraphs and a theorem of Sarkaria on the chromatic number of generalized Kneser hypergraphs.

- We show that the discrepancy of a regular hypergraph can be bounded in terms of its spectral information. Let $\mathcal{H} \subset 2^{[n]}$ be a t-regular hypergraph where $|\mathcal{H}| \geq n$, and M be the $|\mathcal{H}| \times n$ incidence matrix. Define $\lambda := \max_{v \perp \overline{1}, \|v\|=1} \|Mv\|$. We show that the discrepancy of $\mathcal{H}$ is at most $O(\sqrt{t} + \lambda)$. In particular, this shows that for every t, a random t-regular hypergraph on $m \geq n$ hyperedges has discrepancy $O(\sqrt{t})$ with high probability as n grows. This bound also comes with an efficient algorithm that takes $\mathcal{H}$ as input and outputs a coloring that has the guaranteed discrepancy.

- We show that every q-ary error-correcting code of distance $1 - q^{-1} - \epsilon^2$ can be punctured to rate $\tilde{\Omega}\left(\frac{\epsilon}{\log q}\right)$ so that it is $(O_{\rho,\delta}(\epsilon^{-2}), \delta, \rho)$-list-recoverable. In particular, this shows that there are Reed-Solomon codes that are list-recoverable beyond the Johnson radius. Instantiating this for the zero-error regime immediately gives improved degree bounds for unbalanced expanders obtained from randomly punctured Reed-Solomon codes.

Acknowledgements

I am extremely thankful to my advisor Swastik Kopparty. Swastik gave me the freedom and support to explore, and has always been optimistic, and encouraging. I learnt a lot from him through many courses, meetings, e-mails, and just in general being in the same room as him.

This dissertation also contains joint works with Per Austrin, Amey Bhangale, and Ben Lund. I am extremely thankful to all of them. Working with each of them was a lot of fun and I have learnt a lot from each of them.

Some of the best parts of my grad school experience were my interactions with Jeff Kahn. Most of my knowledge, interests, and tastes in mathematics were shaped by the numerous very inspiring and enjoyable conversations I had with him. It's almost suspicious how often I would stumble into Jeff's papers in my research!

Another highlight of my grad school experience is the CS theory reading group organized by Mike Saks. Mike's near constant presence, patience, and enthusiasm made it extremely easy for me to explore topics that I would not have normally. I quickly realized that presenting a paper to Mike and the rest of the group is often the fastest way to understand it. I would like to thank everyone in the reading group for their time and enthusiasm.

I am thankful to Shubhangi Saraf and Raghu Meka for agreeing to be on my defense committee, to the graduate director Martin Farach-Colton, and to Maryann Holtsclaw, Ginger Olszewski and the rest of the department staff for help with the bureaucracy. Extra thanks to Michelle Walezak for patiently guiding me through the graduation procedure during very strange times.

I have been fortunate enough to interact with several amazing people during my stay at Rutgers. I am thankful to Bhargav Narayanan for the numerous conversations in the last three years. Bhargav taught me many clever and interesting ideas, and was often available to discuss any new papers. I am also thankful to Eric Allender, Sepehr Assadi, Pranjal Awasthi, Per Austrin, Yuval Filmus, Venkatesan Guruswami, Prahladh Harsha, Nutan Limaye, Abhishek Khetan, Pravesh Kothari, Periklis Papakonstantinou, Noga Ron-Zewi, Jaikumar Radhakrishnan, Ramprasad Saptharishi, Srikanth Srinivasan, Mario Szegedy, Avishay Tal, Mary Wootters, and many others for extremely helpful and interesting conversations. I would also like to thank Prahladh for hosting me at TIFR in the summer of '16, Yuval for hosting me at Technion in the summer of '18, and Noga for hosting me at the

University of Haifa in the summer of '19. These visits were extremely helpful and enjoyable.

I am also thankful to my friends Abhishek, Aditi, Aditya, Amartya, Cole, Deepti, Harsha, Jay, Mrinal, Surya, Tulasi, Vishwas, Vishwajeet, Zach, and others. I am especially thankful to Anurag Bishnoi for introducing me to research, and for the wonderful conversations about math, music, life, and more.

I feel extremely indebted to Abhilasha. Her support has been constant during every stage of my journey in the last few years.

Finally, I am incredibly thankful to my parents and my sister. I feel really fortunate to have had their support on every endeavor I made.

Dedication

To my parents

Table of Contents

Chapter 1
Introduction

1.1 Overview of the thesis

It is well known that the field of Theoretical Computer Science has many deep connections to topics in mathematics and statistics. The development of new tools in one field, can, and often do result in solutions to old problems in another. This dissertation is a modest step towards broadening and deepening some of these interdisciplinary connections and discovering new ones. The ultimate goal is to (I) discover new techniques that help solve interesting problems in computer science, and (II) discover problems in other fields that are amenable to techniques from computer science.

One common theme that ties the results in this dissertation together is that the motivation for all these problems come from existing problems in Computer Science. The solutions and approaches, on the other hand, involve techniques that are more combinatorial in nature. Below, the aforementioned results and the techniques underlying each are described.

1.1.1 Organization

- Chapter 2 is based on [KP18]. A brief overview of this is given in Section 1.2.
- Chapter 3 is based on [ABP20]. A brief overview of this is given in Section 1.3.
- Chapter 4 is based on [Pot19]. A brief overview of this is given in Section 1.4.
- Chapter 5 is based on [LP]. A brief overview of this is given in Section 1.5.

1.2 Syndrome decoding of Reed-Muller codes and tensor decomposition over finite fields

1.2.1 Background on Reed-Muller codes

Reed-Muller codes are one of the most studied family of error-correcting codes. Apart from the usual applications for transmission of messages/storing data, they are also studied in Mathematics, Cryptography, and Computational Complexity Theory for their properties. One of the more recent

reasons for interest in these codes is based on their ability to handle *random* errors seemingly much better than the worst-case errors. Let $n = 2^m$, the Reed-Muller code $\mathrm{RM}(m, d)$ is a subspace of $\mathbb{F}_2^n$ where the coordinates are identified with $\mathbb{F}_2^m$. Every *codeword* $c \in \mathrm{RM}(m, d)$ is identified with a polynomial p_c over $\mathbb{F}_2$ in m variables and degree at most d. For a point $x = (x_1, \ldots, x_m) \in \mathbb{F}_2^m$, we have $c[x] = p_c(x_1, \ldots, x_m)$. The Reed-Muller code $\mathrm{RM}(m, d)$ has *distance* 2^{m-d} which means that the Hamming distance between any two distinct $c_1, c_2 \in \mathrm{RM}(m, d)$ is at least 2^{m-d}. Using the fact that $\mathrm{RM}(m, d)$ is a linear space, this is just a restatement of the fact that every nonzero polynomial over $\mathbb{F}_2$ in m variables of degree at most d has at least 2^{m-d} nonzero points. Moreover, this is the truth, ie., there are degree-d polynomials with exactly 2^{m-d} nonzero points, for example, the monomial $X_1 \cdots X_d$.

Since $\mathrm{RM}(m, d)$ is a linear space, it must be the nullspace of some matrix. This is called *a* parity check matrix. One such matrix, that we call H is given as follows: H has columns indexed by elements of $\mathbb{F}_2^m$ and rows indexed by monomials of degree at most $d-1$. The entry $H(M, x)$ is given by $M(x)$, i.e., the evaluation of the monomial M evaluated at the point x.

The *rate* of the code, which is defined to be the quantity $\frac{\log |\mathrm{RM}(m,d)|}{n} = \frac{\binom{m}{\leq d}}{n}$. We will be interested in the case when d is very close to m. For the sake of this discussion, let us set $d = m - r$ where we think of r as a constant. This code has rate $1 - \Theta_r\left(\frac{m^r}{2^m}\right) = 1 - \Theta_r\left(\frac{\log^r n}{n}\right)$, and distance 2^r, and so, when one is allowed to adversarially flip 2^r (which we think of as a constant) bits of c to obtain c', then one cannot hope to recover c from c'. However, when the bits that are flipped are *randomly* chosen, the story is somewhat different.

1.2.2 Reed-Muller codes in the Binary Symmetric Channel and previous work

Let $c \in \mathrm{RM}(m, r)$ be a codeword that is passed through a *binary symmetric channel*, and let c' be the output. This just means that every coordinate of c is flipped independently with probability p. We would like to know the value of p below which one can recover c from c' with high probability. In coding theory terms, we are *uniquely decoding* the corrupted codeword. As stated, this is just a combinatorial question. Heuristically, this should at least depend on the rate of $\mathrm{RM}(m, d)$.

Abbe, Shpilka, and Wigderson [ASW15] gave quantitative bounds for flipping probability below which one can decode uniquely. Later, Saptharishi, Shpilka, and Volk [SSV17] made this result algorithmic. These results say that if the rate is $1 - \Theta_r\left(\frac{(\log n)^r}{n}\right)$, then one can uniquely decode from $o\left(\log^{\lfloor \frac{r-1}{2} \rfloor} n\right)$ randomly chosen errors. However, it is not known that one cannot uniquely decode from $o((\log n)^r)$ errors. This gap is related to whether or Reed-Muller codes 'achieve capacity' in symmetric channels, which is an extremely interesting open problem that is relevant here.

1.2.3 The main result in Chapter 2

In [KP18], we recover the bound of [ASW15] in a stronger way. Namely, we show that when one changes $o(\log^{\lfloor \frac{r-1}{2} \rfloor} n)$ randomly chosen locations in c to obtain c', the set of coordinates flipped can be recovered with high probability just from $H \cdot c'$. Here, $H \cdot c'$ is called the *syndrome* of the error. The point is that c' determines $H \cdot c'$ but $H \cdot c' = H \cdot (c' - c)$ is much smaller in size than c'. The main theorem is informally stated below:

Theorem 1A (Informal) : *Let $c \in RM(m, m-r)$ be an arbitrary codeword, and let $e \in \mathbb{F}_2^n$ be a uniformly random string with Hamming weight at most $o\left((\log n)^{\lfloor (r-1)/2 \rfloor}\right)$. There is a deterministic $(\log n)^{O(r)}$ time algorithm, which when given the syndrome $S = H \cdot e$, computes the set of nonzero coordinates of e (with high probability over the choice of e).*

We prove this theorem in a couple of ways.

Tensor decompositions over finite fields

The first way we solve the above problem is via. a connection to the classic *Tensor Decomposition problem*. For the 3 dimension case, it is stated as follows: For sets of vectors $\{a_1, \ldots, a_n\}$, $\{b_1, \ldots, b_n\}$, and $\{c_1, \ldots, c_n\}$, we are given the sum $\sum_{i \in n} a_i \otimes b_i \otimes c_i$. From this, we would like to recover the a_i', b_i's and the c_i's. We are particularly interested in conditions that ensure that the recovered vectors are unique. This problem is quite well studied in the world of Data Science, Statistics, and Machine learning, and an elegant algorithm known as 'Jennrich's algorithm' from over 30 years ago is still essentially the best algorithmic solution known for this problem. We show that the ideas here can be adapted to the finite fields setting and recover the bound of [ASW15] in an algorithmic way.

Roots of a space of polynomials

The second way we do this is by reducing the above problem to finding the set of roots of a space of low degree polynomials in m variables over $\mathbb{F}_2$. This is much closer to [SSV17], and in fact, it relies on one of the main theorems in their work. Once this reduction is complete, we then proceed to show how to solve it using random restrictions to subspaces. This is very closely related to the Valiant-Vazirani isolation lemma.

1.3 Improved inapproximability of rainbow coloring

1.3.1 Background on the complexity of hypergraph coloring

Graph and hypergraph coloring have always been fundamental problems in Combinatorics and Algorithms. These are among the most famous examples of computational problems which are *very* hard for a variety of notions of 'hardness'. A (hyper)graph $\mathcal{H} \subset 2^{[n]}$ is said to be c-colorable if there is a coloring $\chi : [n] \to [c]$ so that no edge $e \in \mathcal{H}$ has all vertices of the same color. On the algorithmic side, we only know how to color 3 colorable graphs on n vertices with n^ϵ colors where ϵ is a small positive constant less than 1. On the hardness side, only recently [BKO19], it was shown that it is **NP**-hard to 5-color a 3-colorable graph. In general, for $c \geq 4$, it is **NP**-hard to c-color a graph with $\binom{c}{\lfloor c/2 \rfloor} - 1$ colors. It is also known that for every $\epsilon > 0$ it is **NP**-hard to approximate the chromatic number to $n^{1-\epsilon}$ factor.

Slightly better hardness results are known for hypergraph coloring. For instance, it is *quasi*-**NP**-hard to color a 3-colorable 3-uniform hypergraph on n vertices with $(\log n)^{\gamma/\log\log\log n}$ colors [GHH+17]. Also, given a 2-colorable 12-uniform hypergraph, it is **NP**-hard to color it with $2^{\log^{1-o(1)} n}$ colors [KS14]. In general, the situation for hardness is much better than for graphs, however, the picture is far from complete.

1.3.2 Rainbow coloring and previous work

A hypergraph $\mathcal{H} \subset 2^{[n]}$ is said to be r-rainbow colorable if there is a coloring $\chi : [n] \to [r]$ so that every edge $e \in \mathcal{H}$ has a vertex of every color. In [AGH17], rainbow coloring was introduced in order to try and understand when coloring hypergraphs can be done efficiently. As an example, for a large (say 1000) k, given a k uniform, $r < k$ rainbow colorable graph, can one even 10 color it efficiently? Note that even a 2 coloring always exists, and efficient algorithms (for the 2-coloring) were conjectured to not exist in [BG16], and [BG17] (the conjecture in [BG17] was stronger) unless $\mathbf{P} = \mathbf{NP}$. More formally,

Conjecture [BG16] : *For $k \geq 3$, it is* **NP***-hard to find a 2-coloring of a k-uniform hypergraph that is $(k-1)$-rainbow colorable.*

The case where $k = 3$ just says that it is **NP**-hard to decide if a 3-uniform hypergraph is 2-colorable. However, for $k = 4$, it was unknown whether the above conjecture was true.

A result of Guruswami and Lee [GL15] states that for $k \geq 4$, it is **NP**-hard to c color a k-colorable $\lfloor k/2 \rfloor$-rainbow colorable hypergraph. Guruswami and Saket [GS17] show the same result even when one is guaranteed a *balanced* rainbow coloring (this is explained in more detail in Section 3.1). More

recently, Guruswami and Sandeep [GS19], building on some results from [ABP20] show that it is **NP**-hard to $\lfloor \frac{k-1}{2} \rfloor$-rainbow color a k-uniform $(k-1)$-rainbow colorable hypergraph.

1.3.3 The main results in Chapter 3

In [ABP20], we make some progress in this direction and show the following

Theorem 2A : *For $k \geq 4$ and $c \geq 2$, it is* **NP***-hard to find a* 2*-coloring of a k-uniform hypergraph that is $(k - 2\lfloor\sqrt{k}\rfloor)$-rainbow colorable.*

Doing the proof $k = 4$ gives:

Theorem 2B : *It is* **NP***-hard to find a* 2*-coloring of a* 3*-rainbow colorable* 4*-uniform hypergraph.*

The proofs are by adapting a previous result by Dinur, Regev and Smyth [DRS02]. We modify the main graph gadget used by [DRS02] (which was the so-called *Schrijver graph*) into a hypergraph gadget. This hypergraph has chromatic number at least 3, which is proved combinatorially, and is interesting in its own right.

In attempts to replace '2' above by any constant, say even '3', we make partial progress by showing hardness of a seemingly slightly harder problem. We call a hypergraph $\mathcal{H} \subseteq 2^{[n]}$ almost (p, q) rainbow colorable if there is a coloring of the vertices $\chi : [n] \to [p]$ such that every hyperedge gets at least q distinct colors. We show the following:

Theorem 2C : *For $k \geq 4$, it is* **NP***-hard to find a c-coloring of a $(k + \lfloor\sqrt{ck}\rfloor)$-uniform hypergraph that is $(k, k - 2\lfloor\sqrt{ck}\rfloor)$-almost rainbow colorable.*

Here too, the main starting point is [DRS02]. However, the hypergraph gadget we now end up with is analyzed using topological methods. More specifically, we rely on a generalization of the Borsuk-Ulam theorem to free $\mathbb{Z}_p$ actions on the sphere where p is a prime. We also require a covering bound on sets (Theorem 3.4.6) which follows from a result of Sarkaria [Sar90], also using topological methods. This led to a number of combinatorial problems which, to our best knowledge, have not been explored. Most of them have to do with understanding the aforementioned hypergraph gadget, which is a key component in the reduction, and interesting in its own right.

1.4 A spectral bound on hypergraph discrepancy

1.4.1 Background on hypegraph discrepancy

Discrepancy of hypergraphs (or set systems) is a classic problem that was the cause of several beautiful theorems, algorithms, and connections. Discrepancy is also at the heart of several algorithmic

problems, particularly in (Computational) Geometry. The classical hypergraph discrepancy setting is as follows: Let $\mathcal{H} \subseteq 2^{[n]}$ be a hypergraph, and $\chi : [n] \to \{-1, 1\}$ be a coloring. One can extend $\chi : \mathcal{H} \to \mathbb{Z}$ by defining $\chi(e) = \sum_{v \in e} \chi(v)$ for $e \in \mathcal{H}$. The *discrepancy* of $\mathcal{H}$ is defined as

$$\operatorname{disc}(\mathcal{H}) := \min_{\chi} \max_{e \in \mathcal{H}} |\chi(e)|.$$

We call $\mathcal{H}$ t-regular if every $i \in [n]$ is in exactly t edges in $\mathcal{H}$, and t-bounded if every $i \in [n]$ is in at most t edges in $\mathcal{H}$. A famous and seemingly very difficult conjecture of Beck and Fiala [BF81] states that the discrepancy of a t-bounded hypergraph is $O(\sqrt{t})$. Beck and Fiala also show [BF81] that such a hypergraph has discrepancy at most $2t - 1$. This bound has not moved much since, and the current record bound, due to Bukh [Buk16] is 'stuck at' $2t - \log^* t$ for large enough t.

1.4.2 Discrepancy of random regular hypergraphs and previous work

Motivated by the seeming difficulty of the Beck-Fiala conjecture, Ezra and Lovett, in [EL15], initiated the study of discrepancy of random t-regular hypergraphs. Let us take $\mathcal{H}_t$ to mean a random t-regular hypergraph with n vertices and m edges. Let us restrict ourselves to the case $m = \Omega(n)$, where the conjecture is also wide open. The result in [EL15] shows that $\mathcal{H}$ has discrepancy at most $O(\sqrt{t \log t})$ with high probability. Here 'with high probability' means with probability tending to 1 as t grows. Note that the Beck-Fiala conjecture is also for constant t. A recent result of Bansal and Meka gives that the discrepancy of random t-regular set systems is almost surely $O(\sqrt{t})$ provided $t = \Omega((\log \log n)^2)$.

1.4.3 The main result in Chapter 4

In [Pot19], this gap is closed. The main result implies the following:

Theorem 3A : *There is an absolute constant $C > 0$ such that the following holds: Let $\mathcal{H}_t$ be a random t-regular hypergraph on n vertices and $m \geq n$ hyperedges where $t = o(\sqrt{m})$. Then,*

$$\mathbb{P}\left(\operatorname{disc}(\mathcal{H}_t) \leq C\sqrt{t}\right) \geq 1 - o(1).$$

So in particular, this takes care of the case when t is small (fixed) as well. Moreover, this comes with an efficient algorithm to output such a coloring. The main idea in [Pot19] is to use the usual partial coloring approach [LM15] in conjunction with spectral methods. Let M be the matrix with rows induced by $\mathcal{H}$, and columns by $[n]$, and entry $(e, v) = 1$ if $v \in e$ and 0 otherwise. Define $\lambda = \lambda(M) := \max_{v \perp \bar{1}, \|v\|=1} \|Mv\|$. The main theorem in [Pot19] says the following:

Theorem 3B : *Let $\mathcal{H}$ be a t-regular hypergraph on n vertices and m edges with M. Then*

$$\mathrm{disc}(\mathcal{H}) = O\left(\sqrt{t} + \lambda(M)\right).$$

Theorem 3A then follows from the following, which is proved using the methods of Kahn and Szemerédi [FKS89].

Theorem 3C : *Let M be the incidence matrix of a random t-regular set system on n vertices, where $t = o(\sqrt{m})$, and $m \geq n$ edges. Then with probability at least $1 - n^{\Omega(1)}$,*

$$\lambda(M) = O\left(\sqrt{t}\right).$$

This gives a different direction for future progress on the Beck-Fiala conjecture, where one can bound the discrepancy in terms of 'weaker' spectral information. The reason is that some tools in discrepancy theory, such as partial coloring, are much better understood, and can be done in seemingly different ways (such as [LM15], [Rot17], [LRR17]). Thus one can hope for some sort of control over this procedure.

1.5 List recovery of randomly punctured codes

1.5.1 Background on list recovery

We say that a code $\mathcal{C} \subset [q]^n$ is (ℓ, δ, ρ) list recoverable if, for every collection of sets $\{L_i \subseteq [q]\}_{i \in [n]}$ with $|L_i| \leq \ell$ for each i, we have

$$|\{c \in \mathcal{C} \mid \Delta(c, L_1 \times \cdots \times L_n) \leq \rho n\}| \leq \ell(1+\delta)$$

A well known result is that every code of good enough distance has good list-recovery properties. This is usually called the *Johnson bound* for list recovery.

Johnson bound for list recovery : *Every q-ary code of relative distance ρ is $\left(\ell, \frac{\rho}{1-\ell(1-\rho)} - 1\right)$-zero-error list recoverable.*

One extremely interesting family of codes is the degree-d Reed-Solomon codes for a fixed degree d. The codewords of the degree-d Reed-Solomon code over $\mathbb{F}_q$ with evaluation set $S \in \binom{[q]}{m}$ are the evaluations of all univariate polynomials of degree at most d on elements of S. In other words, suppose $S = \{s_1, \ldots, s_m\}$, the degree-d Reed-Solomon code on S is the set $\{(p(s_1), \ldots, p(s_m)) \mid \deg(p) \leq d\}$.

The degree-d Reed-Solomon codes with evaluation set of size n have distance $n-d$. Taking $n=q$, i.e., using $\mathbb{F}_q$ as the evaluation set, this shows that there are Reed-Solomon codes of rate ϵ^2 that are $(O_{\delta,\rho}(\epsilon^2),\delta,\rho)$-list recoverable.

1.5.2 The main result of Chapter 5

The main result in [LP] is the following:

Theorem 4A (Informal) : *Every code $\mathcal{C}\subset[q]^n$ with distance at least $n(1-q^{-1}-\epsilon^2)$ can be punctured to rate $\Omega\left(\frac{\epsilon}{\log q}\right)$ so that it is $(O_{\delta,\rho}(\epsilon^{-2}),\delta,\rho)$-list recoverable.*

In particular, this shows that there are codes that are list recoverable *beyond* the Johnson Bound.

1.5.3 Motivation for the result

The main motivation for this result is the existence of puncturings of Reed-Solomon codes that give unbalanced expanders. Given a q-ary code $\mathcal{C}\subseteq[q]^n$, one can construct a bipartite graph $G(\mathcal{C})$ on vertex $\mathcal{C}\sqcup([n]\times[q])$. Every $c=(c_1,\ldots,c_n)$ has the set $\{(i,c_i)\}_{i\in[n]}$ as neighbors. This graph is called a (k,ϵ)-unbalanced expander if for every set $\mathcal{C}'\subseteq\mathcal{C}$ of size at most k, we have that $|N(\mathcal{C}')|\geq kn(1-\epsilon)$. The following is an old question in Complexity theory that is attributed to Guruswami [Gur], which is also explicitly mentioned in [CZ18].

Question : *Let $\mathcal{C}_S$ be the degree-d Reed-Solomon code on evaluation set S. What is the smallest m such that when S is chosen uniformly at random, $G(\mathcal{C}_S)$ is, with high probability, a $(o(q),o(1))$-unbalanced expander?*

This is a very interesting problem, and as far as we know, only the almost trivial $m=O(q)$ was known. Theorem 4A implies that $m=\tilde{O}(\sqrt{q})$.

Chapter 2

Syndrome decoding of Reed-Muller codes and tensor decomposition over finite fields

Reed-Muller codes are some of the oldest and most widely studied error-correcting codes, of interest for both their algebraic structure as well as their many algorithmic properties. A beautiful result of Saptharishi, Shpilka and Volk [SSV17] (building on Abbe, Shpilka and Wigderson [ASW15]) showed that for binary Reed-Muller codes of length n and distance $d = O(1)$, there is a $\mathrm{poly}(n)$-time algorithm[1] that can correct $\mathrm{poly}\log(n)$ random errors (which is well beyond the worst-case error tolerance of $d/2 = O(1)$ errors). In this paper, we show that the $\mathrm{poly}\log(n)$ random error locations can in fact be computed in $\underline{\mathrm{poly}\log(n)\text{ time}}$ given the *syndrome vector* of the received word. In particular, our main result shows that there is a $\mathrm{poly}(n)$-time, $\mathrm{poly}\log(n)$-space algorithm that can compute the error-locations.[2]

Syndrome decoding of Reed-Muller codes turns out to be equivalent to a basic problem about tensor decompositions over finite fields. We give two algorithms for our main result, one coming from the Reed-Muller code world (and based on [SSV17]), and another coming from the tensor-decomposition world (and based on algorithms for tensor decompositions over the real numbers).

2.1 Introduction

A binary error-correcting code is simply a subset $\mathcal{C} \subseteq \mathbb{F}_2^n$. We say the code $\mathcal{C}$ has minimum distance $\geq d$ if for any distinct $c_1, c_2 \in \mathcal{C}$, the Hamming distance $\Delta(c_1, c_2) \geq d$. The main nontrivial algorithmic task associated with an error-correcting code $\mathcal{C}$ is *decoding*: for a codeword c and a sparse error-vector e, if we are given the "received word" $y = c + e$, we would like to compute the original codeword c.

A *linear code* $\mathcal{C}$ is a code which is also an $\mathbb{F}_2$-linear subspace of $\mathbb{F}_2^n$. Let k denote the dimension of the code, and let $k' = n - k$. Linear codes are usually specified either by giving a *generating matrix* G (whose rows span $\mathcal{C}$) or an $k' \times n$ *parity-check matrix* H (whose rows span the orthogonal space

[1] In fact, the algorithm of [SSV17] runs in near linear time $n\,\mathrm{poly}\log(n)$.

[2] This algorithm is in fact a one-pass streaming algorithm which spends $\mathrm{poly}\log(n)$-time per coordinate as it scans the received word, and at the end of the pass it computes the error-locations in time $\mathrm{poly}\log(n)$.

$\mathcal{C}^{\perp}$). Given a received word $y = c + e$, where c is a codeword and e is a sparse vector, the *syndrome* of y is simply the vector $S \in \mathbb{F}_2^{k'}$ given by:

$$S = H \cdot y = H \cdot (c + e) = 0 + H \cdot e = H \cdot e.$$

Observe that the syndrome can easily be computed from the received word. An important fact here is that the syndrome is exclusively a function of e, and does not depend on c. Given the syndrome $S = H \cdot y$ (where $y = c + e$ for a codeword c and a sparse error vector e), the algorithmic problem of *syndrome decoding* is to compute the error vector e. Clearly, a syndrome decoding algorithm can also be used for standard decoding: given a received word y we can compute the syndrome $H \cdot y$, and then apply a syndrome decoding algorithm to it.

Reed-Muller codes are algebraic error-correcting codes based on polynomial evaluation [Ree54, Mul54]. Here we focus on Reed-Muller codes over $\mathbb{F}_2$ with constant distance (although our results apply to larger fields and larger distances too). Let m be a large integer, and let $r = O(1)$ be an integer. Associated to these parameters, the Reed-Muller code $RM(m, m-r)$ is defined as follows. The coordinates of the code correspond to the points of $\mathbb{F}_2^m$ (and thus the length $n = 2^m$. To each polynomial $P(X_1, \ldots, X_m)$ of individual degree ≤ 1 and total degree $\leq m - r$, we associate a codword in $RM(m, m-r)$: this codeword is given by evaluations of P at all the points of $\mathbb{F}_2^m$. This code has codimension $\Theta(m^r) = \Theta((\log n)^r)$ and minimum distance $d = 2^r = \Theta(1)$.

Decoding algorithms for Reed-Muller codes have a long history. It has been known for a long time that one can decode from $d/2$ worst case errors in polynomial time (recall that d is the distance of the code). There has been much work on decoding these codes under random errors [Dum17] and the local testing, local decoding and local list-decoding of these codes [BLR93, RS96, GL89, AS03, STV01, AKK$^+$05, BKS$^+$10].

A beautiful and surprising result of Saptharishi, Shpilka and Volk [SSV17] (building on Shpilka Abbe, Shpilka and Wigderson [ASW15], Kumar and Pfister [KP15], and Kudekar et.al. [KMSU15]) gave new insights into the error-correction capabilities of Reed-Muller codes under random errors. In the constant distance regime, their results showed that the above Reed-Muller codes $RM(m, m-r)$ (with codimension $\Theta\left((\log n)^r\right)$ and distance $O(1)$ can in fact be decoded in poly(n) time from $\Theta\left((\log n)^{\lfloor (r-1)/2 \rfloor}\right)$ random errors with high probability (which is well beyond the worst-case error-correction radius of $O(1)$).

The main result in this chapter is a syndrome decoding version of the above.

Theorem A (Informal) : *Let $c \in RM(m, m-r)$ be an arbitrary codeword, and let $e \in \mathbb{F}_2^n$ be a uniformly random string with Hamming weight at most $o\left((\log n)^{\lfloor (r-1)/2 \rfloor}\right)$. There is a deterministic $(\log n)^{O(r)}$ time algorithm, which when given the syndrome $S = H \cdot e$, computes the set of nonzero*

coordinates of e (with high probability over the choice of e).

As an immediate corollary, there is a streaming algorithm for computing the error-locations in the above setting, which makes one pass over y, uses only $\operatorname{poly}\log(n)$ space, and spends only $\operatorname{poly}\log(n)$ time per coordinate. Indeed, the syndrome $H \cdot y$ (where H is parity check matrix of Reed-Muller codes) can be easily computed in one pass over y (using the $\operatorname{poly}\log(n)$ space and $\operatorname{poly}\log(n)$ time per coordinate), after which the syndrome decoding algorithm of Theorem A can compute the nonzero coordinates of e.

2.1.1 Techniques

We give two proofs of our main result. The first goes via a connection to the problem of tensor-decomposition of random low-rank tensors over finite fields. We give an efficient algorithm for this tensor-decomposition problem, by adapting a known algorithm (due to Jennrich) for the analogous problem over the real numbers. The second goes via the original approach of [SSV17], which is a novel variant of the Berlekamp-Welch decoding algorithm for Reed-Solomon codes. We show how to implement their steps in a compact form; an important technical step in this is a new algorithm to solve certain systems of polynomial equations, using ideas related to the Valiant-Vazirani isolation lemma.

Approach via tensor decomposition

It will be useful to understand how a parity-check matrix H of the Reed-Muller code $RM(m, m - 2r - 2)$ looks. Recall that H is a $k' \times n$ matrix (where k' is the codimension of $RM(m, m - 2r - 2)$ in $\mathbb{F}_2^n$). The rows of H are indexed by elements of $\mathbb{F}_2^m$, and for $x \in \mathbb{F}_2^m$, the x-column of H turns out to (essentially) equal $x^{\otimes \leq 2r+1}$, the $\leq 2r + 1$'th tensor powers of x. Thus for a random sparse vector e whose nonzero coordinates are $E \subseteq \mathbb{F}_2^m$, the syndrome $S = H \cdot e$ ends up equalling:

$$S = \sum_{e \in E} e^{\otimes \leq 2r+1}.$$

Having written the problem in this way, the problem of computing the error locations E from the syndrome S is basically just the problem of tensor decomposition of an appropriately constructed random low rank tensor over $\mathbb{F}_2$.

We show how this problem can be solved efficiently. We adapt an elegant algorithm of Jennrich for this task over the real numbers. This algorithm is based on taking two random flattenings of the tensor S into matrices, using properties of the pseudoinverse (a.k.a. the Moore-Penrose generalized inverse) of a singular matrix, and spectral ideas. Two difficulties show up over finite fields. The

more serious one is that the Moore-Penrose generalized inverse does not exist in general over finite fields [Rao03] (and even in our special situation). We overcome this by developing an alternate algorithm that does not use the pseudoinverse of a singular matrix, but instead keeps track of a full rank minor of the singular matrix. The other difficulty is that small finite fields do not have enough elements in them for a matrix to have all distinct eigenvalues in the field. We overcome this by moving to a large enough extension field $\mathbb{F}_{2^{10m}}$.

Finally we note that this gives a new proof of the main theorem of [SSV17]. The details appear in Section 2.4. There we also mention how to derandomize this algorithm.

Approach via solving polynomial equations

The original approach of [SSV17] works as follows. Given the received word $y \in \mathbb{F}_2^n$, we view it as a function from $F_2^m \to \mathbb{F}_2$. We then look for all polynomials $A(X_1, \ldots, X_m), B(X_1, \ldots, X_m)$ of degree at most $r+1$, $m-r-1$ respectively, such that for all $x \in \mathbb{F}_2^m$:

$$A(x) \cdot y(x) = B(x).$$

[SSV17] suggested to consider the linear space V of *all* $A(X_1, \ldots, X_m)$ for which there exists such a $B(X_1, \ldots, X_m)$[3]. The main property they show is that for E is completely characterized by V; namely, E is precisely the set of common zeroes of all the elements of V. Then [SSV17] simply check for each point $x \in \mathbb{F}_2^m$ whether it is a common zero of all elements of V.

Our syndrome decoder tries to do the same, in poly(m) time instead of poly(2^m) time, using only the syndrome. We begin by observing that a basis for the space V can be found given only the syndrome of y. This reduces us to the problem of finding the common zeroes of the collection of polynomials in V.

In full generality, given a collection of low degree polynomial finding their common solutions is NP-hard. Indeed, this very easily encodes SAT. However our situation is different in a subtle but important way. It turns out that V is the space of *all* low degree polynomials that vanish on E. So we are not solving an arbitrary system of polynomial equations! The next theorem says that such systems of polynomial equations are solvable efficiently.

Theorem B (Informal): *Let $E \subseteq \mathbb{F}_2^m$ be a uniformly random subset of size $o(m^r)$. Let V be the space of all polynomials of degree at most $r+1$ which vanish on E. There is a deterministic polynomial time algorithm that, when given a basis for V as input, computes E (with high probability over the choice of E).*

[3]This idea of considering *all* solutions of this "error-locating equation" instead of just one solution is the radical new twist over the Berlekamp-Welch algorithm that makes [SSV17] so powerful.

Our algorithm for this problem uses ideas related to the Valiant-Vazirani isolation lemma (which reduces SAT to Unique-SAT). If E turned out to be of size exactly 1, it turns out that there is a very simple way to read off the element of E from V. We show how to reduce the general case to this case: by choosing a random affine subspace G of a suitable small codimension c, we can ensure that $|E \cap G| = 1$. It also turns out that when E is random, given the space of all m-variate polynomials of degree at most $r+1$ vanishing on E, we can compute the space of all $m-c$-variate polynomials (viewing G as $\mathbb{F}_2^{m-c}$) of degree at most $r+1$ vanishing on $G \cap E$. This lets us reduce to the case of a unique solution, and we can recover an element of E. Repeating this several times gives us all elements of E.

We also give a different algorithm for Theorem B using similar ideas, which has the advantage of being deterministic. The key subroutine for this algorithm is that given an affine subspace $H \subseteq \mathbb{F}_2^m$, we can compute the size of $E \cap H$ from V (for this it is important that E is random). This subroutine then easily allows us to zoom in on the elements of E.

2.2 Notation

We give some notation that will be used throughout this chapter.

- We say that $a = b \pm c$ to mean $a \in [b-c, b+c]$.
- We use ω to denote the exponent of matrix multiplication.
- For a matrix $M_{m \times n}$, and subsets $A \subseteq [m]$ and $B \subseteq [n]$, we say $M_{A,B}$ to mean to submatrix of M with rows and columns indexed by elements in A and B respectively. Further, $M_{A,\cdot} := M_{A,[n]}$, and $M_{\cdot,B} := M_{[m],B}$.
- We use $\mathcal{M}_r^n$ to denote the set of all monomials of degree $\leq r$ in n variables $X_1, \ldots, X_n$.
- For a vector $v \in \mathbb{F}_2^m$, let us write $v^{\otimes \leq t}$ to mean the vector of length $\binom{m}{\leq t}$, whose entries are indexed by the monomials in $\mathcal{M}_t^m$. The entry corresponding to $M \in \mathcal{M}_r^m$ is given by $M(v)$.
- For a set of points $A \subseteq \mathbb{F}_2^n$, we use $A^{\otimes \leq t} := \{v^{\otimes \leq t} \mid v \in A\}$.
- A set of points $A \subseteq \mathbb{F}_2^n$ is said to satisfy property U_r if the vectors in $A^{\otimes \leq r}$ are linearly independent.
- For a set $A \subseteq \mathbb{F}_2^t$, we denote $\mathrm{mat}(A)$ to be the $|A| \times t$ matrix whose rows are elements of A.

2.3 The Main Result

The main result is that we show how to decode high-rate Reed Muller codes $RM(m, m-2r-2)$, where we think of r as growing very slowly compared to m, say, a constant. In this case, the received corrupted codeword is of length $n = 2^m$. However, *syndrome* of this code word is $O(m^{2r})$. We want to find the set of error locations from the syndrome itself *efficiently*. Formally, we prove the following:

Theorem 2.3.1. *Let E be a set of points in $\mathbb{F}_2^m$ that satisfy property U_r. There is a randomized algorithm that takes as input, the syndrome of an $RM(m, m-2r-2)$ codeword corrupted at points in E, and returns the list of error locations E with probability $> .99$. This algorithm runs in time $O(m^{\omega r+4})$.*

Our first proof of this theorem is via the 'Tensor Decomposition Problem' over small finite fields. As the name suggests, this is just the finite field analogue of the well-studied Tensor Decomposition problem (see, for example, [McC87]). The problem is (equivalently) stated as follows: Vectors $e_1, \ldots, e_t$ are picked uniformly and independently from $\mathbb{F}_2^m$. We are given access to

$$\sum_{i \in [t]} e_i^{\otimes \leq 2r+1},$$

and the goal is to recover e_i's. The fact that the e_i's are picked randomly is extremely important, as otherwise, the e_i's can be picked so that the decomposition is not unique. We rely on the results from [ASW15], [KMSU15] and [KP15], which informally state that the Reed-Muller codes achieve capacity in the Binary Erasure Channel (BEC) in the very high rate regime, entire constant rate regime, and the very low rate regime. More precisely, for $RM(m, d)$ when the degree d of the polynomials is $o(m)$, $m/2 \pm O(\sqrt{m})$, $m - o(\sqrt{m/\log m})$. This means that when a set of points are picked independently with probability p, where $p = 1 - R - \epsilon$, where R is the rate of the code, and ϵ is a small constant, these points satisfy property U_r with high probability for this range of R.

Since this is a tensor decomposition problem, one natural approach is to try and adapt existing tensor decomposition algorithms. Assuming only that the $e_i^{\otimes \leq r}$'s are linearly independent, we show how to decompose $\sum_{i \in [t]} e_i^{\otimes \leq 2r+1}$. Indeed, this is a very well studied problem in the machine learning community, and one can adapt existing techniques with a bit of extra work. The advantage of this approach is the simplicity and its ability to give the proof the main result of [SSV17] by giving an efficient algorithm.

Our second approach to solving this problem in finite fields is to reduce it to finding the common zeroes of a space of low degree polynomials, which we then proceed to solve. This algorithm goes via an interesting and natural algebraic route involving solving systems of polynomial equations.

The running time of the resulting algorithm has a worse dependence on the field size that the first approach. We note here that this alsogives a new approach to tensor decomposition, using ideas related to the Berlekamp-Welch algorithm.

2.4 Proof of Theorem 2.3.1 using Jennrich's Algorithm

The key idea is that, we will look the vector $v^{\otimes \leq 2r+1}$ as a 3-tensor $v^{\otimes \leq r} \otimes v^{\otimes \leq r} \otimes v^{\otimes \leq 1}$. Indeed, given the syndrome $\sum_{i \in [t]} e_i^{\otimes \leq 2r+1}$, one can easily construct the 3-dimensional tensor $\sum_{i \in [t]} e_i^{\otimes \leq r} \otimes e_i^{\otimes \leq r} \otimes e_i^{\otimes \leq 1}$, so we may assume that we are given the tensor. This allows to use techniques inspired by existing tensor decomposition algorithms [Har70, LRA93] like Jennrich's Algorithm (see [Blu15]). To our best knowledge, this problem has not been previously studied over finite fields. Although we only need the result for codes over $\mathbb{F}_2$, the proof works almost verbatim over other fields.

2.4.1 An overview and analysis of the algorithm

We first restate the problem:

> *Input:* For a set of vectors $E = \{e_1, \ldots, e_t\} \subset \mathbb{F}_2^m$ that satisfy property U_r, we are given the syndrome as a 3-tensor
> $$S = \sum_{i \in [t]} e_i^{\leq r} \otimes e_i^{\otimes \leq r} \otimes e_i^{\leq 1}.$$
> *Output:* Recover the e_i's

Following in the footsteps of Jennrich's Algorithm, we pick random points in a and b in $\mathbb{F}_{2^{10m}}^{m+1}$ and compute the matrices

$$S^a := \sum_{i \in [t]} \langle a, e_i^{\otimes \leq 1} \rangle e_i^{\otimes \leq r} \otimes e_i^{\otimes \leq r},$$

and

$$S^b := \sum_{i \in [t]} \langle b, e_i^{\otimes \leq 1} \rangle e_i^{\otimes \leq r} \otimes e_i^{\otimes \leq r}.$$

Computing these matrices is the same as taking the weighted linear combination of the slices of the tensor T along one of its axes. Define the $t \times \binom{m}{\leq r}$ matrix $X := \mathrm{mat}(E^{\otimes \leq r})^T$, so we have the matrices $S^a = XAX^T$, and $S^b = XBX^T$ for diagonal matrices A and B respectively. The i'th diagonal entry of A is given by $a_i := \langle a, e_i^{\otimes \leq 1} \rangle$, and the i'th diagonal entry of B is given by $\langle b, e_i^{\otimes \leq 1} \rangle$.

Let K and L be two (not necessarily distinct) maximal linearly independent sets of $t(= |E|)$ rows in X. Denote $X_K := X_{K,\cdot}$, and $X_L := X_{L,\cdot}$ as shorthand. We have that $S^a_{K,L} := X_K A X_L^T$, and $S^b_{K,L} := X_K B X_L^T$ are full rank, since the diagonal entries of A and B are all distinct and nonzero. Therefore, we have the inverse $(S^b_{K,L})^{-1} = (X_L^T)^{-1} B^{-1} X_K^{-1}$. Multiplying with S^a_L, we have:

$$\begin{aligned} S^a_{K,L}(S^b_{K,L})^{-1} &= X_K A X_L^T (X_L^T)^{-1} B^{-1} X_K^{-1} \\ &= X_K (AB^{-1}) X_K^{-1}. \end{aligned}$$

In order to carry out the operations over an extension field, we need to pick an irreducible polynomial of appropriate degree over $\mathbb{F}_2$. Fortunately, this can also be done in time poly(m). The reason that a, and b are chosen from a large extension field is that it ensures that all the entries of AB^{-1} are also nonzero and distinct w.h.p. So, the columns of X_K are just the eigenvectors of this matrix, which we will then proceed to compute. In order to compute the eigenvalues, we need to factor the characteristic polynomial. Here one can use Berlekamp's factoring algorithm [Ber67]. So, we require the following two lemmas:

Lemma 2.4.1. *For nonzero and distinct $x_1, \ldots, x_t \in \mathbb{F}_2^m$, and a uniformly chosen a and b from $\mathbb{F}_{2^{10m}}^{m+1}$, denote $a_i := \{\langle a, x_i^{\otimes \leq 1}\rangle$, and $b_i := \langle a, x_i^{\otimes \leq 1}\rangle\}$. Then we have that w.h.p,*

(1) *$a_1, \ldots, a_t, b_1, \ldots, b_t$ are all distinct and nonzero.*

(2) *$\nexists i, j \in [t]$ such that $i \neq j$ and $a_i b_i^{-1} = a_j b_j^{-1}$.*

Proof. For the proof of (1), we just need to say that there is no subset $S \subseteq [n]$ such that $\langle a, \mathbb{1}_S \rangle = 0$, or equivalently, there are no nontrivial linearly dependencies in the entries of a. Since we picked a and b from a vector space over a large enough field, there are at most $2^{2(m+1)}$ possibilities for nontrivial linear dependencies, and each occurs with probability $\frac{1}{2^{10m}}$. Therefore, there are nontrivial linear dependencies with probability at most 2^{-7m}.

To prove (2), first fix i and j. W.L.O.G, let k be a coordinate where $x_i^{\otimes \leq 1}$ is 1 and $x_j^{\otimes \leq 1}$ is zero. Fixing a, and all but the k'th coordinate of b we see that there is exactly one $b[k]$ such that $a_i b_i^{-1} = a_j b_j^{-1}$. Therefore, with a, and b picked uniformly, this equation is satisfied with probability at most $\frac{1}{2^{10m}}$. Therefore, there are i and j that satisfy this equality with probability at most $\frac{m^2}{2^{10m}}$.

Therefore, by union bound, both (1), and (2) are satisfied with probability at least $1 - 2^{-6m}$.

□ □

Lemma 2.4.2. *With X, A, B as given above, the only eigenvectors of $X_K A B^{-1} X_K^{-1}$ are the columns of X_K with probability at least $1 - 2^{-7m}$.*

Proof. Indeed, the columns of X_K are eigenvectors of $X_K AB^{-1} X_K^{-1}$ since it is easy to verify that $(X_K AB^{-1} X_K^{-1}) X_K = X_K (AB^{-1})$. Moreover, by Lemma 2.4.1, with probability at least $1 - 2^{-7n}$, the matrix AB^{-1} has distinct nonzero diagonal entries. Therefore, all the eigenvalues are distinct, and so no other vector in the span of the columns if X_K is an eigenvector.

□ □

Remark: In the traditional Jennrich's Algorithm, after defining the matrices S^a, and S^b, one usually works with the *pseudo-inverse* of S^b. It turns out that a necessary condition for the pseudo inverse to exist is that $\text{rank}(S^b) = \text{rank}(S^b(S^b)^T) = \text{rank}((S^b)^T S^b)$. However, this need not be the case for us, in fact, one can have that X is full rank, yet, $X^T X = 0$, which gives $S^b(S^b)^T = 0$, while S_b still has full rank. Hence, we needed to find a full rank square submatrix of X and use it to determine the rest of X.

To recover the rest of X, we make use of the entries $(S_{l,i,1})_{l \in L}$ of S. We may assume that the entries of $e_i^{\otimes \leq 1}$ are labelled such that $e_i^{\otimes \leq 1}[1] = 1$, and so $S_{\cdot,\cdot,1} = XX^T$. Now suppose we want to recover row i, we set up a system of linear equation in the variables $x[i] = (x[i,1], \ldots, x[i,t])$:

$$X_L \cdot x[i]^T = S^T_{i,L,1}.$$

This can be solved since X_L is full rank, and therefore, is invertible.

2.4.2 The algorithm and running time

Given the analysis above, we can now state the algorithm:

---

procedure JennrichFF(S)
  $a, b \sim \mathbb{F}_{2^{10m}}^{m+1}$
  $S^a \leftarrow \sum_{i \in [n+1]} S_{\cdot,\cdot,i} a[i]$
  $S^b \leftarrow \sum_{i \in [n+1]} S_{\cdot,\cdot,i} b[i]$
  $K, L \leftarrow$ indices of the largest full rank submatrix of S^a
  $v_1, \ldots, v_t \leftarrow$ eigenvectors of $S^a (S^b)^{-1}$
  $X_K \leftarrow (v_1, \ldots, v_t)$
  initialize matrix X
  for $1 \leq i \leq \binom{m}{\leq r}$ **do**
    $X^T_{i,\cdot} \leftarrow X_K^{-1} S^T_{K,i,1}$
  end for
  return X
end procedure

There are several steps in this algorithm. We will state the running time of each step

0. Constructing the 3-dimensional tensor from the syndrome takes time $O(m^{2r+3})$
1. Finding an irreducible polynomial of degree $10m$ takes time $O(m^4)$ (see, for example, [Sho94]). This is for constructing $\mathbb{F}_{2^{10m}}^{m+1}$.
2. Constructing S^a, and S^b takes time $O\left(\binom{m}{\leq 2r+1}\right)$.
3. Computing K,L takes time $O\left(\binom{m}{\leq r}^{\omega}\right)$.
4. Inverting X_K takes time at most $O\left(t^{\omega}\right)$.
5. Recovering X takes time $O\left(t^2\binom{m}{\leq r}\right)$. In fact, recovering just the relevant coordinates of X takes time just $O(t^2 m)$.
6. Factoring degree $\binom{m}{\leq r}$ polynomials over $\mathbb{F}_{2^{10m}}$ takes $O(m^{r+4})$ time. This is required for computing eigenvectors.

Therefore, the whole algorithm runs in time $O(m^{\omega r+1})$, compared to the input size of $O(m^{2r})$.

2.4.3 A note on derandomization

We needed to pick a and b at random to ensure that all the a_i's and b_i's satisfy the conditions given in Lemma 2.4.1. In order to ensure this deterministically, set the vectors of polynomials

$$a(\alpha) = (1, \alpha, \alpha^2, \ldots, \alpha^m)$$
$$b(\alpha) = (\alpha^{3m}, \alpha^{3m+2}, \ldots, \alpha^{5m})$$

For any $x_i, x_j \in \mathbb{F}_2^{m+1} \setminus \{0\}$, where $x_i \neq x_j$, it is easy to see that the polynomials

$$\begin{aligned}
\langle a(\alpha), x_i\rangle,\\
\langle b(\alpha), x_i\rangle,\\
\langle a(\alpha), x_i\rangle - \langle a(\alpha), x_j\rangle,\\
\langle a(\alpha), x_i\rangle - \langle b(\alpha), x_j\rangle,\\
\langle b(\alpha), x_i\rangle - \langle b(\alpha), x_j\rangle, \qquad \text{and}\\
\langle a(\alpha), x_i\rangle\langle b(\alpha), x_j\rangle - \langle a(\alpha), x_j\rangle\langle b(\alpha), x_i\rangle
\end{aligned}$$

are all nonzero polynomials of degree at most $6m$ in α. In Claim 2.4.3, we prove this for the last polynomial, the others are trivial. Taking α to be some primitive element of the field $\mathbb{F}_{2^{10m}}$ ensures

that is it not a root of any of the above polynomials. Such an element can also be efficiently and deterministically found (see [Sho94]).

Claim 2.4.3. *For two distinct nonzero elements $x_i.x_j \in \mathbb{F}_2^{m+1}$, the polynomial*

$$P(\alpha) = \langle a(\alpha), x_i\rangle\langle b(\alpha), x_j\rangle - \langle a(\alpha), x_j\rangle\langle b(\alpha), x_i\rangle$$

is nonzero.

Proof. W.L.O.G, let $x_i > x_j$ lexicographically. Let u be the largest index such that $x_i[u] = 1$, and $x_j[u] = 0$, and let v be the largest index such that $x_j[v] = 1$, so x_i and x_j agree on all coordinates indexed higher than u. We claim that monomial $\alpha^{3m+2u+v-3}$ in $P(\alpha)$ survives, and therefore P is nonzero. Indeed, this is true since it is easy to see that if this monomial has to be cancelled out, it has to be equal to some $\alpha^{3m+2u'+v'-3}$, where $x_i[u'] = x_j[v'] = 1$ and where $u < u', v' < v$. But by our assumption, x_i and x_j agree on every coordinate indexed higher than u, and therefore, $x_i[v'] = x_j[u'] = 1$, and therefore, the monomial $\alpha^{3m+2u'+v'-3}$ is computed an even number of additional times.

□ □

2.5 Proof of Theorem 2.3.1 by reducting to common zeroes of a space of polynomials

In this section, we prove Theorem 2.3.1 via finding common roots to a space of low degree polynomials. There are two components to this, the first is a reduction to an algebraic problem:

Theorem 2.5.1. *Let y be a corrupted codeword from $RM(m, m-2r-2)$, with error locations at $E \subseteq \mathbb{F}_2^m$. There is an algorithm,* SpaceRoots*, that runs in time $O(m^{(r+1)\omega})$ that takes the syndrome of y as input, and returns the space of all reduced polynomials of degree $\leq r+1$ that vanish on E.*

We are now left with the following neat problem, which is interesting in its own right, namely, finding the roots of a space of low degree polynomials:

Theorem 2.5.2. *For a set of points $E \subseteq \mathbb{F}_2^m$ that satisfy property U_r, given the space V of all reduced polynomials of degree $\leq r+1$ that vanish on E, there is an algorithm,*FindRoots*, that runs in time m^{2r} , and returns the set E with probability $1 - o(1)$.*

The rest of this section is will be dedicated to proving Theorem 2.5.1, and setting up the stage for Theorem 2.5.2.

We set up more notation that will be continue to be used in the paper: For a vector $v \in \mathbb{F}_2^{2^m}$, we will treat v as a function from $\mathbb{F}_2^m$ to $\mathbb{F}_2$ and vice versa in the natural way, i.e., for a point $x \in \mathbb{F}_2^m$, $v(x)$ is the coordinate corresponding to point x in v.

We shall use the following theorem from [SSV17] that completely characterizes the space of polynomials V that we are looking for:

Theorem 2.5.3. *For a set of points E satisfying property U_r, let y be the codeword from $RM(m, m-2r-2)$ which is flipped at points in E. Then, there exists nontrivial polynomials A, and B that satisfy:*

$$A(x) \cdot y(x) = B(x) \ \forall x \in \mathbb{F}_2^m,$$

where $deg(A) \leq r+1$ and $deg(B) \leq m-r-1$. Moreover, E is the set of common zeroes of all such A's which satisfy the equations. Furthermore, for every polynomial A that vanishes on all points of E, there is a B such that the above equation is satisfied.

So, the way we prove Theorem 2.5.1 is by finding polynomials $A(X_1, \ldots, X_m)$ of degree $\leq r+1$ such that $A \cdot y$ is a polynomial of degree $\leq m-r-1$. Most important, we would like to find this space V of polynomials *efficiently*, i.e., in time poly(m^{2r}). For this, we set up a system of linear equations and solve for A.

Proof of Theorem 2.5.1. Let us use s to denote the syndrome vector, whose entries are indexed by monomials of degree at most $2r+1$. Let us denote

$$A = \sum_{M \in \mathcal{M}_{r+1}^m} a_M M(X_1, \ldots, X_m)$$

to be a polynomial whose coefficients are indeterminates a_M for $M \in \mathcal{M}_r^m$. We want that $A \cdot y$ is a polynomial of degree $\leq m-r-1$, so we look at it as a codeword of $RM(m, m-r-1)$. Using the fact that the dual code of $RM(m, m-r-1)$ is $RM(m, r)$, we have, for any monomial $M' \in \mathcal{M}_r^m$,

$$\begin{aligned} 0 &= \sum_{x \in \mathbb{F}_2^m} A(x) y(x) M'(x) \\ &= \sum_{M \in \mathcal{M}_{r+1}^m} a_M \sum_{x \in \mathbb{F}_2^n} y(x) M(x) M'(x) \\ &= \sum_{M \in \mathcal{M}_{r+1}^m} a_M s_{M \cdot M'}. \end{aligned}$$

Hence, the solution space to this system of $|\mathcal{M}_r^m|$ equations in $|\mathcal{M}_{r+1}^m|$ variables gives us the space V of all polynomials of degree $\leq r+1$ that vanish on all points of E. Moreover, we can do this efficiently in time $O(m^{(2r+1)\omega})$ using gaussian elimination.

□ □

Once we have the above result, we can give a proof of Theorem 2.3.1 assuming Theorem 2.5.2.

---

procedure SyndromeDecode(S)
  $V \leftarrow$ SpaceRoots(S)
  return FindRoots(V)
end procedure

Of course, assuming we have an algorithm such as FindRoots, it is obvious that the above algorithm is, indeed what we are looking for. Most of the rest of the paper goes into finding such an algorithm.

2.6 Efficiently finding roots of a space of polynomials

At this point, we are left with the following neat problem:

> *Input:* Given the space V of all degree $\leq r+1$ polynomials which vanish on a set of points E satisfying property U_r.
>
> *Output:* Recover E.

2.6.1 A sketch of the rest of the algorithm

Let us denote $t = |E|$. The main idea in the rest of the algorithm is to restrict the set of points to only those lying on a randomly chosen affine subspace of codimension $\sim \log t$. The hope is that exactly one point in E lies in this subspace. This happens with constant probability, and in fact, for every point $e \in E$, e is the only point that lies in this subspace with probability at least $\frac{1}{4t}$. This is given by the Valiant-Vazirani lemma. If we could somehow find all multilinear polynomials of degree $\leq r+1$ on this subspace that vanish at e, we can just recover this point with relative ease.

We repeat the above procedure $O(t \log t)$ times, and we will have found every error point with high probability.

In order to get into slightly more detail, we will set up the following notation, which we will continue to use:

- For $i \in [m]$, let us denote E_i to be the set of errors left after restricting the last i variables to zero and dropping these coordinates, i.e., $E_i := \{x \mid (x_1, \ldots, x_{m-i}, 0, \ldots, 0) \in E\}$.
- For $i \in [m]$, let us denote $V_i \subseteq \mathbb{F}_2[X_1, \ldots, X_{m-i}]$ be the space of all polynomials of degree $\leq r+1$ vanishing on E_i.

Here is another way to look at the above approach which makes the analysis fairly straightforward: Suppose in the (initially unknown) set E, we restricted ourselves only to points that lie on $X_m = 0$, and the number of points is strictly less than $|E|$. We can find the space of all degree $\leq r+1$ polynomials V_1 that vanish on this subset by simply setting $X_m = 0$ in all the polynomials in V (see Section 2.6.5). Thus, we have reduced it to a problem in $\mathbb{F}_2^{m-1}$ with fewer points.

Suppose after setting the last k variables to zero, there is exactly one point e left. Let the space of polynomials that vanish on this point be V_k. We observe that for every $i \in [m-k]$, there is a polynomial $a - X_i \in V_k$ for exactly one value of $a \in \mathbb{F}_2$. This is because V_k is the space of *all* polynomials of degree $\leq r+1$ that vanish on e. So $e(i) - X_i$, for every $i \in [m-k]$, is a degree 1 polynomial vanishing on e, and therefore must belong to V^k. In fact, we can 'read off' the first $m-k$ coordinates of e from V_k by looking at these polynomials. The other coordinates, as dictated by our restriction, are 0 (see Section 2.6.2).

Of course, there are a few problems with the above approach. Firstly, restricting E to $X_n = 0$ might not reduce the size at all. This is exactly where the randomness of the invertible affine transformation comes to use. The idea is that this ensures that around half the points are eliminated at each restriction. For appropriately chosen k, the Valiant-Vazirani Lemma (see Section 2.6.4) says that an affine linear restriction of codimension k isolates exactly one error location with constant probability. Next, a subtle, but crucial point is that after an invertible linear transformation, the set of points E must still satisfy property U_r. Fortunately, this is not very difficult either (see Section 2.6.3).

A final remark is that we store the error location once we find it, but thinking back to the decoding problem, once an error is found, it can also be directly corrected. This step is easy. For example, over $\mathbb{F}_2$, an error location e is corrected by adding the vector $e^{\otimes 2r+1}$ to the syndrome S. Over fields of other characteristics, adding $e^{\otimes 2r+1}$ to the syndrome does not ensure that the error at location e has been corrected. However, this isn't a problem because any error location that has not been corrected will be found again. At this point, we add a different multiple of $e^{\otimes 2r+1}$ to the syndrome and continue.

We will now proceed to analyze each of the above mentioned steps separately.

2.6.2 Counting the number of error locations

We have briefly mentioned that we can check if the size of the set E_i at stage i is 1 or not. However, something more general is true: we can *count* the number of error locations left $|E_i|$ at any stage. This more general fact will prove to be especially useful in the derandomization of this algorithm, given in Section 2.6.6. It basically follows from the following simple fact:

Claim 2.6.1. *The vectors in $E_i^{\otimes \leq r}$ are linearly independent.*

Proof. Consider vector $e^{\otimes \leq r} \in E_i^{\otimes \leq r}$. The entries of e come from a fixed subset of the nonzero coordinates of some vector $\tilde{e}^{\otimes \leq r} \in E^{\otimes \leq r}$. Since, the vectors in $E^{\otimes \leq r}$ are linearly independent, it follows that the vectors in $E_i^{\otimes \leq r}$ are also linearly independent.

□ □

Let $E_i = \{e_1, \ldots, e_k\}$. Since $e_1^{\otimes \leq r}, \ldots, e_k^{\otimes \leq r}$ are linearly independent, we have that $e_1^{\otimes \leq r+1}, \ldots, e_k^{\otimes \leq r+1}$ are also linearly independent. Therefore, V_i given by the null space of the matrix $\mathrm{mat}(E_i^{\otimes \leq r+1})$, which (recall) is given by:

$$\begin{pmatrix} — & e_1^{\otimes \leq r+1} & — \\ & \vdots & \\ — & e_k^{\otimes \leq r+1} & — \end{pmatrix}$$

and as a consequence, has codimension exactly equal to the number of points in E_i. This gives a general way to count the number of error points that we are dealing with. Thus we have:

$$|E_i| = \mathrm{codim}(V_i). \tag{2.1}$$

However, in the case where there is just one point, e, it is easier to check, and even recover the point. The idea is that for every $j \in [m]$, exactly one of X_j and $1 - X_j$ is in V depending on whether $e(j) = 0$ or $e(j) = 1$ respectively. In this spirit, we define the algorithm to read off a point given the space of all degree $\leq r + 1$ polynomials vanishing on it, in fact, the algorithm returns $\perp$ if there isn't exactly one point.

---

procedure FindUniqueRoot(V)
  Initialize e
  for $j \in [m]$ **do**
    if $X_j \in V$ & $1 - X_j \notin V$ **then**
      $e(j) \leftarrow 0$
    else if $1 - X_j \in V$ & $X_j \notin V$ **then**
      $e(j) \leftarrow 1$
    else
      return $\perp$
    end if
  end for
  return e
end procedure

2.6.3 Applying a random invertible affine map

We had also briefly mentioned that when we analyze the algorithm, we are applying a random invertible affine map to $\mathbb{F}_2^m$. We do need to prove that after this map, the set of points still satisfy property U_r.

Proposition 2.6.2. *For an invertible affine map L, if E satisfies property U_r, then $L(E)$ also satisfies U_r.*

Proof. There is a bijection between set of degree $\leq r$ reduced polynomials vanishing on E, and the set of degree $\leq r$ reduced polynomials vanishing on $L(E)$ given by applying the map L to variables. For a reduced polynomial $P(X_1, \ldots, X_m)$ of degree $\leq r$ vanishing on E, we have that the polynomial reduce$(P(L^{-1}(X_1), \ldots, L^{-1}(X_m)))$ vanishes on $T(E)$. Moreover, this is unique, in that no other $P'(X_1, \ldots, X_m)$ maps to this polynomial. This is easy to see since there is a unique way to go between the evaluation tables of $P(X_1, \ldots, X_m)$ and reduce$(P(L^{-1}(X_1), \ldots, L^{-1}(X_m)))$, and no two distinct reduced polynomials have the same evaluation tables.

Similarly, for a reduced polynomial $Q(X_1, \ldots, X_m)$ of degree $\leq r+1$ vanishing on $T = L(E)$, we have that the polynomial reduce$(Q(L(X_1), \ldots, L(X_m)))$ vanishes on E, and no other $Q'(X_1, \ldots, X_m)$ maps to this polynomial. Therefore, the number of points in the null space of $\text{mat}(E^{\otimes \leq r})^T$ has the same size as the null space of $\text{mat}(L(E)^{\otimes \leq r})^T$. Therefore, the spaces has the same codimension, and the rank of $\text{mat}(L(E)^{\otimes \leq r})$ is the same as the rank of $\text{mat}(L(E)^{\otimes \leq r})$.

□ □

2.6.4 The Valiant-Vazirani isolation lemma

Here, we will make use of a simple fact, commonly referred to as the Valiant-Vazirani Lemma [VV86] to isolate a single point in E using a subspace of appropriate codimension.

Lemma 2.6.3 (Valiant-Vazirani Lemma.)**.** *For integers t and m such that $t \leq \frac{1}{100} 2^{m/2}$, let l and c be such that l is an integer, and $l = \log_2 ct$ where $2 \leq c < 4$. Given $E \subset \mathbb{F}_2^m$ such that $|E| = t$, let $a_1, \ldots, a_l$ be uniform among all sets of l linearly independent vectors, and $b_1, \ldots, b_l$ be uniformly and independently chosen elements of $\mathbb{F}_2$. Let*

$$S := \{x \in E \mid \langle x, a_k \rangle = b_k \ \forall k \in [l]\}.$$

Then, for every $e \in E$,

$$\mathbb{P}(S = \{e\}) \geq \frac{1}{7t}.$$

Proof. Assume that $a_1, \ldots, a_l$ are chosen uniformly and independently from $\mathbb{F}_2^m$. let I denote the event $\{a_1, \ldots, a_l \text{ are linearly independent}\}$. We have $\mathbb{P}(I) \geq 1 - \frac{ct}{2^m}$. We have, for every $i \in [t]$, that $\mathbb{P}(\langle e_i, a_k \rangle = b_k) = \frac{1}{2}$. Moreover, we have the pairwise independence property that $\mathbb{P}(\langle e_i, a_k \rangle = b_k \wedge \langle e_j, a_k \rangle = b_k) = \frac{1}{4}$ for $i \neq j$.

With this in mind, let $E = \{e_1, \ldots, e_l\}$, and $\mathcal{E}_i$ denote the event $\{\langle e_i, a_k \rangle = 0 \mid k \in [l]\}$. We have that $\mathbb{P}(\mathcal{E}_i) = \frac{1}{2^l} = \frac{1}{ct}$, and $\mathbb{P}(\mathcal{E}_i \wedge \mathcal{E}_j) = \frac{1}{4^l} = \frac{1}{c^2t^2}$ for $i \neq j$. We have:

$$\mathcal{E}_i \subseteq \left(\mathcal{E}_i \cap \left(\bigcap_{j \neq i} \overline{\mathcal{E}_j}\right)\right) \cup \left(\bigcup_{j \neq i} (\mathcal{E}_i \cap \overline{E}_j)\right).$$

Therefore, by union bound,

$$\mathbb{P}(\mathcal{E}_i) \leq \mathbb{P}\left(\mathcal{E}_i \cap \left(\bigcap_{j \neq i} \overline{\mathcal{E}_j}\right)\right) + \sum_{j \neq i} \mathbb{P}(\mathcal{E}_i \cap \mathcal{E}_j),$$

or

$$\mathbb{P}\left(\mathcal{E}_i \cap \left(\bigcap_{j \neq i} \overline{\mathcal{E}_j}\right)\right) \geq \frac{1}{t}\left(\frac{1}{c} - \frac{1}{c^2}\right) \geq \frac{1}{6t}.$$

And finally, by the law of total probability,

$$\begin{aligned}\mathbb{P}\left(\mathcal{E}_i \cap \left(\bigcap_{j \neq i} \overline{\mathcal{E}_j}\right) \;\middle|\; I\right) &\geq \frac{1}{6t} - \frac{ct}{2^m} \\ &\geq \frac{1}{7t}.\end{aligned}$$

□ □

So, if we restrict $100t \log t$ times, then the probability that some point is never isolated is at most $t\left(1 - \frac{1}{7t}\right)^{100t \log t} \leq 0.001$. What remains is to ensure that we have *all* the polynomials of the given degree that vanish at that point. This is shown by obtaining this set of polynomials after every affine restriction.

2.6.5 Restricting the points to a hyperplane

Here, we just analyze the case when restricting to $X_m = 0$. Further restrictions are analyzed in exactly the same way.

We have E_1, the set of all $z \in \mathbb{F}_2^{m-1}$ such that $(z, 0) \in E$. Let $\hat{E}_1$ be the set of all $z \in \mathbb{F}_2^{m-1}$ such that $(z, 1) \in E$. We also have V_1, the space of all $n-1$ variate polynomials that vanish on E_1. The following lemma shows that V_1 can be found easily from V.

Lemma 2.6.4. *We have that*

$$V_1 = \{P(X_1, \ldots, X_{m-1}, 0) \in \mathbb{F}_2[X_1, \ldots, X_{m-1}] \mid \\ P(X_1, \ldots, X_m) \in V\}.$$

Proof. Let $P(X_1, \ldots, X_m) \in V$. We first show that $Q(X_1, \ldots, X_{m-1}) = P(X_1, \ldots, X_{m-1}, 0)$ lies in V_1. But this is obvious: since $P(X_1, \ldots, X_m) \in V$, we know that $P(y) = 0$ for all $y \in E$. Thus for any $z \in E_1$, $P(z, 0) = 0$. Thus $Q(z) = 0$.

For the other direction, suppose $Q(X_1, \ldots, X_{m-1}) \in V_1$. We need to show that there exists some $P(X_1, \ldots, X_m) \in V$ such that $Q(X_1, \ldots, X_{m-1}) = P(X_1, \ldots, X_{m-1}, 0)$.

We will show that there is some polynomial $P'(X_1, \ldots, X_{m-1}) \in \mathbb{F}[X_1, \ldots, X_{m-1}]$ of degree $\leq r$ such that

$$Q(X_1, \ldots, X_{m-1}) + X_m \cdot P'(X_1, \ldots, X_{m-1}) \in V.$$

Towards this, let $(a_M)_{M \in \mathcal{M}_r^{m-1}}$ be indeterminates, and let $P'(X_1, \ldots, X_{m-1})$ be given by:

$$P'(X_1, \ldots, X_{m-1}) = \sum_{M \in \mathcal{M}_r^{m-1}} a_M M(X_1, \ldots X_{m-1}).$$

We set up a system of linear equations on the a_M:

$$Q(z) + P'(z) = 0 \qquad \text{For every } z \in E_1$$

We claim that there exists a solution $(a_M^*)_{M \in \mathcal{M}_r^{m-1}}$ to this system of equations.

This is because $\hat{E_1}$ satisfies property U_r. This follows from the fact that for every $e \in \hat{E_1}$, i.e., for any $(e, 1) \in E$, the entries of $(e, 1)^{\otimes \leq r}$ are the same as the entries in $e^{\otimes \leq r}$ with some entries repeated, so any linear dependency among the columns of $\hat{E_1}^{\otimes \leq r}$ corresponds to a linear dependency in the columns of $E^{\otimes \leq r}$).

Finally, it remains to check that for every such P' (actually, just some P' is enough), we have that

$$Q(X_1, \ldots, X_{m-1}) + X_m P'(X_1, \ldots, X_{m-1}) \in V,$$

i.e., $Q(X_1, \ldots, X_{m-1}) + X_m P'(X_1, \ldots, X_{m-1})$ vanishes on E. But this is obvious: the case when $X_m = 0$ is taken care of by the fact that $Q \in \tilde{S}$, and the case when $X_m = 1$ is handled by the fact that P' is a solution to our system of equations.

□ □

In the above lemma, all the polynomials in V are *reduced*, i.e., have degree in each variable at most one. When we apply an invertible affine transformation on the variables, we have to ensure that all the polynomials are reduced. However, this is again easy, as it suffices to reduce the basis polynomials of the space. Henceforth, for a set of polynomial $P \in \mathbb{F}_2[X_1 \ldots, X_n]$, we shall denote reduce(P) to be polynomial obtained after reducing P.

And finally, we present the full algorithm

---

procedure FindRoots(V)
  $t \leftarrow \text{codim}(V)$
  Initialize $E \leftarrow \emptyset$ ▷ error set
  for $100t \log t$ iterations **do**
    $M \sim GL(m, \mathbb{F}_2)$, $b \sim \mathbb{F}_2^m$
    for $P \in V$ **do**
      $P(X) \leftarrow \text{reduce}(P(MX + b))$ ▷ affine transformation
    end for
    $V_l \leftarrow \{P(X_1, \ldots, X_{n-l}, 0, \ldots, 0) \mid P(X_1, \ldots, X_n) \in V\}$
    $e \leftarrow$FindUniqueRoot(V)
    if $e \neq \perp$ **then**
      $E \leftarrow E \cup \{e\}$
    end if
  end for
  return E
end procedure

We do $100t \log t$ iterations, and in each step the most expensive operation is FindUniqueRoot, which takes time $O(m^{r\omega+2})$, since it is essentially equivalent to checking if s given vector is in the span of some set of $\leq m^r$ vectors . Therefore, the total running time is $O\left(m^{(\omega+1)r+4}\right)$

2.6.6 A note on derandomization.

In this section, we show how to run the previous algorithm in a derandomized way. The key tool is that we can count the number of common roots of the space via Equation 2.1 for any instance. So, this suggests a natural approach: we try to restrict variables one by one to 0 or 1, and then finding the corresponding space of polynomials by Lemma 2.6.4, only ensuring that the number of common roots after restricting is still nonzero.

To find the running time, we utilize the following recurrence:

$$T(m, |E|) \leq T(m-1, |E_0|) + T(m-1, |E_1|) + \binom{m}{\leq r+1}^{\omega},$$

where $|E_0| + |E_1| = |E|$. This gives a running time bound of $O\left(m^{(\omega+1)r+2}\right)$.

```
procedure DetFindRoots(V)
    V_0 ← {P(X_1, ..., X_{n-1}, 0) | P(X_1, ..., X_n) ∈ V}
    V_1 ← {P(X_1, ..., X_{n-1}, 1) | P(X_1, ..., X_n) ∈ V}
    if codim(V_0) ≠ 0 then
        E_0 ← DetFindRoots(V_0)
    else
        E_0 ← ∅
    end if
    if codim(V_1) ≠ 0 then
        E_1 ← DetFindRoots(V_1)
    else
        E_1 ← ∅
    end if
    return E_0 ∪ E_1
end procedure
```

2.7 Extension to other small fields

The algorithm given above is easily extended to other fields of small order. The reduction of the syndrome decoding problem to finding roots of a space of low degree polynomials, and the isolation lemma can be adapted with almost no change at all. We will only reproduce the result of Section 2.6.5. We do it for $\mathbb{F}_p$ and show that we can recover the whole space of polynomials that vanish on a set of points after one restriction $X_m = 0$. We carry over the notation too. Let $E_1 := \{e \mid (e, 0) \in E\}$. Let $\hat{E}_1 := E \setminus \{(e, 0) \mid e \in E_1\}$.

Lemma 2.7.1.

$$V_1 = \{P(X_1, \ldots, X_{m-1}, 0) \in \mathbb{F}_p[X_1, \ldots, X_{m-1}] \mid P(X_1, \ldots, X_m) \in V\}$$

Proof. As in the previous case, one direction is obvious. Let $P(X_1, \ldots, X_m) \in V$. For every point $(z, 0) \in E$, we have that $P(X_1, \ldots, X_{m-1}, 0)$ vanishes at z.

For the other direction, again similar to the previous case, let $(a_M^{(i)})_{M \in \mathcal{M}_r^{m-1}, i \in [p-1]}$ be indeterminates, let the polynomials in the indeterminates $A_1(X_1, \ldots X_{m-1}), \ldots, A_{p-1}(X_1, \ldots X_{m-1})$ be given by:

$$\begin{aligned} A_1(X_1, \ldots, X_{m-1}) &= \sum_{M \in \mathcal{M}_r^{m-1}} a_M^{(1)} M(X_1, \ldots, X_{m-1}) \\ &\vdots \\ A_{p-1}(X_1, \ldots, X_{m-1}) &= \sum_{M \in \mathcal{M}_{r-p+1}^{m-1}} a_M^{(p-1)} M(X_1, \ldots, X_{m-1}). \end{aligned}$$

and consider the system of linear equations:

$$\begin{aligned} A_1(y^{(1)}) + \cdots + A_{p-1}(y^{(1)}) &= -Q(y^{(1)}) && \text{for } (y^{(1)}, 1) \in E \\ &\vdots \\ (p-1)A_1(y^{(p-1)}) + (p-1)^{p-1}A_{p-1}(y^{(p-1)}) &= -Q(y^{(p-1)}) && \text{for } (y^{(p-1)}, p-1) \in E \end{aligned}$$

Rearranging, we have:

$$\begin{aligned} A_1(y^{(1)}) + \cdots + A_{p-1}(y^{(1)}) &= -Q(y^{(1)}) && \text{for } (y^{(1)}, 1) \in E \\ &\vdots \\ A_1(y^{(p-1)}) + (p-1)^{p-2}A_{p-1}(y^{(p-1)}) &= -(p-1)^{-1}Q(y^{(p-1)}) && \text{for } (y^{(p-1)}, p-1) \in E \end{aligned}$$

We claim that a solution exists, and therefore such a polynomial

$$Q(X_1, \ldots, X_{m-1}) + \sum_{i \in [p-1]} X_m^i A_i(X_1, \ldots, X_{m-1})$$

vanishes on E, and has degree at most $r+1$ and therefore, must belong to V. Let us denote, for $i \in [p-1]$, $E_1^{(i)} := \{e \mid (e, i) \in E\}$. Writing the coefficients on the L.H.S in matrix form, we get

$$\begin{pmatrix} \operatorname{mat}((E_1^{(1)})^{\otimes \leq r}) & \cdots & \operatorname{mat}((E_1^{(1)})^{\otimes \leq r-p+1}) \\ \vdots & \ddots & \vdots \\ \operatorname{mat}((E_1^{(p-1)})^{\otimes \leq r}) & \cdots & (p-1)^{p-2} \operatorname{mat}((E_1^{(p-1)})^{\otimes \leq r-p+1}) \end{pmatrix}$$

It is easy to see that the above matrix is constructed by dropping some repeated columns of $\operatorname{mat}((\hat{E}_1)^{\otimes \leq r})$, and therefore, is full rank.

□

2.8 Discussion and open problems

A very nice question of [ASW15] is to determine whether Reed-Muller codes achieve capacity for the Binary Symmetric Channel. In the constant distance regime, this would amount to being able to correct $\Theta(m^r)$ random errors in the Reed-Muller code $RM(m, m-r)$. If it turns out that Reed-Muller codes do achieve capacity in the BSC, and further if one could find an $\operatorname{poly}(m^r)$-time syndrome decoding algorithm for this setting, then it would give an efficient randomized zero-error

constructions of tensors with high tensor rank. These objects are of great interest in algebraic complexity theory (eg. see [Raz13] and the references therein).

Another interesting problem comes from our second approach to this syndrome decoding problem. Although our algorithm works well over small fields, over large fields, it has a bad dependence on the field size. This mainly comes because when trying to isolate one point using a subspace. It would be interesting to have an algorithm whose running time grows polynomially in $\log p$ instead of p, where p is the size of the field. More concretely is there a $\text{poly}(m^r, \log p)$ algorithm for the following problem?

- *Input:* The space S of all polynomials in m variables of degree at most $r+1$ over $\mathbb{F}_p$ which vanish on an (unknown) set E of points that satisfy property U_r.
- *Output:* The set E.

Chapter 3
Improved inapproximability of rainbow coloring

This chapter is dedicated to studying the inapproximability of *rainbow coloring*. Roughly speaking, the main result of this chapter is that it is **NP**-hard to 2-color a k-uniform, $(k - o(k))$-rainbow colorable hypergraph. Here, the notion of *almost rainbow colorability* is also introduced. A k-uniform hypergraph is r-almost rainbow colorable if, roughly speaking, one can color the vertices in a way that every edge gets $r - o(r)$ colors. We also show that it is **NP**-hard to color an almost $k - o(k)$-rainbow colorable k-uniform hypergraphs with c colors.

3.1 Introduction

A k-uniform hypergraph $H = (V, E)$ consists of a set of vertices V ($|V| = n$) and a collection $E \subset \binom{V}{k}$ of hyperedges.. A (proper) *c-coloring* of H is a coloring of V using c colors such that every hyperedge is non-monochromatic. The complexity of coloring a hypergraph with few colors has been extensively studied over the years.

For $k = 2$ (i.e., graphs), it is **NP**-hard to find a 3-coloring whereas finding a 2-coloring is easy. For higher uniformity $k \geq 3$, even finding a 2-coloring is **NP**-hard. From the upper bounds side, given a 3-colorable graph or 2-colorable 3-uniform hypergraph, the best approximation algorithms, despite a long line of work [KNS01, Chl07, CS08], only find colorings using $O(n^\delta)$ colors for some constant $0 < \delta < 1$.

At the same time, strong inapproximability results for coloring have been elusive. Given a 3-colorable graph, it is **NP**-hard to find a 4-coloring [KLS00], and assuming the $\ltimes$-Conjecture (a variant of the Unique Games Conjecture) it is hard to find a coloring using any constant number of colors [DMR09]. Recently, [BKO19] showed the **NP**-hardness of coloring a 3-colorable graph with 5-colors and in general the **NP**-hardness of coloring a k-colorable graph with $(2k - 1)$-colors. For large constant c, it was known that it is **NP**-hard to color a c-colorable graph using $2^{\Omega(c^{1/3})}$ colors [Hua13], and in general it is known that the chromatic number is **NP**-hard to approximate within $n^{1-\epsilon}$ for every $\epsilon > 0$ [FK98, Zuc07]. Very recently, the results of [Hua13, BKO19] were improved by [WZ20] by showing that for every $c \geq 4$, given a c-colorable graph, it is **NP**-hard to color it with

$\binom{c}{\lfloor c/2 \rfloor} - 1$ colors.

In the hypergraph case, stronger hardness results are known: for instance, given a 4-colorable 4-uniform hypergraph or a 2-colorable 8-uniform hypergraph, it is *quasi*-**NP**-hard[1] to find a coloring using $2^{(\log n)^{1/20-\epsilon}}$ colors for every $\epsilon > 0$ [Var16] following a series of recent developments [DG13, GHH$^+$17, Hua15, KS17]. In the 3-uniform case, the current best hardness is that given a 3-colorable 3-uniform hypergraph it is *quasi*-**NP**-hard to find a coloring with $(\log n)^{\gamma/\log\log\log n}$ colors for some $\gamma > 0$ [GHH$^+$17]. Stronger results are known when the hypergraph is only guaranteed to be *almost* 2-colorable[2]: given an almost 2-colorable 4-uniform hypergraph, it is *quasi*-**NP**-hard to find an independent set of relative size $2^{-\log^{1-o(1)} n}$ [KS14].

Given the strong hardness of hypergraph coloring, it is natural to consider restricted forms of coloring having some additional structure that might make them more amenable to algorithms. One such variant is *rainbow colorability* which is introduced in [AGH17]. A q coloring of the hypergraph is called a rainbow q-coloring if there exists a coloring of the vertices with q colors such that every hyperedge contains all q colors.

Definition 3.1.1 (Rainbow Coloring)**.** *A q-coloring $\chi : V \to [q]$ of a hypergraph $H = (V, E)$ is a* rainbow q-coloring *if for every hyperedge $e \in E$, $\chi(e) = [q]$.*

A hypergraph is called rainbow q-colorable if there exists a rainbow q-coloring. If we restrict the uniformity of the hypergraph to k then the definition of rainbow q-coloring is meaningful only when $2 \leq q \leq k$. It is easy to observe that the property of H being rainbow q-colorable is stronger the larger q is, and that it is always stronger than 2-colorability. We have the following implications on the structure of hypergraphs:

$$k\text{-RC} \Rightarrow (k-1)\text{-RC} \Rightarrow \ldots \Rightarrow 2\text{-RC}$$
$$\Leftrightarrow 2\text{-C} \Rightarrow 3\text{-C} \Rightarrow \ldots \Rightarrow n\text{-C},$$

where i-RC stands for "H is rainbow i-colorable" and i-C stands for "H is i-colorable".

Since rainbow q-colorable hypergraphs have more structure than 2-colorable hypergraphs for $q > 2$, one can hope to improve on the known upper bounds on the hypergraph coloring results in [KNS01] when the given hypergraph is rainbow q-colorable. In this work, we study the inapproximability of coloring such hypergraphs. More concretely, we study the following problem: what guarantee (in terms of rainbow q-colorability) on H is necessary in order for us to be able (in polynomial time) to

[1] there is a DTIME($2^{\text{poly log n}}$) time reduction from 3SAT.

[2] A hypergraph is called almost c-colorable if there is an induced sub-hypergraph of size$(1-\epsilon)n$ which is c-colorable, for $0 < \epsilon \ll 1$.

certify that it is c-colorable? Conversely, for what rainbow colorability guarantees is it still **NP**-hard to find a normal c-coloring? More formally, we define the following decision problem:

Definition 3.1.2 (RAINBOW(k, q, c), $q \leq k$)**.** *Given a k-uniform hypergraph H, distinguish between the following two cases:*

Yes: *H is rainbow q-colorable.*

No: *H is not c-colorable.*

Note that this problem gets *easier* when q increases for a fixed c as well as when c increases for a fixed q.

3.1.1 Related work.

From the upper bounds side, RAINBOW$(k, k, 2)$ is known to be in **P** – a simple randomized algorithms shows that it is in **RP** [McD93] and the problem can be solved without randomness using an SDP [GL15]. In fact, a stronger result is possible: If a given hypergraph is c colorable with the property that there exists two colors, say *red*, *blue*, such that all the hyperedges contain an equal number of red and blue vertices, then the 2-coloring of such hypergraph can be found in polynomial time.

On the inapproximability side, Guruswami and Lee [GL15] showed that, for all constants $k, c \geq 2$, RAINBOW$(k, \lfloor k/2 \rfloor, c)$ is **NP**-hard. Even in the case of $c = 2$, this remains the current best **NP**-hardness result in terms of rainbow coloring guarantee for any fixed $k > 3$ i.e their result does not rule out RAINBOW$(k, \lfloor k/2 \rfloor + 1, 2) \in \mathbf{P}$. Austrin et al. [AGH17] asked the question whether it is **NP**-hard to find a 2-coloring of rainbow $(k-1)$-colorable k-uniform hypergraph.

Brakensiek and Guruswami [BG16] conjectured that RAINBOW$(k, k-1, 2)$ is **NP**-hard. Later they showed [BG17] that a strong form of this conjecture would follow assuming a "V Label Cover" conjecture. Assuming that conjecture, for any $\epsilon > 0$ it is **NP**-hard to even find an independent set of an ϵ fraction of vertices (and in particular it is hard to find a $1/\epsilon$-coloring) in a rainbow $(k-1)$-colorable k-uniform hypergraph. However, the V Label Cover conjecture (which is essentially a variant of the Unique Games Conjecture with perfect completeness) is very strong and it is not clear yet whether it should be believed.

Recently Guruswami and Saket [GS17], further restrict the guarantee on the rainbow coloring to *balanced* rainbow coloring. More specifically, for $Q, k \geq 2$, suppose we are given a Qk-uniform hypergraph with the guarantee that it is rainbow k-colorable such that in every hyperedge ℓ colors occur exactly $Q-1$ times, ℓ colors occur exactly $Q+1$ and the remaining occur exactly Q times for some parameter $1 \leq \ell \leq k/2$. In this case, they show that it is **NP**-hard to find an independent

set of size roughly $(1 - \frac{\ell+1}{k})$. Note that in their result, the hypergraph might not satisfy rainbow $(\lfloor k/2 \rfloor + 1)$-coloring guarantee and therefore the result in [GS17] does not even rule out efficiently finding a 2-coloring when the k-uniform hypergraph is rainbow $(\lfloor k/2 \rfloor + 1)$-colorable.

A dual notion to rainbow colorability is that of *strong coloring*. A k-uniform hypergraph H is strongly q-colorable for $q \geq k$ if there is a q-coloring of H such that every hyperedge contains k different colors. Note that the two notions coincide when $q = k$. [BG16] studied the problem of finding a c-coloring of a strongly q-colorable hypergraph. On the hardness side, they showed that it is **NP**-hard to find a 2-coloring of a strongly $\lceil 3k/2 \rceil$-colorable k-uniform hypergraph. Since the focus of this paper is on rainbow coloring, we refer interested readers to [BG16] for more details about strong rainbow coloring.

3.2 The main results

We show the following hardness results. First, we give a relatively simple proof that it is **NP**-hard to find a 2-coloring even when the graph is guaranteed to be roughly rainbow $(k - 2\sqrt{k})$-colorable. This significantly improves on the hardness bounds of [GL15] and settles the smallest previous unknown case which was Rainbow$(4, 3, 2)$. Concretely, we show the following.

Theorem 3.2.1. *For every $t \geq 1, d \geq 2$,* Rainbow$(td + \lfloor \frac{d}{2} \rfloor, t(d-1) + 1, 2)$ *is* **NP***-hard.*

We have the following corollary (proved in Section 3.7):

Corollary 3.2.2. *For all $k \geq 6$,* Rainbow$(k, k - 2\lfloor \sqrt{k} \rfloor, 2)$ *is* **NP***-hard.*

The **NP**-hardness result of Rainbow$(4, 3, 2)$ has been improved recently by Guruswami and Sandeep [GS19], who show that for a k-uniform hypergraph, it is **NP**-hard to rainbow q-color a rainbow $(k-1)$-colorable hypergraph where $q = \lfloor \frac{k-1}{2} \rfloor$. In particular, this shows **NP**-hardness of Rainbow$(k, k-1, 2)$ for $4 \leq k \leq 6$.

The techniques used in the proof the Theorem 3.2.1 can only show 2-coloring in the soundness case. Towards obtaining similar results for $c > 2$, we introduce a generalization of rainbow coloring in which we only require that each hyperedge contains at least p different colors for some $p \leq q$.

Definition 3.2.3 (Rainbow (q, p)-Coloring)**.** *A q-coloring $\chi : V \to [q]$ of a hypergraph $H = (V, E)$ is a* rainbow (q, p)-coloring *if for every hyperedge $e \in E$, $|\chi(e)| \geq p$.*

A hypergraph is called rainbow (q, p)-colorable if there exists a rainbow (q, p)-coloring. Note that rainbow (q, q)-coloring is same as rainbow q-coloring, and that as long as $p > \lceil q/2 \rceil$ then a (q, p)-colorable graph is still always 2-colorable. We define the following decision problem analogously to Rainbow(k, q, c).

Definition 3.2.4 (AlmostRainbow(k, q, p, c))**.** *Given a k-uniform hypergraph H, where that $p \leq q \leq k$, $p > \lceil q/2 \rceil$, distinguish between the following two cases:*

Yes: *H is rainbow (q, p)-colorable.*

No: *H is not c-colorable.*

We prove the following hardness result for AlmostRainbow(k, q, p, c).

Theorem 3.2.5. *For every $d \geq c \geq 2$ and $t \geq 2$ such that d and t are primes and d is odd, let $q = t(d - c + 1) + c - 1$ and $k = td$. Then* AlmostRainbow$(k, q, q - d, c)$ *is* **NP***-hard (provided $d < \lfloor q/2 \rfloor$ so that the* AlmostRainbow *problem is well-defined).*

For $q \geq 4c$, setting d to be a prime between $\sqrt{qc}$ and $2\sqrt{qc}$ we have the following more concrete corollary.

Corollary 3.2.6. *For infinitely many $q \geq 4c$,* AlmostRainbow$(q + \lfloor \sqrt{qc} \rfloor, q, q - \lfloor 2\sqrt{qc} \rfloor, c)$ *is* **NP***-hard.*

In particular this means that AlmostRainbow$(q + o(q), q, q - o(q), c)$ is **NP**-hard for infinitely many q and $c = o(q)$.

A key difference between our results and previous hardness results is that we only show hardness of finding a c-coloring, not hardness of finding a large independent set (which is an easier task than finding a c-coloring). In fact, the graphs constructed in our reduction always have independent sets consisting of almost $1/2$ the vertices.

3.3 A sketch of the proofs

Like so many other strong hardness of approximation results, our proof follows the general framework of *long code*-based gadget reductions from the *Label Cover* problem. However, we depart from the predominant approach of analyzing such reductions using tools from discrete Fourier analysis such as (reverse) hypercontractivity or invariance principles. Indeed, such methods appear inherently ill-suited to analyze our gadgets – as alluded to earlier, our gadgets have very large independent sets, and Fourier-analytic methods usually cannot say anything about the chromatic number of such graphs.

Instead we use methods from topological combinatorics to analyze our gadgets. Since its introduction with Lovsz' resolution of Kneser's conjecture in 1978 [Lov78], topological combinatorics has been used to resolve a number of combinatorial problems, many of them regarding the chromatic number of various families of graphs and hypergraphs.

The lower bound on the chromatic number of Kneser graphs (or more accurately, the lower bound on the chromatic number of the Schrijver graphs, which are vertex-critical subgraphs of the Kneser graphs) was used by Dinur et al. [DRS02] and recently by Bhangale [Bha18] to analyze a long code gadget giving **NP**-hardness of coloring 3-uniform hypergraphs with any constant number of colors and of coloring 4-uniform hypergraphs with $\texttt{poly}(\log n)$ number of colors respectively. Recently [KO19, WZ20] used topological methods to analyze the gadgets in the reduction of generalized graph coloring. Their proofs are algebraic in nature compared to ours which are combinatorial.

For our results, we construct a new family of hypergraphs that we call rainbow hypergraphs. These are k-uniform hypergraphs over the n-dimensional k-ary cube $[k]^n$, and k strings $\mathbf{x}^1, \ldots, \mathbf{x}^k$ form a hyperedge if, in all but a constant number t of coordinates $i \in [n]$, it holds that $\mathbf{x}^1_i, \ldots, \mathbf{x}^k_i$ are all different. Our hardness results rely on lower bounds on the chromatic number of these hypergraphs. For Theorem 3.2.1, a simple direct proof yields non-2-colorability of the corresponding rainbow hypergraph, whereas for Theorem 3.2.5, we give a proof that the chromatic number of the corresponding rainbow hypergraph grows with t, based on a generalization of the Borsuk-Ulam theorem (see Theorem 3.8.2).

We now give a brief informal overview of how these rainbow hypergraphs can be used as gadgets in a Label Cover reduction. At their core, these reductions boil down to a type of *dictatorship testing*, in the following sense. We have a large set of functions $f_1, \ldots, f_u : [q]^n \to [q]$, and our task is to define a hypergraph with vertex set $[u] \times [q]^n$ such that:

Completeness If the functions are all the same dictator function (depending only on one coordinate in their input), then using the function values as colors (i.e., the vertex $(i, \mathbf{x})$ gets color $f_i(\mathbf{x})$) results in a rainbow q-coloring.

Soundness Each function f_a can be decoded to a small set of coordinates $S_a \subseteq [n]$ (depending only on f_a and not the other functions) such that if the function induces a proper c-coloring then many pairs of functions f_a, f_b have overlapping decoded coordinates (i.e., $S_a \cap S_b \neq \emptyset$).

One simple way of constructing such a dictatorship test would be as follows: let H be a 3-uniform rainbow hypergraph (over $[3]^n$) which is not 2-colorable. For an edge $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$ of H, we refer to the set of $\leq t$ coordinates where $\{\mathbf{x}^1_i, \mathbf{x}^2_i, \mathbf{x}^3_i\} \neq [3]$ as the *noisy coordinates* of the edge. Now create a 6-uniform hypergraph on $[u] \times [3]^n$ by for every pair $a, b \in [u]$ adding an edge consisting of $\{(a, \mathbf{x}^1), (a, \mathbf{x}^2), (a, \mathbf{x}^3), (b, \mathbf{y}^1), (b, \mathbf{y}^2), (b, \mathbf{y}^3)\}$ whenever (i) $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$ and $\{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$ are edges in H, and (ii) for each $i \in [n]$, $\{\mathbf{x}^1_i, \mathbf{x}^2_i, \mathbf{x}^3_i, \mathbf{y}^1_i, \mathbf{y}^2_i, \mathbf{y}^3_i\} = [3]$. It should be clear that this 6-uniform graph is rainbow 3-colorable using any dictatorship coloring. For the soundness, consider any 2-coloring of the vertices. By the non 2-colorability of H, each f_i has a H-monochromatic edge $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$.

For any pair (a, b) of such f's with an H-monochromatic edge of the same color, it follows that $\{\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{x}_i^3, \mathbf{y}_i^1, \mathbf{y}_i^2, \mathbf{y}_i^3\} \neq [3]$ for some $i \in [n]$, otherwise we would have a monochromatic hyperedge. This means that the set of noisy coordinates for the two H-monochromatic edges overlaps, so if we decode each f_a to the set of $\leq t$ noisy coordinates, then at least half the pairs of functions f_a, f_b have overlapping decoded coordinates. This essentially proves hardness of Rainbow$(6, 3, 2)$.

To get hardness of Rainbow$(4, 3, 2)$, we modify the construction slightly to make it lopsided by only using one vertex $(b, \mathbf{y})$ from the b part, instead of a full hyperedge of H. It turns out that the soundness property still holds, using an additional property that every 2-coloring of H must have a monochromatic hyperedge from a large color class.

For the general cases Theorem 3.2.1,Theorem 3.2.5, the construction is generalized as follows. We use as gadget a non-c-colorable d-uniform rainbow hypergraph H for $c, d < q$, and construct hyperedges as follows: pick any r functions $f_{a_1}, \ldots, f_{a_r}$, and for each such f_{a_j} pick d strings $\mathbf{x}^{j,1}, \ldots, \mathbf{x}^{j,d} \in [q]^n$ such that in each coordinate $i \in [n]$, the set of values seen in the $r \cdot d$ strings is all of $[q]$ (this is the analogue of condition (ii) above). The soundness analysis of this construction is more involved. The key idea here is that for any $\sigma \in \binom{[q]}{d}$, f_a restricted to σ^n induces a coloring of H and thus contains a monochromatic hyperedge. If r is sufficiently large, there is in fact a cover $\sigma_1, \sigma_2, \ldots, \sigma_r \in \binom{[q]}{d}$ of $[q]$ such that the copies of H under each of these σ_j's have a monochromatic hyperedge of the same color. By a pigeon hole argument, a constant fraction of f_a's must have the same monochromatic cover and we show that this can be used to decode each f_a to a small set of candidate coordinates.

The bound on the uniformity we get is $r \cdot d$, where r is lower bounded by the need to obtain the covering property described above. Using a theorem of Sarkaria, we show in Section 3.4.2 that r can be taken as approximately $\frac{q-c+1}{d-c+1}$ (which is tight for the covering property).

3.3.1 Organization of the chapter

Since this chapter contains mutiple results, we give a brief sketch of the layout.

Section 3.4 provides some necessary background material regarding hardness of Label Cover and a combinatorial covering bound. In Section 3.5 we define the rainbow hypergraph gadget used for Theorem 3.2.1 and show that it is not 2-colorable. As a warm-up we then provide in Section 3.6 a special case of Theorem 3.2.1, **NP**-hardness of Rainbow$(4, 3, 2)$, since this is much simpler than the general reductions of Theorem 3.2.1 and Theorem 3.2.5 (experts may want to skip Section 3.6). In Section 3.8 we define the more general rainbow hypergraph gadget used for Theorem 3.2.5 and lower bound its chromatic number, and then proceed to prove Theorem 3.2.5 in Section 3.9. The full proof of Theorem 3.2.1 and Corollary 3.2.2 is given in Section 3.7. In Section 3.10 we give some

concluding remarks and further research directions.

3.4 Preliminaries

We denote the set $\{1, 2, 3, \ldots, n\}$ by $[n]$. Bold face letters $\mathbf{x}, \mathbf{y}, \mathbf{z} \ldots$ are used to denote strings. When we have a collection of several strings we use superscripts to index which string is referred to, and subscripts to index into locations in the strings, e.g., $\mathbf{x}^i_j$ denotes the entry in the j'th position of the i'th string.

3.4.1 Label Cover

The starting point in our hardness reductions is the *Layered Label Cover* problem, defined next.

Definition 3.4.1 (Layered Label Cover)**.** *An ℓ-layered Label Cover instance consists of ℓ sets of variables $X = \{X_1, \ldots, X_\ell\}$. The range of variables in layer i is denoted by $[R_i]$. Every pair of layers $1 \leq i < j \leq \ell$ has a set of constraints Φ_{ij} between the variables in X_i and X_j. The constraint between $x \in X_i$ and $y \in X_j$ is denoted by $\phi_{x \to y}$. Moreover, every constraint between a pair of variables is a projection constraint – for every assignment $k \in [R_i]$ to x there is a unique assignment to y that satisfies the constraint $\phi_{x \to y}$.*

In a Label Cover instance as defined above, for any constraint $\phi_{x \to y} \in \Phi_{i,j}$, we view it as a function $\phi_{x \to y} : [R_i] \to [R_j]$ defined such that for any $k \in [R_i]$, $(k, \phi_{x \to y}(k))$ satisfies the constraint $\phi_{x \to y}$. Thus, where there is no ambiguity, we will use $\phi_{x \to y}$ to denote both the constraint, as well as the function. Moreover, for brevity, we say $x \sim y$, or "x is a neighbour of y" if $\phi_{x \to y} \in \Phi_{i,j}$.

Definition 3.4.2 (Weakly dense, [DGKR05])**.** *An instance of ℓ-layered Label Cover is* weakly dense *if the following property holds. For any m layers $i_1 < \cdots < i_m$, where $1 < m < l$, and any sequence of variable sets $S_k \subseteq X_{i_k}$ for $k \in [m]$ such that $|S_k| \geq \frac{2}{m}|X_{i_k}|$, we have that there are two sets S_k and $S_{k'}$ such that the number of constraints between S_k and $S_{k'}$ is at least a $\frac{1}{m^2}$ fraction of the total number of constraints between layers X_{i_k} and $X_{i_{k'}}$.*

We have the following **NP**-hardness result from [DGKR05], [DRS02], which we use as a starting point in proving Theorem 3.2.1.

Theorem 3.4.3 ([DGKR05], [DRS02])**.** *For any constant parameters $\ell \geq 2, r \in \mathbb{Z}$ the following problem is* **NP***-hard. Given a weakly dense ℓ-layered Label Cover instance where all variable ranges $[R_i]$ are of size $2^{O(\ell r)}$, distinguish between the following two cases:*

Completeness *There is an assignment satisfying all the constraints of the Label Cover instance.*

Soundness *For every $1 \leq i < j \leq \ell$, no assignment satisfies more than a $2^{-\Omega(r)}$ fraction of the set of constraints $\Phi_{i,j}$ between layers i and j.*

3.4.2 A Covering Bound

We say a function $f : \binom{[q]}{d} \to [c]$ has a t-cover if there is a family $\mathcal{S} \subseteq \binom{[q]}{d}$ of size $|\mathcal{S}| = t$ such that $\cup_{S \in \mathcal{S}} = [q]$ and f is constant on $\mathcal{S}$. Let $B(q, d, c)$ be the minimum t such that every $f : \binom{[q]}{d} \to [c]$ has a t-cover.

Claim 3.4.4. *For all $1 \leq c \leq d$, $B(q, d, c) \geq \left\lceil \frac{q-c+1}{d-c+1} \right\rceil$. For $c \geq d+1$ and $q \geq d+1$ a cover may fail to exist.*

Proof. For $S \in \binom{[q]}{d}$, set $f(S)$ to be the smallest $i \in [c-1]$ such that $i \notin S$, or $f(S) = c$ if $[c-1] \subseteq S$.

By definition, $f^{-1}(i)$ does not cover $[n]$ for $i \in [c-1]$, so any cover must use sets from $f^{-1}(c)$. However all such sets contain $[c-1]$, so the total number of elements covered by k sets from $f^{-1}(c)$ is at most $d + (k-1)(d-c+1)$ thus in order to obtain a cover of all q elements we need $d + (k-1)(d-c+1) \geq q$ or equivalently $k \geq \frac{q-c+1}{d-c+1}$. □

In the case $c = 2$, there is a simple inductive proof (see Lemma 3.7.2) that the lower bound of Claim 3.4.4 is tight. By a simple reduction to the *Generalized Kneser Hypergraph*, we get nearly matching upper bounds for all values of c. The Generalized Kneser Hypergraph has vertex set $\binom{[n]}{k}$, and a collection of (not necessarily distinct) sets $\mathcal{S} = \{S_1, \ldots, S_t\}$ forms a hyperedge if each element in $[n]$ is present in at most s sets in $\mathcal{S}$. For our bound, we only need the special case where $s = t-1$, where a hyperedge just translates to a collection of sets with empty intersection.

Sarkaria [Sar90] lower bounded the chromatic number of the Generalized Kneser Hypergraph for many cases, and in particular for the $s = t-1$ case we have the following.

Theorem 3.4.5. *For any choice of integer parameters n, k, c, t with $n \geq k$ and t prime, satisfying $n(t-1) - 1 \geq c(t-1) + t(k-1)$, and any c-coloring of $\binom{[n]}{k}$ there exist t sets $S_1, \ldots, S_t \in \binom{[n]}{k}$ of the same color such that their intersection is empty.*

Sarkaria's Theorem as originally stated [Sar90] did not require t to be prime, but the proof does not work in general for the non-prime case [LZ07], and the result is in general currently only known to hold for t prime or a power of 2 (see also [ACC+18]). Interestingly enough, all the proofs of the aforementioned results heavily use topology and we are not aware of any *non-topological* proof of this covering theorem.

Using Theorem 3.4.5, we get a nearly sharp upper bound on $B(q, d, c)$. If the requirement that t is prime in Theorem 3.4.5 could be dropped, we would get the exact values of $B(q, d, c)$.

Theorem 3.4.6. *For all $1 \le c \le d$, $B(q,d,c) \le p(q,d,c)$, where $p(q,d,c)$ is the smallest prime that is at least $\left\lceil \frac{q-c+1}{d-c+1} \right\rceil$.*

Proof. Let $f : \binom{[q]}{d} \to [c]$ be arbitrary. Let $n = q$, $k = q-d$, and define $\tilde{f} : \binom{[n]}{k} \to [c]$ by $\tilde{f}(S) = f(\overline{S})$, where $\overline{S} = [q] \setminus S$. By Theorem 3.4.5, for any prime t that satisfies $q(t-1)-1 \ge c(t-1)+t(q-d-1)$, or equivalently $t \ge \frac{q-c+1}{d-c+1}$, there exist t sets $T_1, \ldots, T_t \in \binom{[n]}{k}$ such that $\cap_{i=1}^{t} T_i = \emptyset$ and $\tilde{f}(T_1) = \ldots = \tilde{f}(T_t)$. Letting $S_i = \overline{T_i}$ we have $\cup_{i=1}^{t} S_i = [n]$, so f indeed has a monochromatic cover of size t provided $t \ge \frac{q-c+1}{d-c+1}$ □

3.5 Rainbow Hypergraph Gadget for 2-coloring

Definition 3.5.1. *(The hypergraph $\mathtt{H}_r^n([d])$) Let $\mathtt{H}_r^n([d])$ be the d-uniform hypergraph with vertex set $[d]^n$ where d vertices $\mathbf{x}^1, \ldots, \mathbf{x}^d \in [d]^n$ form a hyperedge iff*

$$\sum_{i=1}^{n} |[d] \setminus \{x_i^j \mid j \in [d]\}| \le r.$$

The up to r coordinates $i \in [n]$ where $|\{x_i^j \mid j \in [d]\}| \ne d$ are called noisy coordinates.

In other words, if we write down $\mathbf{x}^1, \ldots, \mathbf{x}^d$ in a $d \times n$ matrix form, and it is possible to change up r entries so that all the columns become permutations of $[d]$, then these vertices form a hyperedge. We sometimes abuse the notation and instead of $[d]$ in $\mathtt{H}_r^n([d])$, either use a finite set or a finite group. The definition of $\mathtt{H}_r^n(\cdot)$ still makes sense with this change.

The following claim shows that the hypergraph $\mathtt{H}_r^n([d])$ is not 2-colorable for $r = \lfloor d/2 \rfloor$.

Lemma 3.5.2. *For all $d \ge 2$, $\mathtt{H}_{\lfloor d/2 \rfloor}^n([d])$ is not 2-colorable.*

Proof. We prove the claim by induction on d. We take the natural convention that the 0-uniform hypergraph, and a 1-uniform hypergraph, are not 2-colorable. Therefore the base cases $d = 0$ or $d = 1$ are trivial.

Suppose the claim is true for $d-2$. For contradiction assume $\mathtt{H}_{\lfloor d/2 \rfloor}^n([d])$ is 2-colorable and that $f : [d]^n \to \{0,1\}$ is some 2-coloring of $\mathtt{H}_{\lfloor d/2 \rfloor}^n([d])$. Since f is not a constant function, there exists $\mathbf{x}^1$ and a coordinate i such that changing i'th coordinate of $\mathbf{x}$ changes the value of f. Without loss of generality, $\mathbf{x}^1 = \mathbf{d}$, $f(\mathbf{x}^1) = 1$, and $f(\tilde{\mathbf{x}}^1) = 0$, where $\tilde{\mathbf{x}}^1$ is a string which differs from $\mathbf{x}^1$ only in the i'th coordinate.

Now, the restricted function on $[d-1]^n$ cannot be a constant function; since otherwise $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{d-1}\}$ along with either $\mathbf{x}^1$ or $(\mathbf{x}^1 + \delta_i)$ form a monochromatic hyperedge, contradicting the assumption that f is a proper 2-coloring of $\mathtt{H}_{\lfloor d/2 \rfloor}^n([d])$. Since, f on $[d-1]^n$ is not a constant function, we can find

$\mathbf{x}^2$ and a coordinate j such that $f(\mathbf{x}^2) \neq f(\tilde{\mathbf{x}}^2)$, where again $\tilde{\mathbf{x}}^2$ differs from $\mathbf{x}^2$ only at coordinate j. Without loss of generality, we can assume $\mathbf{x}^2 = \mathbf{d} - \mathbf{1}$ and $f(\mathbf{x}^2) = 0$ (and hence $f(\tilde{\mathbf{x}}^2) = 1$).

By the induction hypothesis, $\mathtt{H}^n_{\lfloor d/2 \rfloor - 1}([d-2])$ is not 2-colorable and thus there exists a monochromatic hyperedge if we color the vertices $[d-2]^n$ according to f. Let the hyperedge be $\{\mathbf{x}^3, \mathbf{x}^4, \ldots, \mathbf{x}^d\}$ and $f(\mathbf{x}^3) = f(\mathbf{x}^4) = \ldots = f(\mathbf{x}^d)$. If $f(\mathbf{x}^3) = 0$, then $\{\mathbf{x}^3, \mathbf{x}^4, \ldots, \mathbf{x}^d\} \cup \{\mathbf{x}^2, \tilde{\mathbf{x}}^1\}$ is a 0-monochromatic hyperedge. Otherwise, $\{\mathbf{x}^3, \mathbf{x}^4, \ldots, \mathbf{x}^d\} \cup \{\mathbf{x}^1, \tilde{\mathbf{x}}^2\}$ is a 1-monochromatic hyperedge. Thus, f is not a 2-coloring of $\mathtt{H}^n_{\lfloor d/2 \rfloor}([d])$. □

Let $\alpha(H)$ denote the relative size of a maximum independent set of a hypergraph H. We have the following simple fact:

Fact 3.5.3. *For all $n \geq 2$, $\alpha(\mathtt{H}^n_1([3])) \leq \frac{2}{3}$.*

Proof. Consider the equivalence class $\mathbf{x}, \mathbf{x}+\mathbf{1}, \mathbf{x}+\mathbf{2}$ where $+$ is the coordinate wise addition $\mod 3$. Then any independent set must contain at most 2 elements from any equivalence class. □

3.6 Warm-up: Hardness of Rainbow$(4, 3, 2)$

In this section, we prove the special case of Theorem 3.2.1 that Rainbow$(4, 3, 2)$ is **NP**-hard. This illustrates many of the ideas of the reductions for the general results in a simpler context, but an expert reader may want to skip this section and instead go directly to the full proof Theorem 3.2.1, in Section 3.7.

3.6.1 Reduction

We give a reduction from the ℓ-layered Label Cover instance with parameters $\ell = 8$ and r a sufficiently large constant from Theorem 3.4.3 to a 4-uniform hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$. We will select r such that the Label Cover soundness is smaller than $1/48$. The reduction is given as follows:

Vertices $\mathcal{V}$: Each vertex v from layer i in the layered Label Cover instance $\mathcal{L}$ is replaced by a cloud of size 3^{R_i} denoted by $C[v] := v \times \{0, 1, 2\}^{R_i}$. We refer to a vertex from cloud $C[v]$ by a pair $(v, \mathbf{x})$ where $\mathbf{x} \in \{0, 1, 2\}^{R_i}$. The vertex set of the hypergraph is given by

$$\mathcal{V} = \cup_{v \in \cup_i X_i} C[v].$$

Hyperedges $\mathcal{E}$: Hyperedges are given by sets $\{(u, \mathbf{x}), (u, \mathbf{y}), (u, \mathbf{z}), (v, \mathbf{w})\}$ such that:

1. There are i, j such that $u \in X_i$, $v \in X_j$, and $u \sim v$.

2. $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ form an edge in $\mathtt{H}_1^{R_i}(\{0,1,2\})$.

3. $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k, \mathbf{w}_{\phi_{u\to v}(k)}\} = \{0,1,2\}$ for all $k \in [R_i]$

For a hyperedge $\{(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z}),(v,\mathbf{w})\} \in \mathcal{E}$, we say that a coordinate $k \in [R_i]$ is *noisy* if $|\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}| = 2$.

Lemma 3.6.1 (Completeness)**.** *If the Label Cover instance is satisfiable then the hypergraph $\mathcal{H}$ is rainbow 3-colorable.*

Proof. Let $A : \bigcup_i X_i \to \bigcup_i [R_i]$ define the assignment satisfying all constraints of the layered Label Cover instance. The rainbow 3-coloring of the hypergraph is given by assigning a vertex $(v, \mathbf{x})$ the color $\mathbf{x}_{A(v)}$.

A hyperedge $\{(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z}),(v,\mathbf{w})\}$ is thus given the set of colors

$$\{\mathbf{x}_{A(u)}, \mathbf{y}_{A(u)}, \mathbf{z}_{A(u)}, \mathbf{w}_{A(v)}\}.$$

Since A satisfies all constraints, we have that $A(v) = \phi_{u\to v}(A(u))$ and by Item 3 in the definition of $\mathcal{E}$ it follows that we see all three colors. □

Lemma 3.6.2 (Soundness)**.** *If the hypergraph $\mathcal{H}$ is 2-colorable then there exists an assignment A to the Label Cover instance which satisfies a $1/32$ fraction of all constraints between some pair of layers X_i and X_j.*

Proof. Fix a 2-coloring of the hypergraph. Call the colors red and blue. Consider $\mathtt{H}_1^{R_i}([3])$ defined on the cloud $C[v]$ for $v \in X_i$. By Lemma 3.5.2, and Fact 3.5.3, there exists a color class so that more than $\frac{1}{3}$ fraction of vertices in $C[v]$ are colored with that color and there exists a monochromatic hyperedge with the same color. Label a vertex v 'red' if that hyperedge is colored red otherwise label it 'blue' (breaking ties using 'red' by default). Label a layer with a color which we used to label maximal number of clouds in the layer. Out of the 8 layers there are at least 4 layers of the same color. Without loss of generality, let the color be red.

By the weak density property of layered Label Cover instance, out of these 4 layers there exist two layers i and j $(i < j)$ such that the total number of constraints between the red variables in those two layers is at least $\frac{1}{16}$ times the total number of constraints between X_i and X_j. We now give a labeling to the red variables in X_i and X_j which satisfies a constant fraction of the induced constraints.

From now on, let U denote the red variables of X_i and V the red variables of X_j. We know from above that the total number of constraints between U and V is at least $\frac{1}{16}$ times the total number

of constraints between layers i and j. Thus, if we show that we can satisfy a constant fraction of constraints between U and V then we are done.

Labeling: We define the labeling A to vertices $U \cup V$ as follows: for $u \in U$, the copy of $\mathtt{H}_1^{R_i}([3])$ has a monochromatic red edge. Let that edge be $\{(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z})\}$. If the edge has a noisy coordinate $k \in [R_i]$ then set $A(u) = k$, otherwise set $A(u) = \perp$. This defines the labeling of the vertices in U.

For $v \in V$, consider the following collection of labels:

$$S_v = \{\phi_{u\to v}(A(u)) \mid u \in U,\ u \sim v\}.$$

Here we define $\phi_{u\to v}(\perp) = \perp$ for all $u \sim v$. We assign a label to v randomly by picking a uniformly random label from S_v.

Claim 3.6.3. *For every $v \in V$, $\perp \notin S_v$ and it holds that $|S_v| \le 2$.*

Proof. If $\perp \in S_v$ then by definition, there exists $u \sim v$, $u \in U$ and $\mathbf{x},\mathbf{y},\mathbf{z} \in \{0,1,2\}^{[R_i]}$ such that $(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z})$ are colored red and $\{\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k\} = \{0,1,2\}$ for all $k \in [R_i]$. Thus, $\{\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k\}$ along with *any* vertex $(v,\mathbf{w}) \in C[v]$ form a hyperedge in $\mathcal{E}$. Since $v \in V$, it has at least one red colored vertex $C[v]$, but this gives a monochromatic hyperedge with respect to the coloring.

Consider a label $t \in S_v$. Every such label imposes a restriction on the elements in the cloud $C[v]$ that are colored red. By definition there is a $u \in U$ such that $\phi_{u\to v}(A(u)) = t$, and $\mathbf{x},\mathbf{y},\mathbf{z} \in \{0,1,2\}^{[R_i]}$ such that $(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z})$ are colored red and $|\{\mathbf{x}_{A(u)},\mathbf{y}_{A(u)},\mathbf{z}_{A(u)}\}| = 2$. Thus, for every $\mathbf{w} \in \{0,1,2\}^{R_j}$ such that $(v,\mathbf{w})$ is colored red, it must be the case that $\mathbf{w}_t \in \{\mathbf{x}_{A(u)},\mathbf{y}_{A(u)},\mathbf{z}_{A(u)}\}$ because otherwise $\{(u,\mathbf{x}),(u,\mathbf{y}),(u,\mathbf{z}),(v,\mathbf{w})\}$ would form a monochromatic hyperedge of $\mathcal{H}$.

In other words, for every $t \in S_v$, there is at least one value $z_t \in \{0,1,2\}$ such that all red vertices $(v,\mathbf{w})$ of $C[v]$ have $\mathbf{w}_t \neq z_t$. This implies that the fraction of red vertices in $C[v]$ is at most $(2/3)^{|S_v|}$. But by construction, at least a $1/3$ fraction of vertices in $C[v]$ are red, and it follows that $|S_v| \le 2$. □

It now follows that the randomized labeling A defined above satisfies at least a $1/2$ fraction of all constraints between U and V in expectation, and since the constraints between U and V constitute a $1/16$ fraction of all constraints between layers i and j, we are done. □

3.7 The Rainbow $\left(td + \lfloor \frac{d}{2} \rfloor, t(d-1)+1, 2\right)$-hardness

In this section, we give a generalization of the Rainbow$(4,3,2)$ result from the Section 3.6. This gives an elementary proof of Rainbow$(td + \lfloor \frac{d}{2} \rfloor, t(d-1)+1, 2)$-hardness.

Theorem 3.7.1 (Theorem 3.2.1 restated)**.** *For every $t \geq 1$ and $d \geq 2$,* Rainbow$(td + \lfloor \frac{d}{2} \rfloor, t(d-1)+1, 2)$ *is* **NP***-hard.*

In the proof of this theorem, we use the $c = 2$ case of the covering bound Theorem 3.4.6 (c.f. Section 3.4.2). While we are not aware of any non-topological proof of the full version of Theorem 3.4.6, the $c = 2$ case does admit an simple inductive proof, provided here for completeness.

Lemma 3.7.2 ($c = 2$ case of Theorem 3.4.6)**.** *For every $q \geq d \geq 2$, $B(q, d, 2) = \lceil \frac{q-1}{d-1} \rceil$, i.e., for every $f : \binom{[q]}{d} \to \{0, 1\}$, there are $b = \lceil \frac{q-1}{d-1} \rceil$ sets $S_1, \ldots, S_b \in \binom{[q]}{d}$ such that $\cup S_i = [q]$ and f is constant on $S_1, \ldots, S_b$.*

Proof. We prove it by induction on q. The base case when $q = d$ is trivial. Let $q \geq 2d-1$. If f is not a constant function then there exists $T \in \binom{[q]}{d-1}$ and $i, j \in [q] \setminus T$, such that $f(T \cup \{i\}) \neq f(T \cup \{j\})$. By induction, for the restricted function $\tilde{f} : \binom{[q]\setminus T}{d} \to \{0, 1\}$, there exists a cover $\tilde{\mathcal{S}} \subseteq \binom{[q]\setminus T}{d}$ of $[q] \setminus T$ such that $\tilde{f}$ is constant on $\tilde{\mathcal{S}}$ and $|\tilde{\mathcal{S}}| \leq \lceil \frac{q-1-(d-1)}{d-1} \rceil \leq \lceil \frac{q-1}{d-1} \rceil - 1$. Either $\mathcal{S} = \tilde{\mathcal{S}} \cup \{T \cup \{i\}\}$ or $\mathcal{S} = \tilde{\mathcal{S}} \cup \{T \cup \{j\}\}$ gives the required covering whose size is at most $\lceil \frac{q-1}{d-1} \rceil$.

The remaining case $d < q \leq 2d-2$ is handled similarly – in this case we take $T \in \binom{[q]}{q-d}$ in order to end up in the base case and get a cover of size 2, as desired. □

3.7.1 Reduction

We are now ready to give the reduction. We start with a multi-layered Label Cover $\mathcal{L}$ instance with parameters ℓ and r to be determined later. We reduce it to the hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$. The reduction is given as follows. Let $q := t(d-1)+1$, where $t \geq 1$ and $d \geq 2$ are integers.

Vertices $\mathcal{V}$: Each vertex v from layer i in the layered Label Cover instance $\mathcal{L}$ is replaced by a cloud of size q^{R_i} denoted by $C[v] := \{v\} \times [q]^{R_i}$. We refer to a vertex from the cloud $C[v]$ by a pair $(v, \mathbf{x})$ where $\mathbf{x} \in [q]^{R_i}$. The vertex set of the hypergraph is given by

$$\mathcal{V} = \cup_{v \in \cup_i X_i} C[v].$$

Hyperedges $\mathcal{E}$: There are two types of edges.

Type 1: For every $1 \leq \zeta < \eta \leq \ell$, every vertex $v \in X_\eta$ and every set of t neighbors $u_1, u_2, \ldots, u_t$ of v from layer X_ζ, we add the following hyperedges. Let $\pi_i = \phi_{u_i \to v}$ be the projection constraint between u_i and v for $1 \leq i \leq t$.

Let $\mathbf{x}^{i,j} \in [q]^{R_\zeta}$ be a set of td strings indexed by $i \in [d]$ and $j \in [t]$, let $\mathbf{y}^i \in [q]^{R_\eta}$ be $\lfloor \frac{d}{2} \rfloor$ strings indexed by $i \in [\lfloor \frac{d}{2} \rfloor]$. If it holds that for every $\beta \in [R_\eta]$ and all choices of $\alpha_j \in \pi_j^{-1}(\beta) \subseteq [R_\zeta]$

for $j \in [t]$ that

$$\left\{\mathbf{x}^{i,j}_{\alpha_j} \mid i \in [d], j \in [t]\right\} \bigcup \left\{\mathbf{y}^i_\beta \mid i \in [\lfloor d/2 \rfloor]\right\} = [q], \tag{3.1}$$

then we add add the hyperedge

$$\left\{(u_j, \mathbf{x}^{i,j})\right\}_{i\in[d], j\in[t]} \bigcup \left\{(v, \mathbf{y}^i)\right\}_{i\in[\lfloor d/2 \rfloor]}$$

to the hypergraph.

Type 2: For every $1 \le \eta \le \ell$, $v \in X_\eta$, in the cloud $C[v]$, add a hyperedge $\{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{td+\lfloor \frac{d}{2} \rfloor}\}$ if for all $\beta \in [R_\eta]$

$$\left\{\mathbf{y}^i_\beta \mid i \in [td + \lfloor d/2 \rfloor]\right\} = [q].$$

For comparison with the warm-up reduction for Rainbow$(4, 3, 2)$, observe that if we set $t = 1$, $d = 3$, and only take the Type 1 edges from the above reduction, we obtain reduction for Rainbow$(4, 3, 2)$. The sole purpose of the additional Type 2 edges used in this more general reduction is to force any 2-coloring of the resulting hypergraph to be somewhat balanced within each cloud (see further Claim 3.7.5 below). In the Rainbow$(4, 3, 2)$ case this was instead achieved via Fact 3.5.3.

3.7.2 Analysis

Lemma 3.7.3 (Completeness)**.** *If the Label Cover instance is satisfiable then the hypergraph $\mathcal{H}$ is rainbow q-colorable.*

Proof. Let $A : \bigcup_i X_i \to \bigcup_i [R_i]$ define the satisfiable labeling to the layered Label Cover instance. The rainbow q-coloring of the hypergraph is given by assigning a vertex $(v, \mathbf{x})$ with a color $\mathbf{x}_{A(v)}$.

To see that this is a rainbow q-coloring, consider any Type 1 hyperegde in the hypergraph between the clouds $C[u_1], C[u_2], \ldots, C[u_t]$ and $C[v]$ where $u_1, u_2, \ldots, u_t \in X_\zeta$ and $v \in X_\eta$. This hyperegde is of the form

$$\{(u_j, \mathbf{x}^{i,j})\}_{i\in[d], j\in[t]} \bigcup \{(v, \mathbf{y}^i)\}_{i\in[\lfloor \frac{d}{2} \rfloor]} \in \binom{\mathcal{V}}{td + \lfloor \frac{d}{2} \rfloor}$$

satisfying (3.1). By definition, χ assigns color $\mathbf{x}^{i,j}_{A(u_j)}$ to vertices $\{(u_j, \mathbf{x}^{i,j})\}$ for $i \in [d]$ and $j \in [t]$ and $\mathbf{y}^i_{A(v)}$ to $(v, \mathbf{y}^i)$ for $i \in [\lfloor \frac{d}{2} \rfloor]$. It is easy to see from (3.1) that these vertices get q distinct colors since $A(u_j) \in \pi_j^{-1}(A(v))$ for all $1 \le j \le t$.

Also, all Type 2 hyperedges trivially contain all the q colors. Hence χ is a valid rainbow q-coloring. □

We now prove the main soundness lemma.

Lemma 3.7.4 (Soundness)**.** *If $\ell \geq 8 \cdot (td)^{2td}$ and $\mathcal{H}$ is properly 2-colorable then there is an assignment A to the layered Label Cover instance which satisfies an $2^{-O(t^2d^2)}$ fraction of all constraints between some pair of layers X_i and X_j.*

In particular setting the layered Label Cover parameter $r \gg t^2d^2$ in Theorem 3.4.3, proves Theorem 3.2.1.

Proof. Assume for contradiction that the hypergraph $\mathcal{H}$ is 2-colorable. Fix a 2-coloring $\chi : \mathcal{V} \to \{0,1\}$ of the vertices of $\mathcal{H}$.

We have a following simple claim about the upper bound on the density of a color class in every cloud.

Claim 3.7.5. *For every $1 \leq \eta \leq \ell$, $v \in X_\eta$ and $b \in \{0,1\}$, in the cloud $C[v]$, the fraction of vertices colored with color b is at least $1/q$.*

Proof. Consider the class of shifts of $\mathbf{x} \in [c]^{[R_\eta]}$ defined as $[\mathbf{x}] := \{\mathbf{x}+\mathbf{1}, \mathbf{x}+\mathbf{2}, \ldots, \mathbf{x}+\mathbf{q}\}$, where $+$ is coordinate-wise addition (modulo q). Suppose for contradiction that the fraction of vertices in $C[v]$ that are colored b is less than $1/q$. Thus, there exists $\mathbf{x}$ such that $[\mathbf{x}]$ is monochromatic with color $1-b$. Since at least $1-1/q$ fraction of $C[v]$ is colored with color $1-b$, there exist a set of distinct strings $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{(t-1)+\lfloor \frac{d}{2} \rfloor} \notin [\mathbf{x}]$, such that $\chi(\mathbf{y}^i) = 1-b$ for all $i \in [(t-1)+\lfloor \frac{d}{2} \rfloor]$. But then $\{\mathbf{y}^i \mid i \in [(t-1)+\lfloor \frac{d}{2} \rfloor]\} \cup [\mathbf{x}]$ is a hyperedge of Type 2 in $\mathcal{H}$ which is monochromatic w.r.t. the coloring χ. □

For every $u \in X_i$, define functions $f_u : \binom{[q]}{d} \to \{0,1\}$, and $g_u : \binom{[q]}{d} \to \binom{[R_i]}{\leq d}$ as follows. For a $\sigma \in \binom{[q]}{d}$, in a cloud $C[u]$, consider the induced d-uniform hypergraph $\mathtt{H}^{R_i}_{\lfloor \frac{d}{2} \rfloor}(\sigma)$. Look at the coloring on these vertices induced by χ i.e. $\chi_{u,\sigma} : \sigma^{R_i} \to \{0,1\}$ defined by $\chi_{u,\sigma}(\mathbf{x}) = \chi((u,\mathbf{x}))$. By Lemma 3.5.2, there exists a color class, say $b \in \{0,1\}$, such that there exists a monochromatic hyperedge with color b in $\mathtt{H}^{R_i}_{\lfloor \frac{d}{2} \rfloor}(\sigma)$. Set $f_u(\sigma) = b$, where b is one such color class, breaking ties arbitrarily. Also, set $g_u(\sigma) = J_u$ if $J_u \subseteq [R_i]$ is the set of *noisy coordinates* in the b-monochromatic hyperedge, again breaking ties arbitrarily. If none of the coordinates are noisy in the hyperedge, then set $g_u(\sigma) = \{1\}$.

By Lemma 3.7.2, there exist for each variable u subsets $\sigma^u_1, \sigma^u_2, \ldots, \sigma^u_t \in \binom{[q]}{d}$ and a color $b_u \in \{0,1\}$ such that $f_u(\sigma^u_j) = b_u$ for all $j \in [t]$ and $\cup_{j=1}^t \sigma^u_j = [q]$. Write $S_u = (\sigma^u_1, \ldots \sigma^u_t) \in \binom{[q]}{d}^t$. Next, associate each layer i with the most frequent value among (S_u, b_u) over all vertices $u \in X_i$. For each layer $i \in [\ell]$, let $\tilde{X}_i$ be the set of vertices in X_i with the same label as layer i.

Let T be the total number of coverings of $\binom{[q]}{d}$ of size at most t. A trivial upper bound on T is $\binom{q}{d}^t \leq (td)^{td}$. Since $\ell \geq 8 \cdot (td)^{2td} \geq 8T^2$, there exists $m = 4T$ layers which are all associated with

the same pair (S, b), and in each of these $4T$ layers, at least a $1/(2T) = 2/m$ fraction of all variables are associated with (S, b). By the weak density property of the Label Cover instance, it follows that there exist two layers i and j such that the fraction of constraints between $\tilde{X}_i$ and $\tilde{X}_j$ is at least a $\frac{1}{16T^2}$ fraction of all constraints between X_i and X_j.

For the rest of the analysis, we set $U = \tilde{X}_i$ and $V = \tilde{X}_j$ and focus on satisfying the constraints between U and V. Let $S = \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ be the covering.

Labeling: We now proceed to define the labeling. For $u \in U$, define the set of candidate labels as $\mathcal{A}(u) = \cup_{i=1}^{t} g_u(\sigma_i)$. Then construct the labeling A as follows: for $u \in U$ let $A(u)$ be a random label from $\mathcal{A}(u)$ and for $v \in V$ pick a random $u \in U$ such that $u \sim v$ and let $A(v) = \phi_{u \to v}(A(u))$.

To analyze the quality of the labeling, we need the following two claims, which together form a generalization of the simpler Claim 3.6.3 used in the RAINBOW$(4, 3, 2)$ reduction – that if the neighbors $u \in U$ of $v \in V$ suggest many incompatible candidate labels for v, then a large fraction of vertices $(v, \mathbf{y})$ in $C[v]$ must not have color b (contradicting Claim 3.7.5).

Claim 3.7.6. *Let $v \in V$ and let $u_1, \ldots, u_t \in U$ be distinct neighbors of v and let $I_j = \phi_{u_j \to v}(g_{u_j}(\sigma_j))$. Let $I = \cup_{j=1}^{t} I_j$ and suppose that the I_j's are all pairwise disjoint. Then there exists a string $\mathbf{w} \in [q]^I$ such that for all $\mathbf{y} \in [q]^{R_V}$ with $\mathbf{y}_{|I} = \mathbf{w}$, the vertex $(v, \mathbf{y})$ does not have the color b.*

Proof. For all $j \in [t]$, by definition of I_j, there exist $\mathbf{x}^{1,j}, \ldots, \mathbf{x}^{d,j} \in \sigma_j^{R_U}$ such that

1. $(u_j, \mathbf{x}^{i,j})$ has color b for all $i \in [d]$, $j \in [t]$.
2. There exists $J_{u_j} \subseteq [R_U]$, $\phi_{u_j \to v}(J_{u_j}) = I_j$ such that for all $\alpha \notin J_{u_j}$ it holds that $\{\mathbf{x}_\alpha^{i,j}\}_{i \in [d]} = \sigma^j$ and for all $\alpha \in J_{u_j}$, we have $|\{\mathbf{x}_\alpha^{i,j}\}_{i \in [d]}| \geq \lceil \frac{d}{2} \rceil$. Moreover, there exists a subset $S_{u_j} \subseteq \sigma_j$ of size at least $\lceil d/2 \rceil$ such that for all $\alpha \in J_{u_j}$, the set $\{\mathbf{x}_\alpha^{i,j}\}_{i \in [d]}$ contains all the elements from S_{u_j}.

Consider any set of $\lfloor d/2 \rfloor$ strings $\mathbf{y}^1, \ldots, \mathbf{y}^{\lfloor d/2 \rfloor} \in [q]^{R_V}$ such that for all $\beta \in I_j$ it holds that

$$\{\mathbf{y}_\beta^i\}_{i \in [\lfloor d/2 \rfloor]} \supseteq \sigma_j \setminus S_{u_j}. \tag{3.2}$$

Note that $|\sigma_j \setminus S_{u_j}|$ is at at most $\lfloor d/2 \rfloor$ and hence there are $\mathbf{y}^1, \ldots, \mathbf{y}^{\lfloor d/2 \rfloor} \in [q]^{R_V}$ satisfying (3.2) for all $j \in [t]$. By construction it follows that these strings along with $\{\mathbf{x}^{i,j}\}_{i \in [d], j \in [t]}$ satisfy (3.1) and thus

$$\{(u_j, \mathbf{x}^{i,j})\}_{i \in [d], j \in [t]} \cup \{(v, \mathbf{y}^i)\}_{i \in [\lfloor d/2 \rfloor]},$$

forms a hyperedge of $\mathcal{H}$. It follows that at least one of $(v, \mathbf{y}^i)$ must have a color than different b. Let $H \subseteq [\lfloor d/2 \rfloor]$ be the set of indices i such that $(v, \mathbf{y}^i)$ is not colored b.

Suppose for the sake of contradiction, for all such $(v, \mathbf{y}^i)$ which is *not* colored b, there exists a string $\mathbf{z}^i$ agreeing with $\mathbf{y}^i$ at locations I i.e. $\mathbf{y}^i_{|I} = \mathbf{z}^i_{|I}$ such that the color of vertex $(v, \mathbf{z}^i)$ is b. One can check that $\{(u_j, \mathbf{x}^{i,j})\}_{i\in[d],j\in[t]} \cup \{(v, \mathbf{z}^i)\}_{i\in H} \cup \{(v, \mathbf{y}^i)\}_{i\in[\lfloor d/2 \rfloor]\setminus H}$ is a valid hyperedge with color b, a contradiction. Therefore there exists $i \in T$ such that for all strings $\mathbf{y} \in [q]^{R_V}$ with $\mathbf{y}_{|I} = \mathbf{y}^i_{|I}$, the vertex $(v, \mathbf{y})$ does not have color b. □

The following claim *rules out* that for many neighbors of v, the collection of candidate labelings $\phi_{u\to v}(\mathcal{A}(u))$ are pairwise disjoint.

Claim 3.7.7. *Let $B = t \cdot q^{td} \cdot \ln q$ and $v \in V$. Then for any B distinct neighbors $u_1, \ldots, u_B \in U$ of v, it holds that the label sets*

$$\phi_{u_j\to v}(\mathcal{A}(u_j)),$$

for $j \in [B]$ are not all pairwise disjoint.

Proof. Suppose for contradiction that B such neighbors exist where the corresponding label sets are all pairwise disjoint. Split them into $D := B/t$ groups of size t. By Claim 3.7.6 it follows that there exist D disjoint label sets $I_1, \ldots, I_D \subseteq [R_V]$ and strings $\mathbf{w}^1 \in [q]^{I_1}, \ldots, \mathbf{w}^D \in [q]^{I_D}$ such that $(v, \mathbf{y})$ does not have color b whenever $\mathbf{y}_{|I_j} = \mathbf{w}^j$ for some $j \in [D]$. Furthermore the sets I_j have size at most $|I_j| \le td$ so there at most a fraction $1 - q^{-td}$ of strings in $[q]^{R_V}$ differ from $\mathbf{w}^j$ on I_j. By the disjointness of the I_j's we thus have that the total fraction of vertices in the cloud $C[v]$ that have color b is at most

$$(1 - q^{-td})^D \le e^{-\frac{D}{q^{td}}}.$$

However, by Claim 3.7.5, for every $v \in V$ the cloud $C[v]$ must contain at least a fraction $\frac{1}{q}$ of the vertices with color b. Therefore, it follows that we must have $D/q^{td} \le \ln q$ and the claim follows. □

Using Claim 3.7.7 it is straightforward to obtain a lower bound on the quality of the randomized labeling.

Claim 3.7.8. *Let $B = t \cdot q^{td} \cdot \ln q$ be as in Claim 3.7.7. Then the randomized labeling satisfies in expectation at least a $\left(\frac{1}{t^2 B}\right)$ fraction of the constraints between U and V.*

Proof. The expected fraction of satisfied constraints involving $v \in V$ is at least

$$\begin{aligned}
&\mathbf{E}_{\substack{u_1,u_2\in U\\ u_1,u_2\sim v}}\left[\mathbb{P}_{A(u_1),A(u_2)}[\phi_{u_1\to v}(A(u_1)) = \phi_{u_2\to v}(A(u_2))]\right] \\
&\quad\geq \mathbf{E}_{\substack{u_1,u_2\in U\\ u_1,u_2\sim v}}\left[\frac{|\phi_{u_1\to v}(\mathcal{A}(u_1)) \cap \phi_{u_2\to v}(\mathcal{A}(u_2))|}{t^2}\right] \\
&\quad\geq \frac{1}{t^2}\mathbb{P}_{\substack{u_1,u_2\in U\\ u_1,u_2\sim v}}\left[\phi_{u_1\to v}(\mathcal{A}(u_1)) \cap \phi_{u_2\to v}(\mathcal{A}(u_2)) \neq \emptyset\right] \\
&\quad\geq \frac{1}{t^2}\cdot\frac{1}{B}
\end{aligned}$$

where the last inequality follows from Claim 3.7.7 and Claim 3.9.5. □

Thus, the constructed labeling satisfies a $\frac{1}{B}\cdot\left(\frac{1}{2Tt}\right)^2 = \frac{1}{tq^{td}\ln q}\frac{1}{4(td)^{2td}t^2} \geq 2^{-O(t^2d^2)}$ fraction of all constraints between the two layers, and this finishes the proof. □

3.7.3 Proof of Corollary 3.2.2

We start with the following simple claim:

Claim 3.7.9. *If* Rainbow$(k, q, 2)$ *is* **NP***-hard then* Rainbow$(k+1, q, 2)$ *is* **NP***-hard.*

Proof. Let $H(V, E)$ be an instance of Rainbow$(k, q, 2)$. Construct a $k+1$ uniform hypergraph $H_1(V_1, E_1)$ as follows: $V_1 = V \cup \{v_1, v_2, \ldots, v_{k+1}\}$ where $\{v_1, v_2, \ldots, v_{k+1}\}$ are the extra set of vertices not in V. For every hyperedge $e \in E$ add $(e \cup v_i)$ to E_1 for all $1 \leq i \leq k+1$. Also add $\{v_1, v_2, \ldots, v_{k+1}\}$ to E_1. This finishes the reduction. Now, if H is rainbow q-colorable, then coloring $\{v_1, v_2, \ldots, v_{k+1}\}$ with q different colors and keeping the colors of vertices V as given by the rainbow q-coloring of H gives a rainbow q-coloring of H_1. On the other hand, if H_1 is 2-colorable then the restriction of the 2-coloring to V gives a proper 2-coloring of H. □

Proof of Corollary 3.2.2. Let $t = \left\lfloor \frac{1}{2}\sqrt{k} \right\rfloor$ and set d to be the largest integer such that $u := td + \lfloor d/2 \rfloor \leq k$. Observe that $d \leq 2\sqrt{k}$ and that $k - u \leq t + 1$. Applying Theorem 3.2.1 and $k - u$ repetitions of Claim 3.7.9, we have that Rainbow$(k, q, 2)$ is **NP**-hard for $q = t(d-1) + 1 = u - \lfloor d/2 \rfloor - t + 1$. The difference between k and q is

$$k - q = k - u + \lfloor d/2 \rfloor + t - 1 \leq \lfloor d/2 \rfloor + 2t \leq 2\lfloor\sqrt{k}\rfloor.$$

□

3.8 A Generalized Hypergraph Gadget

In order to prove the hardness of almost rainbow coloring, we will work with the following family of hypergraphs:

Definition 3.8.1 (The hypergraph $\mathtt{RH}_t^n(\Sigma)$)**.** *For an alphabet Σ of size p and parameters $0 \le t \le n$, let $\mathtt{RH}_t^n(\Sigma)$ be the p-uniform hypergraph with vertex set Σ^n where p vertices $\mathbf{x}^1, \ldots, \mathbf{x}^p \in \Sigma^n$ form a hyperedge iff*

$$|\{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^p\}| = p \tag{3.3}$$

for at least $n - t$ different coordinates $i \in [n]$.

The set of noisy coordinates *for a hyperedge is the set of $\le t$ values of i where* (3.3) *does not hold.*

The graph $\mathtt{RH}_1^n(\{0,1,2\})$ is very similar to, but not exactly the same as the hypergraph $\mathtt{H}_1^n(\{0,1,2\})$ used in Section 3.6. The difference is that in $\mathtt{H}_1^n(\{0,1,2\})$, we required the single noisy coordinate of a hyperedge to have at least 2 different colors, whereas in $\mathtt{RH}_1^n(\{0,1,2\})$ the noisy coordinate may have only a single color. This difference is mostly superficial, and we could have defined $\mathtt{H}_1^n(\{0,1,2\})$ differently to make it match $\mathtt{RH}_1^n(\{0,1,2\})$ (but the additional edges contained in $\mathtt{RH}_1^n(\{0,1,2\})$ would not have been used in the reduction for $\textsc{Rainbow}(4,3,2)$).

Note that $\mathtt{RH}_t^n(\mathbb{Z}_p)$ has very large "non-junta-like" independent sets containing almost half the vertices, e.g. the set of all strings containing more than $n/p + t$ zeros is independent and has size $1/2 - o(1)$ for fixed t and p as $n \to \infty$.

Generalizing Lemma 3.5.2, we want to obtain lower bounds on the chromatic number of $\mathtt{RH}_t^n(\mathbb{Z}_p)$ that grow with t.

Our main combinatorial result is the following.

Theorem 3.8.2. *For every odd prime p, $c \ge 1$, and $n \ge p^2 c$, the chromatic number of $\mathtt{RH}_{p^2c}^n(\mathbb{Z}_p)$ is at least $c + 1$.*

The proof is given in Section 3.8.2. This bound is likely far from tight (for one thing, note that for fixed t, the value of c even decreases with p).

3.8.1 Topology Background

In this subsection, we cover some necessary topological notions and theorems that will be used in the proof of Theorem 3.8.2. The curious reader is referred to Matoušek's excellent book [Mat07] for proofs and further details.

We use $S^d = \{\mathbf{x} \in \mathbb{R}^{d+1} \mid \|\mathbf{x}\| = 1\}$ to denote the unit d-sphere.

Definition 3.8.3 (Free $\mathbb{Z}_p$-action)**.** *For a topological space X, a $\mathbb{Z}_p$-action on X is a collection $\Phi = \{\psi_g\}_{g \in \mathbb{Z}_p}$ of homeomorphisms $X \mapsto X$ such that for every $g \in \mathbb{Z}_p$, the map ψ_g is continuous,*

and for every $g, h \in \mathbb{Z}_p$, we have that $\psi_g \circ \psi_h = \psi_{gh}$. Moreover, the action is free *is for every nonzero $g \in \mathbb{Z}_p$, and every $\mathbf{x} \in X$, we have $\psi_g(\mathbf{x}) \neq \mathbf{x}$.*

We shall mainly talk about $\mathbb{Z}_p$-actions on a sphere S^k, where p is a prime and k is odd. In this case, every nonzero element of $\mathbb{Z}_p$ has essentially the same kind of action, i.e., for every nonzero $g \in \mathbb{Z}_p$, and every $\mathbf{x} \in S^k$, we have

1. $\psi_g(\mathbf{x}) \neq \mathbf{x}$.
2. $(\psi_g)^p(\mathbf{x}) = \mathbf{x}$.

Hence, we shall just pick an arbitrary nonzero element g of $\mathbb{Z}_p$, and define $L \mid \psi_g$. By slight abuse of notation, we shall call L the free $\mathbb{Z}_p$-action, also since it determines how every other element acts.

Let $\omega_p = \exp(2\pi i/p)$ be the primitive p'th root of unity in $\mathbb{C}$. In our uses, p will always be some fixed prime and we omit the subscript and simply write ω. Let $\phi : \mathbb{R}^{2n} \to \mathbb{C}^n$ be the bijection $\phi(\mathbf{x}) = (x_{2j-1} + ix_{2j})_{j\in[n]}$ (i.e., we clump together pairs of coordinates in $\mathbb{R}^{2n}$).

Fact 3.8.4. *For every odd prime p and integer $n \geq 1$ the map $L : S^{2n-1} \to S^{2n-1}$ defined by $L(\mathbf{x}) = \phi^{-1}(\omega\phi(\mathbf{x}))$ is a free $\mathbb{Z}_p$-action on S^{2n-1}.*

It is important that the sphere in the above fact is an odd sphere as only $\mathbb{Z}_2$ acts freely on even spheres. We use the following generalization of the classic Borsuk-Ulam Theorem.

Theorem 3.8.5 ([Woj96], or [Mat07] Theorem 6.3.3)**.** *Let p be an odd prime, and let $S = S^{(p-1)d+1}$. Let $f : S \to \mathbb{R}^d$ be a continuous map, and L be any free $\mathbb{Z}_p$-action on S. Then, there is some point $\mathbf{x} \in S$ such that*

$$f(\mathbf{x}) = f(L\mathbf{x}) = f(L^2\mathbf{x}) = \cdots = f(L^{p-1}\mathbf{x})$$

With the above general theorem at hand, we can draw the same covering conclusion as in the Lusternik-Schnirelmann theorem on covering (see, for example, [Mat07], Exercise 6.3.4).

Corollary 3.8.6. *For any covering of $S^{(p-1)(c-1)+1}$ by c closed sets $A_1, \ldots, A_c$, there is an $i \in [c]$ and a point $\mathbf{x} \in S^{(p-1)(c-1)+1}$ such that $\mathbf{x}, L\mathbf{x}, \ldots, L^{p-1}\mathbf{x}$ are all contained in A_i.*

3.8.2 Bound on the Chromatic Number

In this section we give a lower bound on the chromatic number of $\mathtt{RH}^n_t(\mathbb{Z}_p)$.

The proof is basically an adaptation of Bárány's proof [Bár78] of Lovász's theorem [Lov78] on the chromatic number of Kneser graphs. In order to carry this out, one needs to adapt an equivalent formulation of Gale's theorem.

Before proceeding with the proof, we develop some notation that will be useful. For an even integer d, we have the bijection $\phi : \mathbb{R}^d \to \mathbb{C}^{d/2}$ and the free $\mathbb{Z}_p$-action L from Fact 3.8.4 acting on S^{d-1} by taking $\mathbf{z}$ to $\phi^{-1}(\omega\phi(\mathbf{z}))$. Define a bilinear function $M : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^2$ by

$$M(\mathbf{w}, \mathbf{z}) = \phi^{-1}\Big(\Big\langle \phi(\mathbf{w}), \overline{\phi(\mathbf{z})} \Big\rangle\Big)$$

where $\langle \cdot, \cdot \rangle$ is the usual inner product over $\mathbb{C}^{d/2}$ and by slight abuse of notation we view ϕ also as a bijection between $\mathbb{R}^2$ and $\mathbb{C}$. For brevity, we will parameterize this function by the first variable and denote $M_{\mathbf{w}}(\mathbf{z}) = M(\mathbf{w}, \mathbf{z})$. The key properties to note are:

(M1) M is bilinear and in particular for $L\mathbf{z} = \phi^{-1}(\omega\phi\mathbf{z})$ we have

$$M_{\mathbf{w}}(L\mathbf{z}) = \phi^{-1}(\omega\langle \phi(\mathbf{w}), \overline{\phi(\mathbf{z})}\rangle)$$

which equals both $M_{L\mathbf{w}}(\mathbf{z})$ and $LM_{\mathbf{w}}(\mathbf{z})$ (where, just like with ϕ, we view L as also acting on $\mathbb{R}^2$ by rotating every point counter-clockwise by $2\pi/p$ around the origin).

(M2) For $w \neq \overline{0}$, we have that $M_{\mathbf{w}}$ is a full rank map, i.e., $\text{image}(M_{\mathbf{w}}) = \mathbb{R}^2$.

Next, we define a function $T : \mathbb{R}^d \times \mathbb{R}^d \to \{\perp, 0, \ldots, p-1\}$ which is almost like a p-way threshold function as follows: Denote by $\ell_j \in \mathbb{R}^2$ the ray $\left\{ \left(\alpha \cos(\frac{2\pi j}{p}), \alpha \sin(\frac{2\pi j}{p}) \right) \mid \alpha \geq 0 \right\}$ for $0 \leq j \leq p-1$. For $j = 0, \ldots, p-1$, let r_j denote the open region between ℓ_j and $\ell_{j+1 \bmod p}$. We define:

$$T_{\mathbf{w}}(\mathbf{z}) = \begin{cases} j & \text{if} \quad M_{\mathbf{w}}(\mathbf{z}) \in r_j \text{ for some } j \\ \perp & \text{otherwise} \end{cases}$$

Note that $T_{\mathbf{w}}$ almost acts like a threshold function except it does not deal with "ties" – in case of a tie, $T_{\mathbf{w}}$ is simply defined as $\perp$. The most important property of $T_{\mathbf{w}}$ is that it interacts well with L:

Claim 3.8.7. *For all integers $j \geq 0$, and all $\mathbf{w}, \mathbf{z} \in \mathbb{R}^d$, it holds that*

$$T_{L^j\mathbf{w}}(\mathbf{z}) = T_{\mathbf{w}}(L^j\mathbf{z}) = \begin{cases} (T_{\mathbf{w}}(\mathbf{z}) + j) \bmod p & \text{if } T_{\mathbf{w}}(\mathbf{z}) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

Proof. By Property (M1), $M_{L^j\mathbf{w}}(\mathbf{z}) = M_{\mathbf{w}}(L^j\mathbf{z})$ equals $M_{\mathbf{w}}(\mathbf{z})$ rotated $2\pi j/p$ radians counter-clockwise around the origin. Thus if $M_{\mathbf{w}}(\mathbf{z}) \in r_k$ for some k then $M_{\mathbf{w}}(L^j\mathbf{z}) \in r_{k+j \bmod p}$ (and thus $T_{\mathbf{w}}(L^j\mathbf{z}) = (k+j) \bmod p$) and similarly if $M_{\mathbf{w}}(\mathbf{z}) \in \ell_k$ then $M_{\mathbf{w}}(L^j\mathbf{z}) \in \ell_{k+j \bmod p}$ (and thus $T_{\mathbf{w}}(L^j\mathbf{z}) = \perp$). □

Let $\mathbf{u} : \mathbb{R}_{\geq 0} \to S^{d-1}$ be the normalized moment curve in $\mathbb{R}^d$, i.e., $\mathbf{u}(s) = \gamma(s)/\|\gamma(s)\|_2$ where $\gamma(s) = (1, s, s^2, \ldots, s^{d-1})$. One important property to note is that for any subset $S \subset \mathbb{R}$ such that $|S| \leq d$, we have that the vectors $\{\mathbf{u}(s)\}_{s \in S}$ are linearly independent. We have the following basic fact.

Claim 3.8.8. *For every* $\mathbf{w} \in S^{d-1}$, $T_{\mathbf{w}}(\mathbf{u}(s)) = \perp$ *for less than* pd *different values of* $s \in \mathbb{R}$.

Proof. Suppose for contradiction that at least pd points $M_{\mathbf{w}}(\mathbf{u}(s))$ lie on the p rays $\ell_0, \ell_1, \ldots, \ell_{p-1}$. Of these at least d lie on a line. Since any subset of at most d $\mathbf{u}(s)$'s are in general position, this contradicts Property (M2) that image$(M_{\mathbf{w}}) = \mathbb{R}^2$. □

The choice of $\mathbf{u}$ is somewhat arbitrary – any continuous curve whose image under $M_{\mathbf{w}}$ intersects the ℓ_k's in a finite number of points would work. With these facts in hand, we are ready to prove Theorem 3.8.2.

Theorem (Theorem 3.8.2 restated)**.** . For every odd prime p and $c, n \geq p^2 c$, the chromatic number of $\mathtt{RH}^n_{p^2 c}(\mathbb{Z}_p)$ is at least $c + 1$.

Proof. Let $d := (p-1)(c-1)+2$. We construct a set of n points $\mathcal{V} = \{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^n\}$ on S^{d-1}, one for every index in $[n]$, as follows:

$$\mathbf{v}^i = L^{i-1}\mathbf{u}(i)$$

The key property of these points is that they give a correspondence between points in S^{d-1} and the vertices in $\mathtt{RH}^n_{p^2 c}(\mathbb{Z}_p)$ (i.e., $\mathbb{Z}_p^n$) in the following sense. We say that $\mathbf{x} \in \mathbb{Z}_p^n$ *matches* $\mathbf{w} \in S^{d-1}$ if

$$\mathbf{x}_i = T_{\mathbf{w}}(\mathbf{v}^i)$$

for all $i \in [n]$ such that $T_{\mathbf{w}}(\mathbf{v}^i) \neq \perp$. Now, given a coloring $\chi : \mathbb{Z}_p^n \to [c]$, we define a covering $\{A_1, A_2, \ldots, A_c\}$ of S^{d-1} as follows: for every point $\mathbf{w} \in S^{d-1}$, put $\mathbf{w} \in A_j$ if there is a $\mathbf{x} \in \mathbb{Z}_p^n$ that matches $\mathbf{w}$ and has $\chi(\mathbf{x}) = j$. Observe that it is possible that a point $\mathbf{a}$ belongs to many A_j's and that every point $\mathbf{a} \in S^{d-1}$ is matched by at least one $\mathbf{x} \in \mathbb{Z}_p^n$ (so that this is indeed a cover).

Next, we observe that the sets $A_1, \ldots, A_c$ are closed.

Claim 3.8.9. *Each* A_j *is closed.*

Proof. Note that the map $\mathbf{w} \mapsto M_{\mathbf{w}}(\mathbf{v}^i)$ is continuous for each $i \in [n]$. Thus for every $\mathbf{w} \in S^{d-1}$, there is some $\epsilon > 0$ such that for every $\mathbf{w}'$ within distance ϵ of w it holds that

$$\text{for every } i \in [n], \text{ either } T_{\mathbf{w}'}(\mathbf{v}^i) = T_{\mathbf{w}}(\mathbf{v}^i) \text{ or } T_{\mathbf{w}}(\mathbf{v}^i) = \perp \tag{3.4}$$

Now let $\mathbf{w}$ be a point in the closure of A_j. Taking $\epsilon > 0$ as above, there is an $\mathbf{w}' \in A_j$ within distance ϵ of $\mathbf{w}$ satisfying (3.4). But any $\mathbf{x}$ that matches such an $\mathbf{w}'$ also matches $\mathbf{w}$ and in particular it follows that $\mathbf{w} \in A_j$ and hence $\overline{A_j} = A_j$. □

Thus, $\{A_1, \ldots, A_c\}$ is a cover of $S^{d-1} = S^{(p-1)(c-1)+1}$ by c closed sets, so by Corollary 3.8.6 there is a point $\mathbf{w}^\star \in S^{d-1}$ such that $\mathbf{w}^\star, L\mathbf{w}^\star, \ldots, L^{p-1}\mathbf{w}^\star$ are all covered by the same set. Suppose that this set is A_1. For each $j \in \mathbb{Z}_p$, let $\mathbf{x}^j$ be any vertex of $\mathtt{RH}^n_{p^2c}(\mathbb{Z}_p)$ that has $\chi(\mathbf{x}^j) = 1$ and that matches $L^j\mathbf{w}^\star$. By construction these p vertices have the same color and all that remains to prove is the following claim.

Claim 3.8.10. $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{p-1}$ *form a hyperedge in* $\mathtt{RH}^n_{p^2c}(\mathbb{Z}_p)$

Proof. To prove this, it suffices to show that for every $i \in [n]$ such that $T_{\mathbf{w}^\star}(\mathbf{v}^i) \neq \perp$, we have $\{\mathbf{x}^0_i, \mathbf{x}^1_i, \ldots, \mathbf{x}^{p-1}_i\} = \mathbb{Z}_p$, since the number of $i \in [n]$ s.t. $T_{\mathbf{w}^\star}(\mathbf{v}^i) = \perp$ is at most $pd \leq p^2c$. To prove this, first note that by definition $\mathbf{x}^j_i = T_{\mathbf{L}^j w^\star}(\mathbf{v}^i)$ for all i such that $T_{\mathbf{w}^\star}(\mathbf{v}^i) \neq \perp$. By Claim 3.8.7 it thus follows that $\mathbf{x}^j_i = (\mathbf{x}^0_i + j) \bmod p$. □

Thus any $\chi : V(\mathtt{RH}^n_{p^2c}(\mathbb{Z}_p)) \to [c]$ must have a monochromatic hyperedge and the proof of Theorem 3.8.2 is done. □

3.9 Almost Rainbow Hardness

In this section we prove Theorem 3.2.5. Recall from Section 3.4.2 that $B(q, d, c)$ is the worst case covering size t such that every function $g : \binom{[q]}{d} \to [c]$ has a monochromatic cover of size t.

Theorem 3.9.1 (Theorem 3.2.5 restated)**.** *For every $d \geq c \geq 2$ and $t \geq 2$ such that d and t are primes and d is odd, let $q = t(d - c + 1) + c - 1$ and $k = td$. Then* AlmostRainbow$(k, q, q - d, c)$ *is* **NP***-hard (provided $d < \lfloor q/2 \rfloor$).*

In the rest of this section, fix $t := \frac{q-c+1}{d-c+1}$ which is equal to $B(q, d, c)$ using Theorem 3.4.6, as t is a prime number for the setting of q in the above theorem.

For this result, we do not need the full power of layered Label Cover, but use Theorem 3.4.3 with $\ell = 2$ layers (i.e., normal Label Cover). To simplify notation in this case, we refer to the two vertex sets as $U = X_1$ and $V = X_2$, and denote the alphabet size of U by R and the alphabet size of V by L. In other words, our starting point is a Label Cover instance on variables $U \cup V$ with alphabet sizes R and L of size $2^{O(r)}$ and soundness $2^{-\Omega(r)}$ for some parameter r that will be chosen to a large enough constant as a function of q, d and c later.

We reduce it to a hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$ using the reduction given as follows

Vertices $\mathcal{V}$: Each vertex $u \in U$ in the Label Cover instance $\mathcal{L}$ is replaced by a cloud of size q^R denoted by $C[u] := \{u\} \times [q]^R$. We refer to a vertex from the cloud $C[u]$ by a pair $(u, \mathbf{x})$ where

$\mathbf{x} \in [q]^R$. The vertex set of the hypergraph is given by

$$\mathcal{V} = \cup_{u \in U} C[u].$$

Hyperedges $\mathcal{E}$: For every vertex $v \in V$ and every set of t neighbors $u_1, u_2, \ldots, u_t$ of v from U, we add the following hyperedges:

Let $\pi_i = \phi_{u_i \to v}$ be the projection constraint between u_i and v for $1 \leq i \leq t$. Let $\mathbf{x}^{i,j} \in [q]^R$ be a set of td strings indexed by $i \in [d]$ and $j \in [t]$. If it holds that for every $\beta \in [L]$ and all choices of $\alpha_j \in \pi_j^{-1}(\beta) \subseteq [R]$ for $j \in [t]$ that

$$\left| \left\{ \mathbf{x}^{i,j}_{\alpha_j} \mid i \in [d], j \in [t] \right\} \right| \geq q - d \tag{3.5}$$

then we add add the hyperedge

$$\{(u_j, \mathbf{x}^{i,j})\}_{i \in [d], j \in [t]} \in \binom{\mathcal{V}}{td}.$$

Lemma 3.9.2 (Completeness)**.** *If the Label Cover instance is satisfiable then the hypergraph $\mathcal{H}$ is rainbow $(q, q-d)$-colorable.*

Proof. Let $A : U \cup V \to [R] \cup [L]$ define the satisfiable labeling to the Label Cover instance. The rainbow $(q, q-d)$-coloring of the hypergraph is given by assigning a vertex $(u, \mathbf{x})$ with a color $\mathbf{x}_{A(u)}$.

To see that this is a rainbow $(q, q-d)$-coloring, consider any hyperegde in the hypergraph between the clouds $C[u_1], C[u_2], \ldots, C[u_t]$ where $u_1, u_2, \ldots, u_t \in U$ and $v \in V$ be their common neighbor. This hyperegde is of the form

$$\{(u_j, \mathbf{x}^{i,j})\}_{i \in [d], j \in [t]} \in \binom{\mathcal{V}}{td}$$

satisfying the (3.5). By definition, χ assigns color $\mathbf{x}^{i,j}_{A(u_j)}$ to vertices $\{(u_j, \mathbf{x}^{i,j})\}$ for $i \in [d]$ and $j \in [t]$. It is easy to see from (3.5) that these vertices get $q - d$ distinct colors since $A(u_j) \in \pi_j^{-1}(A(v))$ for all $1 \leq j \leq t$.

Hence χ is a valid rainbow $(q, q-d)$-coloring. □

We now prove the main soundness lemma.

Lemma 3.9.3 (Soundness)**.** *If $\mathcal{H}$ is properly c-colorable then there is an assignment A to the Label Cover instance which satisfies an $\frac{1}{d^4 c^3 t^4 2^{td \log q}}$ fraction of all constraints between U and V.*

Proof. Assume for contradiction that the hypergraph $\mathcal{H}$ is c-colorable. Fix a c-coloring $\chi : \mathcal{V} \to [c]$ of the vertices of $\mathcal{H}$.

Set $h = d^2 c$. For every $u \in U$, define functions $f_u : \binom{[q]}{d} \to [c]$, and $g_u : \binom{[q]}{d} \to 2^{[R]}$ as follows. For a $\sigma \in \binom{[q]}{d}$, in a cloud $C[u]$, consider the induced d-uniform hypergraph $\mathtt{RH}^R_h(\sigma)$. Look

at the coloring on these vertices induced by χ i.e. $\chi_{u,\sigma} : \sigma^R \to [c]$ defined by $\chi_{u,\sigma}(\mathbf{x}) = \chi((u, \mathbf{x}))$. By Theorem 3.8.2, there exists a color class, say $b \in [c]$, such that there exists a monochromatic hyperedge with color b in $\mathtt{RH}_h^R(\sigma)$. Set $f_u(\sigma) = b$, where b is one such color class, breaking ties arbitrarily. Also, set $g_u(\sigma) = J$ if $J \subseteq [R]$ are the set of *noisy coordinates* in the b-monochromatic hyperedge, again breaking ties arbitrarily. If none of the coordinates are noisy in the hyperedge, then set $g_u(\sigma) = \{1\}$.

Recall that $t = B(q, d, c)$, so by definition, for each variable u, subsets $\sigma_1^u, \sigma_2^u, \ldots, \sigma_t^u \in \binom{[q]}{d}$ and a color $b_u \in [c]$ such that $f_u(\sigma_j^u) = b_u$ for all $j \in [t]$ and $\cup_{j=1}^t \sigma_j^u = [q]$. Write $S_u = (\sigma_1^u, \ldots \sigma_t^u) \in \binom{[q]}{d}^t$ and label a variable u as (S_u, b_u). Let T be the total number of coverings of $\binom{[q]}{d}$ of size at most t. A trivial upper bound on T is $\binom{q}{d}^t \leq 2^{td \log q}$. By an averaging argument, there is a label (S, b) such that at least a $\frac{1}{cT}$ fraction of all constraints of the Label Cover instance are incident upon vertices $u \in U$ with label (S, b). Let that subset be U'. Thus, between U' and V, we have at least a $\frac{1}{cT}$ fraction of all constraints.

For the rest of the analysis, we focus on satisfying the constraints between U' and V. Let $S = \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ be the covering.

We now proceed to define the labeling. For $u \in U'$, define the set of candidate labels as $\mathcal{A}(u) = \cup_{i=1}^t g_u(\sigma_i)$. Then construct the labeling A as follows: for $u \in U'$ let $A(u)$ be a random label from $\mathcal{A}(u)$ and for $v \in V$ pick a random $u \in U'$ such that $u \sim v$ and let $A(v) = \phi_{u \to v}(A(u))$ (if v has no neighbors in U', set $A(v)$ arbitrarily).

The quality of this labeling hinges on Claim 3.9.4 below.

Claim 3.9.4. *Let $v \in V$ and $u_1, \ldots, u_t \in U'$ be distinct neighbors of v and write $I_j = \phi_{u_j \to v}(g_{u_j}(\sigma^j))$. Then, the I_j's are not pairwise disjoint.*

It is possible that v has fewer than t neighbors in U' but in this case the claim is vacuously true.

Proof. Suppose for contradiction that the I_j's are pairwise disjoint. By the definition of I_j, there exist $\mathbf{x}^{1,j}, \ldots, \mathbf{x}^{d,j} \in \sigma_j^{R_U}$ such that

1. $(u_j, \mathbf{x}^{i,j})$ has color b for all $i \in [d]$, $j \in [t]$.
2. For all $\beta \notin I_j$ and $\alpha_j \in \phi_{u_j \to v}^{-1}(\beta)$ it holds that $\{\mathbf{x}_{\alpha_j}^{i,j}\}_{i \in [d]} = \sigma^j$.

From the pairwise disjointness of I_j's, it follows that these strings satisfy (3.5) for every $\beta \in [L]$ and for all choices of $\alpha_j \in \phi_{u_j \to v}^{-1}(\beta) \subseteq [R]$ for $j \in [t]$. Thus,

$$\{(u_j, \mathbf{x}^{i,j})\}_{i \in [d], j \in [t]},$$

forms a hyperedge of $\mathcal{H}$ which is monochromatic w.r.t. χ, a contradiction to the fact that χ was a valid c-coloring. □

We also need the following simple claim:

Claim 3.9.5. *For any set family $\mathcal{S} \subseteq 2^{[n]}$ such that no Δ of them are pairwise disjoint,*

$$\mathbb{P}_{s_1,s_2\in\mathcal{S}}[s_1 \cap s_2 \neq \emptyset] \geq \frac{1}{\Delta - 1}.$$

Proof. Define a graph $G(\mathcal{S}, E)$ on $\mathcal{S}$ where $s_1 \sim s_2$ if they do not intersect. By the property of $\mathcal{S}$, G does not contain a clique of size Δ. By Turán's theorem, the number of edges in G is at most

$$|E| \leq \frac{\Delta - 2}{\Delta - 1} \cdot \frac{|\mathcal{S}|^2}{2}.$$

Now, the probability that $s_1, s_2 \in \mathcal{S}$ do not intersect is equivalent to saying $(s_1, s_2) \in E$. Thus, the probability is at most

$$\frac{2|E|}{|\mathcal{S}|^2} \leq 2 \cdot \frac{\Delta - 2}{\Delta - 1} \cdot \frac{|\mathcal{S}|^2}{2} \cdot \frac{1}{|\mathcal{S}|^2} = 1 - \frac{1}{\Delta - 1}$$

□

Using Claim 3.9.4 it is straightforward to obtain a lower bound on the quality of the randomized labeling.

Claim 3.9.6. *The randomized labeling satisfies in expectation at least a $\frac{1}{h^2t^3}$ fraction of the constraints between U' and V.*

Proof. The expected fraction of satisfied constraints involving $v \in V'$ is at least

$$\begin{aligned}
&\mathbf{E}_{\substack{u_1,u_2\in U' \\ u_1,u_2\sim v}} \left[\mathbb{P}_{A(u_1),A(u_2)}[\phi_{u_1\to v}(A(u_1)) = \phi_{u_2\to v}(A(u_2))]\right] \\
&\quad \geq \mathbf{E}_{\substack{u_1,u_2\in U \\ u_1,u_2\sim v}} \left[\frac{|\phi_{u_1\to v}(\mathcal{A}(u_1)) \cap \phi_{u_2\to v}(\mathcal{A}(u_2))|}{(ht)^2}\right] \\
&\quad \geq \frac{1}{(ht)^2} \mathbb{P}_{\substack{u_1,u_2\in U \\ u_1,u_2\sim v}} [\phi_{u_1\to v}(\mathcal{A}(u_1)) \cap \phi_{u_2\to v}(\mathcal{A}(u_2)) \neq \emptyset] \\
&\quad \geq \frac{1}{(ht)^2} \cdot \frac{1}{t}
\end{aligned}$$

where the last inequality follows from Claim 3.9.4 and Claim 3.9.5. □

To summarize, the constructed labeling satisfies a $\frac{1}{h^2t^3} \cdot \frac{1}{cT}$ fraction of all constraints between the U and V, and we are done. □

Proof of Theorem 3.2.5. The proof follows from Lemma 3.9.2 and Lemma 3.9.3 and by setting r such that the soundness of the Label Cover is $2^{-\Omega(r)} \ll \frac{1}{d^4c^3t^3 2^{td\log q}}$. □

3.10 Discussion and open problems

We have shown improved hardness of finding 2-colorings in rainbow colorable hypergraphs, and of finding c-colorings of almost rainbow colorable hypergraphs. There are a number of interesting open questions. For the Rainbow problem, the smallest open case is currently Rainbow$(7, 6, 2)$ ([GS19]). It would be interesting to know whether this problem is **NP**-hard or not.

In some sense, the reason why we only get hardness for 2-colorings is that the soundness argument contains steps along the following lines: (i) no cloud can be almost monochromatic, (ii) therefore since there are only two colors, each cloud contains a constant fraction of vertices of each color, (iii) in order for the randomized labeling to fail, the involved clouds would need to have a very small fraction of vertices of some color. Here, step (ii) is clearly not true for colorings with more than 2 colors.

On the combinatorial side, the two most interesting problems here are:

(1) What is the independence number of the $\mathtt{RH}_t^n(p)$? [*Conjecture:* $(1/2 - o(1))k^m$]

(2) What is the chromatic number of the $\mathtt{RH}_t^n(p)$?

The second and third problems are stated separately despite the apparent similarity because we believe that the approaches to them should be completely different and the answers are completely unrelated. For instance, we believe that the independence number is more related to generalizing Kletiman's Isodiametric Theorem [Kle66], whereas the chromatic number is more closely related to the *discrete Borsuk graph*, whose chromatic number is not known. Answers to the above problems would not only help us understand the hardness of rainbow coloring better, but would also help in understanding the underlying objects, which could be applicable elsewhere and are interesting in their own right.

Chapter 4
A spectral bound on hypergraph discrepancy

The main aim of this section is to give a spectral condition that is sufficient for the discrepancy of a regular hypergraph to be small. This is proved via the partial coloring approach while using some combinatorial properties of the hypergraph that are given by this spectral condition. This immediately implies, via an old proof technique of Kahn and Szemerédi, that for every t, the discrepancy of a random t-regular hypergraph on n vertices and $m \geq n$ edges is almost surely $O\left(\sqrt{t}\right)$. Previously, a result of this form was proved by Ezra and Lovett [EL15] who show that the discrepancy of a random t-regular hypergraph on n vertices and $m \geq n$ edges is $O(\sqrt{t \log t})$ almost surely as t grows. More recently, Bansal and Meka [BM19] showed that for random t-regular hypergraphs on n vertices and m edges, the discrepancy is $O\left(\sqrt{t}\right)$ almost surely provided $t = \Omega\left((\log\log m)^2\right)$. To state our result formally, we make some definitions.

4.1 Introduction

Let $\mathcal{H} = (V, E)$ be a hypergraph, with V as the set of vertices, and $E \subseteq 2^V$ as the set of (hyper)edges. Let $\mathcal{X} = \{\chi : V \to \{\pm 1\}\}$, be the set of ± 1 colorings of V, and for $\chi \in \mathcal{X}$, and $e \in E$, denote $\chi(e) := \sum_{v \in e} \chi(v)$. The discrepancy of $\mathcal{H}$, denoted by $\mathrm{disc}(\mathcal{H})$ is defined as:

$$\mathrm{disc}(\mathcal{H}) := \min_{\chi \in \mathcal{X}} \max_{e \in E} |\chi(e)|.$$

We call a hypergraph *t-regular* if every vertex is present in exactly t hyperedges. These will be the main focus of this paper. For a hypergraph $\mathcal{H}$, let $M = M(\mathcal{H})$ be the $|E| \times |V|$ incidence matrix of $\mathcal{H}$, i.e., M has rows indexed by E, columns indexed by V, and entries are $M(e, v) = 1$ if $v \in e$ and 0 otherwise. We will use $\|\cdot\|$ to denote the Euclidean norm. The main result is the following:

Theorem 4.1.1. *Let $\mathcal{H}$ be a t-regular hypergraph on n vertices and m edges with M as its incidence and let $\lambda = \max_{v \perp \bar{1}, \|v\|=1} \|Mv\|$. Then*

$$\mathrm{disc}(\mathcal{H}) = O\left(\sqrt{t} + \lambda\right).$$

Moreover, there is an $\tilde{O}((\max\{n,m\})^7)$ time algorithm that takes the hypergraph $\mathcal{H}$ as input and outputs the coloring with the above guarantee.

4.1.1 Background

The study of hypergraph discrepancy, which seems to have been first defined in a paper of Beck [Bec81], has led to some very interesting results with diverse applications (see, for example [Mat99], [Cha00]). One of the most interesting open problems in discrepancy theory is what is commonly known as the Beck-Fiala conjecture, regarding the discrepancy of general t-regular hypergraphs.

Conjecture 4.1.2 (Beck-Fiala conjecture)**.** *For a t-regular hypergraph $\mathcal{H}$, we have*

$$\mathrm{disc}(\mathcal{H}) = O(\sqrt{t}).$$

Although this conjecture is usually stated for *bounded degree* hypergraphs (as opposed to regular ones), this is not really an issue. One can always add hyperedges containing just a single vertex and make it regular, which increases the discrepancy of the original hypergraph by at most one. Beck and Fiala [BF81] also proved that for any t-regular hypergraph $\mathcal{H}$,

$$\mathrm{disc}(\mathcal{H}) \leq 2t - 1.$$

This is more commonly known as the Beck-Fiala theorem. Essentially the same proof can be done a bit more carefully to get a bound of $2t-3$ (see [BH97]). Given Conjecture 4.1.2, it is perhaps surprising that the best upper bound, due to Bukh [Buk16], is "stuck at" $2t-\log^* t$ for large enough t.

It is possible that one of the reasons that the discrepancy upper bounds are so far away from the conjectured bound (assuming it's true) is our inability to handle many 'large' hyperedges. Indeed, if one is offered the restriction that each hyperedge is also of size $O(t)$ (regular and 'almost uniform'), then a folklore argument using the Lovász Local Lemma shows that the discrepancy is bounded by $O(\sqrt{t \log t})$. The proof of Theorem 4.1.1 also relies on being able to avoid dealing with large edges (which are few, if any, in number).

4.1.2 Discrepancy in random settings

Motivated by the long-standing open problem of bounding discrepancy of general t-regular hypergraphs, Ezra and Lovett [EL15] initiated the study of discrepancy of *random* t-regular hypergraphs. By random t-regular hypergraph, we mean the hypergraph sampled by the following procedure: We fix n vertices V and m (initially empty) hyperedges E. Each vertex in V chooses t (distinct)

hyperedges in E uniformly and independently to be a part of. They showed that if $m \geq n$, then the discrepancy of such a hypergraph is almost surely $O(\sqrt{t \log t})$ as t grows. The proof idea is the following: First observe that most of the hyperedges have size $O(t)$. For the remaining large edges, one can delete one vertex from every hyperedge and make them pairwise disjoint. This allows one to apply a folklore Lovász Local Lemma based argument, but with a slight modification which makes sure that the large edges have discrepancy at most 2. More recently, Bansal and Meka [BM19] reduced the discrepancy bound to $O(\sqrt{t})$ almost surely as long as $t = \Omega\left((\log \log n)^2\right)$ for all m and n. A corollary of Theorem 4.1.1 states that one can get the bound of $O(\sqrt{t})$ for every (not necessarily growing) $t = t(n)$ as n grows and $m \geq n$. More formally,

Corollary 4.1.3. *There is an absolute constant $C > 0$ such that the following holds: Let $\mathcal{H}_t$ be a random t-regular hypergraph on n vertices and $m \geq n$ hyperedges where $t = o(\sqrt{m})$. Then,*

$$\mathbb{P}\left(\operatorname{disc}(\mathcal{H}_t) \leq C\sqrt{t}\right) \geq 1 - o(1)$$

The theorem that implies Corollary 4.1.3 from Theorem 4.1.1 is the following:

Theorem 4.1.4. *Let M be the incidence matrix of a random t-regular set system on n vertices, where $t = o(\sqrt{m})$, and $m \geq n$ edges. Then with probability at least $1 - n^{\Omega(1)}$,*

$$\max_{v \perp \overline{1}, \|v\|=1} \|Mv\| = O\left(\sqrt{t}\right).$$

A couple of remarks here: First, observe that it suffices to prove Theorem 4.1.4 for $m = n$. Indeed, let M and N be random $m \times m$ and $m \times n$ random matrices (m ¿ n) respectively distributed by choosing t random 1's in each column independently. Notice that the distribution of N is exactly the same as that of the first n columns of M. Then, setting M_n to be the matrix consisting of the first n columns of M, we observe that $\lambda(M_n) \leq \lambda(M)$. Second, we point out that $t = o(\sqrt{m})$ is just a limitation of the proof technique in [FKS89] (also see [BFSU98]) that we use to prove this theorem. Although we believe that Theorem 4.1.4 should hold for all $t < m$, we do not make any attempt to verify this, especially since the result of Bansal and Meka [BM19] already takes care of the discrepancy of random hypergraphs in this case. Although many variations of Theorem 4.1.4 are known and standard, one needs to verify it for our setting too. It should come as no surprise that the proof follows that of Kahn and Szemerédi's [1] in [FKS89], which is postponed to Section 4.3.2.

[1] [FKS89] is combination of two papers that prove the same result upto a constant factor: one by Friedman using the so-called trace method, and the other by Kahn and Szemerédi using a more combinatorial approach which is flexible enough to be easily adapted here.

4.1.3 The partial coloring approach

Most of the bounds and algorithms on hypergraph discrepancy proceed via a *partial coloring approach*. In general, a partial coloring approach [Bec81] works by coloring a fraction of the (still uncolored) vertices in each step, while ensuring that no edge has discrepancy more than the desired bound. Perhaps the most famous successful application of this is Spencer's celebrated 'six standard deviations' result [Spe85], which gives a bound of $6\sqrt{n}$ for any hypergraph on n vertices and n edges. The original proof of Spencer was not algorithmic, i.e., it did not give an obvious way to take as input a hypergraph on n vertices and n edges, and efficiently output a coloring that achieves discrepancy $O(\sqrt{n})$. In fact, Alon and Spencer([AS00], §14.5) suggested that such an algorithm is not possible. However, this was shown to be incorrect by Bansal [Ban10] who showed an efficient algorithm to do the same task. However, the analysis of this algorithm still relied on the (non-algorithmic) discrepancy bound of $6\sqrt{n}$. Later, Lovett and Meka [LM15] gave a 'truly constructive' proof of the fact that the discrepancy is $O(\sqrt{n})$. This proof did not rely on any existing discrepancy bounds and the novel and simple analysis proved to be extremely influential. The proof of Theorem 4.1.1 will rely on a somewhat technical feature of the main partial coloring from this work. More recently, a result due to Rothvoss [Rot17] gives a simpler proof of the same $O(\sqrt{n})$ bound, which is also constructive, and more general.

4.1.4 Proof sketch

The proof of Theorem 4.1.1 is proved via the aforementioned partial coloring approach. The main source of inspiration is a later paper of Spencer [Spe88], which computes the discrepancy of the projective plane (i.e., the hypergraph where the vertices are the points and the hyperedges are the lines of $\mathrm{PG}(2, q)$) upto a constant factor. A more general bound was also obtained by Matoušek [Mat95], who upper bounds the discrepancy of set systems of bounded VC-dimension (note that the projective plane has VC-dimension 2).

We also use the aforementioned result of Lovett and Meka [LM15] heavily, in particular, the partial coloring theorem. Informally, this says that one can 'color' roughly an α fraction of the hypergraph with real numbers in $[-1, 1]$ so that (1) at least half the vertices get colors 1 or -1 and (2) every edge e has discrepancy $O(\sqrt{e})$. We now sketch the proof.

Consider the following 'dream approach' using partial coloring: In every step, one colors an α fraction of vertices. Suppose that at the start, every edge has size $O(t)$ and that each step of partial

coloring colors exactly an α fraction of the remaining uncolored vertices (i.e., these vertices are colored from $\{-1, 1\}$). Then the discrepancy of an edge e is at most $O\left(\sum_i \sqrt{\alpha^i |e|}\right) = O(\sqrt{t})$. Of course, this is too much to hope for, since some edges can potentially be large, and more importantly, there is no guarantee on how much of each edge gets colored in this partial coloring procedure.

This is precisely where the spectral condition on M saves us. One can establish standard combinatorial 'pseudorandomness' properties of $\mathcal{H}$ in terms of λ. In particular, if λ is small, then an α fraction of $V(\mathcal{H})$ take up an α fraction of most edges. This means, intuitively, that in the partial coloring approach, if one colors an α fraction of the vertices, then most of the edge sizes will have also reduced by an α fraction. The partial coloring method of Lovett and Meka (and, curiously, none of the older ones) also allows one to color in such a way that $\Omega(n)$ edges can be made to have discrepancy zero in each step. This allows one to maintain that in every round of the partial coloring, the edges that don't behave according to the 'dream approach', i.e., those that are too large (i.e., $\Omega(t)$) or don't reduce by an α fraction can be made to have discrepancy *zero* in the next step. Thus, most other edges reduce in size by an α fraction. This lets one not have to deal with the discrepancy of these 'bad' edges until they become small.

4.2 Proof of Theorem 4.1.1

4.2.1 Preliminaries and notation

We will need the aforementioned partial coloring theorem due to Lovett and Meka:

Theorem 4.2.1 ([LM15])**.** *Given a family of sets $M_1, \ldots, M_m \subseteq [n]$, a vector $x_0 \in [-1, 1]^n$, positive real numbers $c_1, \ldots, c_m$ such that $\sum_{i \in [m]} \exp\left(-c_i^2/16\right) \leq n/16$, and a real number $\delta \in [0, 1]$, there is a vector $x \in [-1, 1]^n$ such that:*

1. *For all $i \in [m]$, $\langle x - x_0, \mathbb{1}_{M_i} \rangle \leq c_i \sqrt{|M_i|}$.*
2. *$|x_i| \geq 1 - \delta$ for at least $n/2$ values of i.*

Moreover, this vector x can be found in $\tilde{O}((m + n)^3 \delta^{-2})$ time.

Lovett and Meka initially gave a randomized algorithm for the above. It has since been made deterministic [LRR17].

A technical remark:

The reason we use the Lovett-Meka partial coloring, as opposed to Beck's partial coloring is not just the algorithmic aspect that the former offers, but also because it also offers the technical condition:

$$\sum_{i\in[m]} \exp\left(-c_i^2/16\right) \leq n/16.$$

This means one can set $\Omega(n)$ edges to have discrepancy 0. To compare, we first state Beck's partial coloring lemma (for reference, see [Mat99]):

Theorem 4.2.2 (Beck's partial coloring lemma)**.** *Given a family of sets $M_1, \ldots, M_m \subseteq [n]$, and positive real numbers $c_1, \ldots, c_m$ such that $\sum_{i\in[m]} g(c_i) \leq n/5$, where*

$$g(x) = \begin{cases} e^{-x^2/9} & x > 0.1 \\ \ln(1/x) & x \leq 0.1 \end{cases}$$

there is a vector $x \in \{-1, 0, 1\}^n$ such that:

1. *For all $i \in [m]$, $\langle x, \mathbb{1}_{M_i}\rangle \leq c_i\sqrt{|M_i|}$.*
2. *$|x_i| = 1$ for at least $n/2$ values of i.*

If one ignores the algorithmic aspect, Beck's partial coloring, while assigning vertices to $\{-1, 1, 0\}$ (instead of $[-1, 1]$, thus making it a 'partial coloring' in the true sense) only guarantees that $\Omega\left(\frac{n}{\log t}\right)$ edges can be made to have discrepancy 0. Although [LM15] did not really need this particular advantage, they do mention that this feature could potentially be useful elsewhere. This seemingly subtle advantage turns out to be crucial in the proof of Theorem 4.1.1, where we set $\Omega(n)$ edges (that will be called 'bad' and 'dormant' edges) to have discrepancy 0.

Henceforth, let V and E denote the vertices and edges of our hypergraph respectively. We will need a 'pseudorandomness' lemma that informally states that an α fraction of vertices takes up around an α fraction of most edges:

Lemma 4.2.3. *For any $S \subseteq V$ with $|S| = \alpha n$ where $\alpha \in (0, 1)$ and a positive real number K, there is a subset $E' \subset E$ of size at most $K^{-2} \cdot \alpha n$ such that for every $e \notin E'$, we have $||e \cap S| - \alpha|e|| \leq K\lambda$, where $\lambda = \max_{v\perp\overline{\mathbf{1}}, \|v\|=1} \|Mv\|$.*

Proof. Consider a vector $v \in \mathbb{R}^n$ where $v(i) = 1 - \alpha$ for $i \in S$ and $-\alpha$ otherwise. Clearly, $v \in \mathbf{1}^\perp$ and so

$$\|Mv\|^2 \leq \lambda^2 \cdot \|v\|^2 = \lambda^2\alpha(1-\alpha)n. \tag{4.1}$$

On the other hand, $Mv(e) = (1-\alpha)|e \cap S| - \alpha|e \setminus S| = |e \cap S| - \alpha|e|$, and so

$$\|Mv\|^2 = \sum_e (|e \cap S| - \alpha|e|)^2. \tag{4.2}$$

Putting (4.1) and (4.2) together, we get that there at most $K^{-2} \cdot \alpha n$ edges e such that $||e \cap S| - \alpha|e|| \geq K\lambda$. □

Since this proof is via partial coloring, let us use i to index the steps of the partial coloring. For a partial coloring $\chi : V \to [-1, 1]$, we call the set of vertices u for which $|\chi(u)| < 1$ as *uncolored*. Let us use V^i to denote the still uncolored vertices at step i and for an edge $e \in E$, let us denote $e^i := e \cap V^i$. In every step, we invoke Theorem 4.2.1 setting $\delta = \frac{1}{n}$ to get the partial coloring, so will have $|V^i| \leq 2^{-i}n$. Let $t' := \max\{t, \lambda\}$ [2].

We call an edge *dormant* at step i if $|e^i| > 100t'$. Let us call an edge *bad* in step i if $\left||e^i| - 2^{-i}|e|\right| \geq 10\lambda$. Edges that are neither dormant not bad are called *good*. Finally, we say that e is *dead* in step i if $|e^i| \leq 100\lambda$.

Informally, the roles of these sets are as follows: In the partial coloring step i, we ensure that an edge e edges only get nonzero discrepancy if it is good, i.e., if $|e^i|$ is close to what is expected and is not too large. Even dead edges can be good or bad, and we will not distinguish them while coloring the vertices. However, in the analysis we will break the total discrepancy accumulated by e into two parts: Before it is dead and after. The main point is to bound the discrepancy gained before it becomes dead. After it becomes dead, we simply bound the discrepancy incurred since by its remaining size, i.e., at most 100λ.

First, we make two easy observations:

Claim 4.2.4. *If $|V^i| = 2^{-i}n$, then at step i, the number of dormant edges is at most $\frac{1}{100}2^{-i}n$.*

Proof. This is just Markov's inequality, using the fact that the average edge size is $\frac{|V^i|t}{m} \leq \frac{|V^i|t'}{m}$. □

Claim 4.2.5. *If $|V^i| = 2^{-i}n$, then at step i, the number of bad edges is at most $\frac{1}{100}2^{-i}n$.*

Proof. This is by setting $K = 10$ and $\alpha = 2^{-i}$ in Lemma 4.2.3. □

[2] In fact, we may assume w.l.o.g. that $\lambda \leq t$ and so $t' = t$ since in the other case, the Beck-Fiala Theorem gives us that the discrepancy is $O(t) = O(\lambda)$. However, this is not needed and the techniques here also handle this case with this minor change.

4.2.2 Partial coloring using Lemma 4.2.3

Proof of Theorem 4.1.1. Setting $V^0 = V$, we proceed by partial coloring that colors exactly half the remaining uncolored vertices at each stage. For a step $i \geq 0$, suppose that $|V^i| = 2^{-i}n$. We will describe a partial coloring given by $\chi_i : V^i \to [-1, 1]$ that colors half the vertices of V^i.

For $\ell \geq 1$, let $A_\ell := \{e \in E \mid |e| \in [100 \cdot 2^\ell t', 100 \cdot 2^{\ell+1} t')\}$, and $A_0 := \{e \in E \mid |e| < 200t'\}$. Observe that the edges in A_ℓ for $\ell \geq 1$ are either bad or dormant in steps $i < \ell$. Also observe that $|A_\ell| \leq \frac{2^{-\ell}}{100} n$ for $\ell \geq 1$, Define constants $\{c_e\}_{e \in E}$ as follows:

$$c_e = \begin{cases} 4\sqrt{2\ln\left(\frac{1}{2^{\ell-i}}\right)} & \text{if } e \in A_\ell \text{ for } \ell \geq 1 \text{ is good} \\ 4\sqrt{\ln\left(\frac{200t'}{2^{-i}|e|}\right)} & \text{if } e \in A_0 \text{ is good} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{B} = \mathcal{B}^i$ and $\mathcal{D} = \mathcal{D}^i$ denote the bad and dormant edges respectively. We handle the edges in A_0 and $E \setminus A_0$ separately. For edges in $E \setminus A_0$, we have:

$$\begin{aligned} \sum_{e \in E \setminus A_0} e^{-\frac{c_e^2}{16}} &\leq \sum_{e \in E \setminus (\mathcal{B} \cup \mathcal{D} \cup A_0)} e^{-\frac{c_e^2}{16}} + |\mathcal{B}| + |\mathcal{D}| \\ &\leq \sum_{1 \leq \ell \leq i} \sum_{e \in A_\ell} e^{2\ln(2^{\ell-i})} + \frac{2^{-i}n}{50} \\ &= \sum_{1 \leq \ell \leq i} |A_\ell| 2^{2(\ell-i)} + \frac{2^{-i}n}{50} \\ &\leq \frac{n}{100} \sum_{\ell \leq i} 2^{-\ell} \cdot 2^{2\ell-2i} + \frac{2^{-i}n}{50} \\ &= \frac{2^{-i}n}{100} \sum_{\ell \leq i} 2^{\ell-i} + \frac{2^{-i}n}{50} \\ &\leq \frac{2^{-i}n}{25}. \end{aligned}$$

The second inequality above follows from Claim 4.2.4 and Claim 4.2.5. For the other case, we have

$$\sum_{e \in A_0} e^{-\frac{c_e^2}{16}} \leq \sum_{e \in A_0} e^{\ln\left(\frac{2^{-i}|e|}{200t}\right)} = \frac{2^{-i}}{200} \sum_{e \in E} \frac{|e|}{t'} = \frac{2^{-i}n}{200}.$$

Here we have used the fact that since the hypergraph is t-regular, we have $\sum_{e \in E} |e| = nt \leq nt'$. Putting these together, we have

$$\sum_{e\in E} e^{-\frac{c_e^2}{16}} \leq \frac{2^{-i}n}{200} + \frac{2^{-i}n}{50} \leq \frac{|V^i|}{20}.$$

Therefore, Theorem 4.2.1 guarantees that there is a fractional coloring $\chi_i : V^i \to [-1,1]$ such that

1. $|\chi_i(v)| \geq 1 - \frac{1}{n}$ for at least half of V^i.
2. All the bad and dormant edges get discrepancy 0.
3. A good and live edge e gets discrepancy at most $c_e\sqrt{|e^i|}$.

Finally, we pick an arbitrary subset of all the vertices v such that $|\chi_i(v)| \geq 1 - \frac{1}{n}$ of size exactly $(1/2)\cdot|V^i|$ and round them to the nearest integer. It is easy to see that since every edge has size at most n, this rounding, over all the steps of the partial coloring adds discrepancy of at most 1 for every edge. This completes step i of the partial coloring and we are left with $2^{-(i+1)}n$ uncolored vertices for the next step.

For an edge e, let i be a round where e had incurred non-zero discrepancy and e^i was not dead. Since only good edges incur nonzero discrepancy, $|e^i| = 2^{-i}|e| \pm 10\lambda$. Since e is also not dead at step i, we must have that $|e^i| \geq 100\lambda$. This gives us that $2^{-i}|e| \geq 90\lambda$ and therefore $(1/2)\cdot 2^{-i}|e| \leq |e_i| \leq 2\cdot 2^{-i}|e|$. So, if $e \in A_\ell$ where $\ell \geq 1$, the total discrepancy incurred by e at step i without the rounding step is at most

$$4\sqrt{2\ln(1/2^{j-i})e^i} \leq 8\sqrt{200\ln(1/2^{\ell-i})\cdot(2^{\ell-i})\cdot t'}.$$

Here, we have used the fact that $|e| \leq 100\cdot 2^{\ell+1}t'$. If $e \in A_0$, the discrepancy incurred by e at step i without the rounding is at most

$$4\sqrt{2\ln\left(\frac{200t'}{2^{-i}|e|}\right)e^i} \leq 8\sqrt{200\ln(1/2^{-i})\cdot(2^{-i})\cdot t'}.$$

Therefore, the discrepancy of an edge $e \in A_\ell$ for $\ell \geq 0$ until it becomes dead is at most

$$\sum_{i\geq\ell} 8\sqrt{200\ln(1/2^{\ell-i})\cdot(2^{\ell-i})\cdot t'} = O(\sqrt{t'}) = O(\sqrt{t}+\lambda).$$

Here we have used the fact that $t' = \max\{t,\lambda\}$. Finally, rounding the color of every vertex to its nearest integer increases the discrepancy by at most 1. When the edge becomes dead, we simply bound its discrepancy by its size $O(\lambda)$.

It remains to check that each of the $O(\log n)$ stages of partial coloring can be done in time $\tilde{O}((m+n)^3 n^2)$, and the constants $\{c_e\}_{e\in E}$ take $\tilde{O}(mn)$ time to compute at each stage, thus establishing the algorithmic part. □

4.3 Proof of Theorem 4.1.4

4.3.1 A martingale inequality

We will state a martingale inequality that we will use in the proof of Theorem 4.1.4. A sequence of random variables $X_0, X_1, \ldots, X_n$ martingale with respect to another sequence of random variables $Z_0, Z_1, \ldots, Z_n$ such that for all $i \in [n-1]$, we have $X_i = f_i(Z_1, \ldots Z_i)$ for some function f_i, and $\mathbf{E}[X_{i+1}|Z_i, \ldots, Z_1] = X_i$.

A martingale is said to have the C-bounded difference property if $|X_{i+1} - X_i| \leq C$.

The variance of a martingale is the quantity:

$$\sigma^2 = \sum_{i\in[n-1]} \sup_{(Z_1,\ldots,Z_i)} \mathbf{E}[(X_{i+1} - X_i)^2 | Z_1, \ldots, Z_i].$$

We get good large deviation inequalities for martingales with bounded differences and variances (see, for example, [CL06], Theorem 6.3 and Theorem 6.5). For a martingale $X_0, X_1, \ldots, X_n$ with respect to $Z_0, Z_1, \ldots, Z_n$, with the C-bounded difference property and variance σ^2, we have

$$\mathbb{P}(|X_n - X_0| \geq \lambda) \leq e^{-\frac{t^2}{2(\sigma^2 + C\lambda/3)}}. \tag{4.3}$$

4.3.2 Proof of Theorem 4.1.4

We shall now prove Theorem 4.1.4. Recall that we only need to prove the case where $m = n$, As mentioned before, we adapt the proof technique of Kahn and Szemerédi for our random model (also see [BFSU98]). We have that the regularity is $t \ll m^{1/2}$.

We shall prove that for every x, and y such that $\|x\| = \|y\| = 1$ and $x \perp \overline{1}$, we have that $|y^t M x| \leq O(\sqrt{t})$. First, we 'discretize' our problem by restricting x to belong to the ϵ-net

$$T := \left\{ x \in \left(\frac{\epsilon}{\sqrt{m}}\mathbb{Z}\right)^m \mid \|x\| \leq 1 \text{ and } x \perp \overline{1} \right\}$$

and y belonging to

$$T' := \left\{ y \in \left(\frac{\epsilon}{\sqrt{m}}\mathbb{Z}\right)^m \mid \|y\| \leq 1 \right\}$$

for a small enough constant ϵ.

Claim 4.3.1 ([FKS89], Proposition 2.1)**.** *If for every $x \in T$, and $y \in T'$, we have that $\|y^t M x\| \leq \alpha$, then we have that for every $z \in \mathbb{R}^m$ such that $\|z\| = 1$, we have that $\|Mz\| \leq (1-3\epsilon)^{-1}\alpha$.*

Proof. Let $z = \text{argmax}_{\|z\|=1} \|Mz\|$. We shall use the fact that there are $x \in T$, and $y \in T'$ such that $\|x - z\| \leq \epsilon$, and $\left\|y - \frac{Mz}{\|Mz\|}\right\| \leq \epsilon$. With this in mind, we have:

$$\begin{aligned}\|Mz\| &= \left\langle \frac{Mz}{\|Mz\|}, Mz \right\rangle = \langle y + w_1, M(x + w_2)\rangle \\ &= y^t Mx + \langle w_1, Mx\rangle + \langle y, Mw_2\rangle + \langle w_1, Mw_2\rangle.\end{aligned}$$

Where $|w_1|, |w_2| \leq \epsilon$. We note that each of the terms $\langle w_1, Mx\rangle$ and $\langle y, Mw_2\rangle$, and $\langle w_1, Mw_2\rangle$ are upper bounded by $\epsilon\|Mz\|$, and $\langle w_1, Mw_2\rangle \leq \epsilon^2\|Mz\|$. Combining this, and using the fact that $\epsilon^2 \leq \epsilon$, we have

$$\|Mz\| \leq (1-3\epsilon)^{-1} y^t Mx \leq (1-3\epsilon)^{-1}\alpha.$$

□

So now, will need to only union bound over $T \cup T'$. It is not hard to see that each of these has size at most $|T|, |T'| \leq \left(\frac{C_v}{\epsilon}\right)^m$ for some absolute constant C_v.

Indeed, we have:

$$\begin{aligned}|T| &\leq \left(\frac{\sqrt{m}}{\epsilon}\right)^m \text{Vol}\left\{x \in \mathbb{R}^m \mid \|x\| \leq 1 + \epsilon\right\} \\ &\leq \left(\frac{\sqrt{m}}{\epsilon}\right)^m \cdot \frac{1}{\sqrt{\pi m}} \left(\frac{2\pi e}{m}\right)^{m/2} (1+\epsilon)^m \\ &\leq \left(\frac{C_v}{\epsilon}\right)^m\end{aligned}$$

for some constant C_v.

We split the pairs $[m] \times [m] = L \cup \overline{L}$ where $L := \{(u, v) \mid |x_u y_v| \geq \sqrt{t}/m\}$, which we will call 'large entries' and write our quantity of interest:

$$\sum_{(u,v)\in[m]\times[m]} x_u M_{u,v} y_v = \sum_{(u,v)\in L} x_u M_{u,v} y_v + \sum_{(u,v)\in\overline{L}} x_u M_{u,v} y_v.$$

For the large entries: For a set of vertices $A \subset [m]$ and a set of edges $B \subset [m]$, let us denote $I(A, B)$ to be the number of vertex-edge incidences in A and B. Let us use $\mu(A, B) := \mathbf{E}[|I(A, B)|]$.

Lemma 4.3.2. *There is a constant C such that, for every set A of vertices and every set B of hyperedges where $|A| \leq |B|$, we have that with probability at least $1 - m^{-\Omega(1)}$, $I := |I(A, B)|$ and $\mu := \mu(A, B)$ satisfy at least one of the following:*

1. $I \leq C\mu$

2. $I \log(I/\mu) \leq C|B| \log(m/|B|)$.

This lemma is sufficient to show that the large pairs do not contribute too much, as shown by the following lemma, which is the main part of the proof of Kahn and Szemerédi.

Lemma 4.3.3 ([FKS89], Lemma 2.6, [BFSU98], Lemma 17)**.** *If the conditions given in Lemma 4.3.2 are satisfied, then* $\sum_{(u,v)\in L} |x_u M_{u,v} y_v| = O(\sqrt{t})$ *for all* $x, y \in T$.

Notice that since we are bounding $\sum_{(u,v)\in L} |x_u M_{u,v} y_v| = O(\sqrt{t})$, which is much stronger than what we really need, it is okay to consider both x and y from T.

Proof of Lemma 4.3.2. First, we observe that it is enough to consider $|B| \leq m/2$, since otherwise, $|I(A,B)| \leq d|A| \leq 2\mu(A,B)$. Let $\mathcal{B}_i(a,b)$ denote the event that there is an A of size a and a B of size b which do not satisfy either of the conditions (with a fixd constant C to be specified later) and $|I(A,B)| = i$. Before, we prove the lemma, let us make some observations, which (in hindsight) help us compute the probabilities much easier. Let A be a set of a vertices and B be a collection of b edges, such that $a \leq b \leq m/2$.

The point here is that we basically want to evaluate the sum:

$$\mathbb{P}\left(\bigcup_{a,b,i} \mathcal{B}_i(a,b)\right) \leq \sum_i \mathbb{P}\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right)$$
$$= \sum_{i \leq \log^2 m} \mathbb{P}\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right) + \sum_{i \geq \log^2 m} \mathbb{P}\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right).$$

The first observation is that every term in the second sum is small. Towards this, we have the straightforward claim.

Claim 4.3.4. *For a set of vertices* A *and edges* B *and a set of possible incidences* $J \subset A \times B$*, we have that* $\mathbb{P}(I(A,B) = J) \leq \left(\frac{2t}{m}\right)^{|J|}$.

Proof. W.L.O.G, let $A = \{1, \ldots, a\}$, and for $i \in A$, let $t_i = I(\{i\}, B)$. We have that:

$$\mathbb{P}(I(A,B) = J) = \prod_{i \in A} \frac{\binom{m-b}{t-t_i}}{\binom{m}{t}} \leq \prod_{i \in A} 2\frac{(m-b)^{t-t_i}}{(t-t_i)!}\frac{t!}{m^t} \leq \prod_{i \in A} \left(\frac{2t}{m}\right)^{t_i} \leq \left(\frac{2t}{m}\right)^{|J|}.$$

□

Here, the first inequality uses the fact that $t = o(\sqrt{m})$. Therefore, we have:

$$\mathbb{P}\left(\mathcal{B}_i(a,b)\right) \leq \binom{m}{a}\binom{m}{b}\binom{ab}{i}\left(\frac{2t}{m}\right)^i \leq \binom{m}{b}^2\left(e\frac{abt}{mi}\right)^i \leq \binom{m}{b}^2\left(\frac{\mu}{i}\right)^i (e)^i.$$

If $i \geq 2e\mu$ and $i \geq \log^2 m$, this probability is at most $2^{2m} \cdot 2^{-\log^2 m} \ll m^{-\Omega(\log m)}$. Thus

$$\sum_{i \geq \log^2 m} \mathbb{P}\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right) \leq \sum_{a,b} \sum_{i \geq \log^2 m} \mathbb{P}\left(\mathcal{B}_i(a,b)\right) \leq m^{-\Omega(\log m)}.$$

It remains to deal with the sum $\sum_{i \leq \log^2 m} \Pr\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right)$. For these summands, we have that if $|I(A,B)| \leq \log^2 m$ and $I \log(I/\mu) > Cb \log(m/b)$, then

$$I \log m \geq I \log(I/\mu) > Cb \log(m/b) \geq Cb.$$

and so $Cb \leq \log^3 m$. The first inequality above comes from the observation that $I \leq ab$ and so $I/\mu \leq m/t \leq m$. Now, using that $I \log m \geq Cb \log(m/\log^3 m)$, we have that $I \geq Cb/2$.

Therefore, we only need to evaluate the sum:

$$\begin{aligned}
\sum_{i=Cb/2}^{\log^2 m} \mathbb{P}\left(\mathcal{B}_i(a,b)\right) &\leq \binom{m}{a}\binom{m}{b} \sum_{i=Cb/2}^{\log^2 m} \binom{ab}{i}\left(\frac{10et}{m}\right)^i \leq \binom{m}{b}^2 \sum_{i=Cb/2}^{\log^2 m}\left(\frac{10e^2abt}{im}\right)^i \\
&\leq \log^2 m \left(\frac{em}{b}\right)^{2b}\left(\frac{20e^2at}{Cm}\right)^{Cb/2} \\
&= m^{2b-Cb/2} b^{-2b} a^{Cb/2} t^{Cb/2} (20e^2)^{2b} \\
&\leq m^{2b-Cb/4} b^{Cb/2-2b} (20e^2)^{2b} \\
&= m^{-\Omega(b)}.
\end{aligned}$$

We have used the fact that $t = o(\sqrt{m})$, $b \geq a$ and $b \leq \log^3 m$. Thus union bounding over $\log^3 m$ many values of a and b, we have $\sum_{a,b \leq \log^3 m} \sum_{i \leq \log^2 m} \mathbb{P}\left(\bigcup_{a,b} \mathcal{B}_i(a,b)\right) = m^{-\Omega(1)}$. □

For the small entries: Bounding the contribution from the small entries is much easier. The analysis given here is slightly different to the one given in [FKS89] and [BFSU98]. However, it does not make much of a difference, and is still, essentially, the same large deviation inequality. We will first compute the expected value of the quantity of interest using the following claim:

Claim 4.3.5. *We have that:*

$$\left|\sum_{(u,v) \in \overline{L}} x_u y_v\right| \leq \frac{m}{\sqrt{t}}.$$

Proof. Since $\sum x_i = 0$, we have $\left(\sum x_i\right)\left(\sum y_i\right) = \sum_{(u,v)\in L} x_u y_v + \sum_{(u,v)\in \overline{L}} x_u y_v = 0$ or

$$\left|\sum_{(u,v) \in \overline{L}} x_u y_v\right| = \left|\sum_{(u,v) \in L} x_u y_v\right|.$$

To bound this, we note that

$$1 = \left(\sum x_u^2\right)\left(\sum y_u^2\right) \geq \sum_{(u,v)\in L} x_u^2 y_v^2 \geq \frac{\sqrt{t}}{m}\left|\sum_{(u,v)\in L} x_u y_v\right| = \frac{\sqrt{t}}{m}\left|\sum_{(u,v)\in \overline{L}} x_u y_v\right|.$$

which gives us what we want. □

Given Claim 4.3.5 above, we can easily compute the expectation:

$$\mathbf{E}\left[\sum_{(u,v)\in\overline{L}} x_u M_{u,v} y_v\right] = \frac{t}{m}\sum_{(u,v)\in\overline{L}} x_u y_v \in [-\sqrt{t}, \sqrt{t}].$$

Claim 4.3.6. *We have that with high probability, $\sum_{(u,v)\in\overline{L}} x_u M_{u,v} y_v = O(\sqrt{t})$.*

Proof. We set up a martingale and use the method of bounded variances. Let us write the quantity that we wish to estimate as

$$X := \sum_{(u,v)\in B} x_u M_{u,v} y_v.$$

We imagine M being sampled one column at a time, and in each column, t entries are sampled. For column i, let us denote these by $e_{i,1}, \ldots, e_{i,t}$. Clearly, $X = X(e_{1,1}, \ldots, e_{m,t})$. Denote $X_{i,j} := \mathbf{E}[X|e_{1,1}, \ldots, e_{i,j}]$. For distinct $k, k' \in [m]$, it is easy to see that we have the 'Lipschitz property':

$$|\mathbf{E}[X|e_{1,1}, \ldots, e_{i,j-1}, e_{i,j} = k] - \mathbf{E}[X|e_{1,1}, \ldots, e_{i,j-1}, e_{i,j} = k']| \leq |x_i y_k| + |x_i y_{k'}|.$$

Therefore, we have a bounded difference property on $|X_{i,j} - X_{i,j-1}|$ as follows:

$$\begin{aligned}|X_{i,j} - X_{i,j-1}| = \Bigg|&\mathbf{E}[X|e_{1,1}, \ldots, e_{i,j-1}, e_{i,j}] \\ &- \frac{1}{m-j+1}\sum_{k'\in[m]\setminus\{e_{i,1},\ldots,e_{i,j-1}\}} \mathbf{E}[X|e_{1,1}, \ldots, e_{i,j-1}, e_{i,j} = k']\Bigg| \\ &\leq |x_{e_j}||y_i| + \frac{1}{m-j+1}\sum_{k'\in[m]\setminus\{e_{i,1},\ldots,e_{i,j-1}\}} \mathbb{1}[(k', i) \in \overline{L}]|x_{k'} y_i|\end{aligned}$$

We will use that the above quantity is bounded by $\frac{2\sqrt{t}}{m}$ since we only consider $|x_{e_j} y_i|$ where $(e_j, i) \in \overline{L}$. However, another way to upper bound the above is by using

$$\begin{aligned}
&\frac{1}{m-j+1}\sum_{k'\in[m]\setminus\{e_{i,1},\ldots,e_{i,j-1}\}} \mathbb{1}[(k',i)\in\overline{L}]|x_{k'}y_i| \\
&\leq \frac{1}{m-j+1}\sum_{k'\in[m]\setminus\{e_{i,1},\ldots,e_{i,j-1}\}} |x_{k'}y_i| \\
&\leq \frac{|y_i|}{n-j+1}\sum_{k'\in[m]} |x_{k'}| \\
&\leq \frac{2|y_i|}{\sqrt{m}}.
\end{aligned}$$

Using this, we now compute the variance of the martingale:

$$\begin{aligned}
\mathrm{Var}(X_{i,j}-X_{i,j-1}|e_{1,1},\ldots,e_{i,j-1}) &\leq \frac{1}{m-j+1}\sum_{k\in[m]}\left(|x_ky_i|+\frac{2|y_i|}{\sqrt{m}}\right)^2 \\
&\leq \frac{2}{m-j+1}\sum_{k\in[m]}\left(|x_ky_i|^2+\frac{4y_i^2}{m}\right) \\
&\leq \frac{10y_i^2}{m-j+1}.
\end{aligned}$$

Where the last inequality uses that $\sum_k x_k^2 \leq 1$. Therefore, the variance of the martingale is at most$t\cdot\frac{10}{m-t}\sum_i y_i^2 \leq \frac{20t}{m} =: \sigma^2$. This is because $\sum_i y_i^2 \leq 1$. Therefore, by the bounded variance martingale inequality (4.3), using $|X_i - X_{i-1}| \leq \frac{2\sqrt{t}}{m} =: C$:

$$\mathbb{P}(X \geq (D+1)\sqrt{t}) \leq \exp\left\{-\frac{D^2t}{2\sigma^2+tC/3}\right\} \leq \exp\left\{-\frac{D^2t}{\frac{40t}{m}+\frac{2t}{3m}}\right\} \leq \exp\left\{-\Omega(D^2m)\right\}.$$

For a large enough constant D, this lets us union bound over all $x, y \in T$, whose number can be bounded by $\left(\frac{C_v}{\epsilon}\right)^m$.

□

4.4 Discussion and open problems

We have given an upper bound on t-regular hypergraph discrepancy in terms of t and a spectral property of the incidence matrix. However, when one restricts attention to random t-regular hypergraphs, the $O(\sqrt{t})$ bound is achieved only when $m = \Omega(n)$. In the case where $m = o(n)$, one can replace λ in Theorem 4.1.1 by λ' where

$$\lambda'(\mathcal{H}) := \max_{\substack{U\subset V\\|U|=16m}} \max_{\substack{v\perp\overline{1},\\\|v\|=1,\\\mathrm{supp}(v)\subseteq U}} \|Mv\|$$

and the proof would remain the same. This is because using the partial coloring theorem (Theorem 4.2.1), one may assign colors to all but at most $16m$ vertices while maintaining that the

discrepancy of *every* edge is 0. However, when $\mathcal{H}$ is a random t regular hypergraph with n vertices and $m = o(n)$ edges, we need not have $\lambda'(\mathcal{H}) = O\left(\sqrt{t}\right)$ (in fact, the guess would be $O(\sqrt{tn/m})$). The problem is that Claim 4.3.6 (In Section 4.3.2) does not extend. However, in this regime, we believe that with high probability, the discrepancy is much *lower* than $\sqrt{t}$ (in contrast to λ growing).

Recently, Franks and Saks [FS18] showed that for $n = \tilde{\Omega}(m^3)$, the discrepancy is $O(1)$ almost surely. Independently, Hoberg and Rothvoss [HR19] considered a different model of random hypergraphs with n vertices and m edges and each vertex-edge-incidence is an i.i.d. Ber(p) random variable. They show that if $n = \tilde{\Omega}(m^2)$, the discrepancy is $O(1)$ almost surely. Both [FS18] and [HR19] used similar Fourier analytic techniques inspired by [KLP12]. Moreover, it was an open question in [HR19] whether the hypergraph with i.i.d Ber(1/2) incidences where $n = O(m \log m)$ almost surely has discrepancy $O(1)$. This was shown to be true by the author [Pot18].

We argue that this is an interesting regime for random regular hypergraphs, as this kind of discrepancy bound is not implied by the Beck-Fiala conjecture. The case where $n = \Omega(m \log m)$, is of particular interest, since we believe there is a phase transition for constant discrepancy at this point. On the one hand, we do not know if the discrepancy bound given by Corollary 4.1.3 is the truth, and on the other hand, we do not know if random regular hypergraphs with, for example, $n = \Theta(m^{1.5})$ almost surely has discrepancy $O(1)$. We conclude with a conjecture, building on an open problem (open problem 1) in [FS18]:

Conjecture 4.4.1. *There is an absolute constant $K > 0$ such that the following holds. Let $t > 0$ be any integer and $\mathcal{H}$ be a random t-regular hypergraph on n vertices and $K\frac{n}{\log n}$ edges. Then with high probability,*

$$\operatorname{disc}(\mathcal{H}) = O(1).$$

Chapter 5
On the list recoverability of randomly punctured codes

The main result of this section is that one can puncture codes with good distance appropriately to obtain codes that are zero-error list recoverable beyond the Johnson bound. This shows the existence of Reed-Solomon codes that are zero-error list recoverable beyond the Johnson bound. It was previously known that there are Reed-Solomon codes that do not have this property, thus showing that the choice of evaluation points in a Reed-Solomon codes does make a difference. As an immediate corollary, we obtain better degree bounds on unbalanced expanders that come from Reed-Solomon codes. To state the result formally, we need some definitions

5.1 Introduction

List recoverable codes were defined by Guruswami and Rudra [GR06] to demonstrate a barrier to improving known algorithms for list decoding. Here, we study list recoverable codes in their own right, showing that random puncturings of codes over a sufficiently large alphabet are list recoverable. Our result is analogous to earlier work by Rudra and Wooters [RW14, RW15] on the list decodability of randomly punctured codes.

We use q to denote the alphabet size, and n to denote the block length of an arbitrary code. Given two codewords $c_1, c_2 \in [q]^n$, denote the Hamming distance between c_1 and c_2 by $\Delta(c_1, c_2)$. Denote the minimum distance between a codeword $c \in [q]^n$ and a set $\mathcal{L} \subseteq [q]^n$ by $\Delta(c, \mathcal{L})$.

Definition 5.1.1 (List recoverability)**.** *Let q, n, k be positive integers, and let $\delta > 0$ and $0 \leq \rho < 1$ be real numbers. A code $\mathcal{C} \subset [q]^n$ is (ℓ, δ, ρ) list recoverable if, for every collection of sets $\{L_i \subseteq [q]\}_{i \in [n]}$ with $|L_i| \leq \ell$ for each i, we have*

$$|\{c \in \mathcal{C} \mid \Delta(c, L_1 \times \cdots \times L_n) \leq \rho n\}| \leq \ell(1 + \delta)$$

In the above definition, ℓ is called the *list size* from which the code can be recovered. The case $\rho = 0$ is already interesting, and called *zero-error* list recoverability. We say that a code $\mathcal{C}$ is (ℓ, δ) zero-error list recoverable if it is $(\ell, \delta, 0)$ list recoverable. Also, in the definition, we do not require δ

to be less than 1. This was of stating simply allows us to carry the notation over for the discussion in Section 5.1.1.

A *puncturing* of a code $\mathcal{C} \subset [q]^n$ to a set $S \subset [n]$ is the code $\mathcal{C}_S \subset [q]^S$ defined by $\mathcal{C}_S[i] = \mathcal{C}[i]$ for each $i \in S$. A punctured code will typically have higher rate, but lower distance, than the unpunctured version. Our main result is that every code over a large enough alphabet $[q]$ can be punctured to a code of rate $R > q^{-1/2}$ while being list recoverable with list size roughly R^{-2}. On a first reading, it may be helpful to first consider the case $\rho = 0$.

Theorem 5.1.2. *There are positive constants c, n_0, and q_0 so that the following holds. Let $0 < \delta \leq 1$ and $0 \leq \rho < 1-(1+\delta)^{-1/2}$ be real numbers. Denote $\gamma = (1+\delta)(1-\rho)^2 - 1$ and $\sigma = (1-\rho)(2-\rho)^{-1}$. Let $n > n_0$ and $q > q_0$ be integers. Let $q^{-1/2} < \epsilon < \min(c, 2^{-1}\gamma\sigma)$. Then, every code $\mathcal{C} \subset [q]^n$ with distance at least $n(1 - q^{-1} - \epsilon^2)$ can be punctured to rate $\Omega\left(\frac{\epsilon}{\log q}\right)$ so that it is $(\epsilon^{-2}\sigma^2\gamma, \delta, \rho)$-list recoverable.*

To attempt to make the parameters more transparent, we would like to draw the reader's focus to the list size, i.e., $\epsilon^{-2}\sigma^2\gamma$. The main point here is that this is as large as ϵ^{-2}, so one way to interpret the above theorem is that we get $(O_{\delta,\rho}(\epsilon^{-2}), \delta, \rho)$-list recoverability after the aforementioned puncturing. In fact, we show a random puncturing of $\mathcal{C}$ is list recoverable with the same list size with high probability; see Theorem 5.3.1 for a precise statement.

Theorem 5.1.2 is analogous to a theorem of Rudra and Wooters [RW14, RW15] on the *list decodability* of punctured codes over large alphabets. A code $\mathcal{C} \subset [q]^n$ is (ρ, ℓ)-list decodable if for each $x \in [q]^n$, there are at most ℓ codewords of $\mathcal{C}$ that differ from x in fewer than ρn coordinates.

Theorem 5.1.3 ([RW15])**.** *Let $\epsilon > q^{-1/2}$ be a real number, and q, n be sufficiently large integers. Every code $\mathcal{C} \subset [q]^n$ with distance $n(1 - q^{-1} - \epsilon^2)$ can be punctured to rate $\tilde{\Omega}\left(\frac{\epsilon}{\log q}\right)$ so that it is $(1 - O(\epsilon), O(\epsilon^{-1}))$-list decodable.*

Theorems 5.1.2 and 5.1.3 are most interesting in the case of Reed-Solomon codes. The codewords of the degree-d Reed-Solomon code over $\mathbb{F}_q$ with evaluation set $S \in \binom{[q]}{m}$ are the evaluations of all univariate polynomials of degree at most d on elements of S. In other words, suppose $S = \{s_1, \ldots, s_m\}$, the degree-d Reed-Solomon code on S is the set

$$\{(p(s_1), \ldots, p(s_m)) \mid \deg(p) \leq d\}.$$

The block length of this code is $m \leq q$. Since two distinct polynomials of degree at most d can agree on at most d locations, the distance of any degree-d Reed-Solomon code is at least $m - d$.

A fundamental result, which gives a lower bound on the list decodability of a code with given distance, is the *Johnson bound* (see, for example Corollary 3.2 in [Gur06]).

Theorem 5.1.4 (Johnson bound for list decoding)**.** *Every code $\mathcal{C} \subset [q]^n$ of minimum distance at least $n(1-(1/q)-\epsilon^2)$ is $(n(1-q^{-1}-\epsilon), O(\epsilon^{-1}))$- list decodable.*

One of the main points of Theorem 5.1.3 is that it shows that there are Reed-Solomon codes that are list decodable beyond the Johnson bound.

A similar result as Theorem 5.1.4, using a similar argument, also known as the Johnson bound, is known for list recoverability (see for example, Lemma 5.2 in [GKdO+18]).

Theorem 5.1.5 (Johnson bound for list recovery)**.** *Let $\mathcal{C} \subseteq [q]^n$ be a code of relative distance r. Then $\mathcal{C}$ is (ℓ, δ, ρ)-list recoverable for any $\rho \leq 1 - \sqrt{\ell(1-r)}$ where $\delta = \frac{r}{(1-\rho)^2 - \ell(1-r)} - 1$.*

A result of Guruswami and Rudra [GR06]) shows that there are Reed-Solomon codes that are not list recoverable beyond the Johnson bound.

Theorem 5.1.6. *Let $q = p^m$ where p is a prime, and let $\mathcal{C}$ denote the degree-$\left(\frac{p^m-1}{p-1}\right)$ Reed-Solomon code over $\mathbb{F}_q$ with $\mathbb{F}_q$ as the evauation set. Then there are lists $S_1, \ldots, S_q$ each of size p such that*

$$|\mathcal{C} \cap (S_1 \times \cdots \times S_q)| = q^{2^m}$$

To understand this, recall that a degree-d Reed-Solomon code has relative distance $1 - \frac{1}{q} - \frac{d}{q}$. Setting $\ell = p - 1$ and $\rho = 0$ in the Johnson bound tells us that such a code is $(p-1, O(q), 0)$-list recoverable. Setting the list size as p in the bound gives us nothing, and Theorem 5.1.6 says that the number of codewords grows superpolynomially in q. On the other hand, Theorem 5.1.2 immediately gives the following corollary.

Corollary 5.1.7. *For a prime power q and $\epsilon \geq q^{-1/2}$, there are Reed-Solomon codes of rate $\tilde{\Omega}\left(\frac{\epsilon}{\log q}\right)$ which are $(q/2, 1/2)$-list recoverable.*

Again, one can easily check that setting $k = q/2$ in the Johnson bound gives nothing.

5.1.1 Unbalanced expander graphs from codes

The zero-error case of Theorem 5.1.2 leads to some progress on a question of Guruswami regarding unbalanced expanders obtained from Reed-Solomon graphs. This was also the main motivation behind this theorem.

Informally, an expander graph is a graph where every small set of vertices has a relatively large neighborhood. In this case, we say that all small sets *expand*. One interesting type of expander graphs are *unbalanced expanders*. These are bipartite graphs where one side is much larger than the other side, and we want that all the small subsets of the *larger* side expand.

Definition 5.1.8 (Unbalanced expander)**.** *A (k, d, ϵ)-regular unbalanced expander is a bipartite graph on vertex set $L \sqcup R$, $|L| \geq |R|$ where the degree of every vertex in L is d, and for every $S \subseteq L$ such that $|S| = k$, we have that $|N(S)| \geq d|S|(1-\epsilon)$.*

Note that in the above definition, $|N(S)| \leq d|S|$. We are typically interested in infinite families of unbalanced expanders for which $\epsilon = o(1)$, $d = o(|R|)$, and $k = \tilde{\Omega}(|R|/d)$.

For a q-ary error correcting code $\mathcal{C} \subset [q]^n$, and a subset $S := \{i_1, \ldots, i_{|S|}\} \subseteq [n]$ with $i_1 < \cdots < i_{|S|}$, we use $\mathcal{C}_S$ to denote the S-punctured code given by

$$\mathcal{C}_S := \{(c_{i_1}, \ldots, c_{i_{|S|}}) \mid (c_1, \ldots, c_n) \in \mathcal{C}\}.$$

Thus, $\mathcal{C}_S$ is just the set of codewords of $\mathcal{C}$ restricted to the coordinates in S.

Given a code $\mathcal{C} \subseteq [q]^n$, it is natural to look at the bipartite graph, which we will denote by $G(\mathcal{C})$ where the vertex sets are $|C| \sqcup ([n] \times [q])$. For every $c = (c_1, \ldots, c_n) \in \mathcal{C}$ the set of neighbors is $\{(1, c_1), \ldots, (n, c_n)\}$. This graph is especially interesting when $\mathcal{C}$ is a low-degree Reed-Solomon code evaluated at an appropriate set.

The following is a open question in the study of pseudorandomness that is attributed to Guruswami [Gur], (also explicitly stated in [CZ18]): Fix an integer d. For a subset $S \in \binom{[q]}{m}$, define $\mathcal{C}_S$ to be the degree-d Reed-Solomon code with S as the evaluation set, where d is a constant.

Question: *What is the smallest m such that when S is chosen uniformly at random, $G(\mathcal{C}_S)$ is, with high probability, a $(o(q), o(1))$-unbalanced expander?*

There are examples of explicit constructions unbalanced expanders that come from other means (in fact, other codes) [GUV09]. However, the above "natural" geometric/combinatorial question is still interesting in its own right and so far, seems to evade known techniques.

It was probably well known that $m = \Omega(\log q)$, and we also give a proof of this (Theorem 5.4.1) since we could not find it in the literature. But for upper bounds, it seems nothing better than the almost trivial $m = O(q)$ was known [Che]. Since the zero-error list recoverability of $\mathcal{C}$ is equivalent to the expansion of $G(\mathcal{C})$, an immediate Corollary to Theorem 5.3.1 gives an improved upper bound.

Corollary 5.1.9. *Let q, n be sufficiently large integers and $\delta \in (0,1)$, $\epsilon > q^{-1/2}$ be real numbers. For every code $\mathcal{C} \subset [q]^n$ with relative distance $1 - q^{-1} - \epsilon^2$, there is a subset $S \subset [n]$ such that $|S| = O(\epsilon n \log q)$ such that $G(\mathcal{C}_S)$ is a $(\delta\epsilon^{-2}, |S|, \delta)$-unbalanced expander.*

Instantiating the above theorem for degree-d Reed-Solomon codes, we have $n = q$ and $\epsilon = (d/q)^{-\frac{1}{2}}$. This gives, $m = \tilde{O}(\sqrt{q})$.

5.2 Algebraic view of expanders

In this section, we give some perspective on why obtaining unbalanced expanders are in some sense, harder than just regular expanders. The point is essentially that regular expansion is basically determined by one quantity, i.e., the second largest eigenvalue of the adjacency matrix. This quantity is relatively easy to get a handle on for proving crude bounds (good enough for *some* expansion). Unbalanced expansion is a weaker condition (in the sense that regular expansion implies that small sets also expand) and is, so far, not determined by a single relatively tractable quantity, and in particular, not by the second eigenvalue of the adjacency matrix, which forms most of the base for our understanding of expander graphs.

We recall the algebraic definition of expander graphs, that is usually much easier to deal with. Let G be a bipartite graph on the vertex set $L \sqcup R$ where every vertex in L has degree d_L and every vertex in R had degree d_R. Let us use $M = M_G$ to denote it's adjacency matrix. Since M is a real symmetric matrix, it has real eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. The largest eigenvalue of this matrix is $\lambda_1 = \sqrt{d_L d_R}$. We use $\lambda = \lambda_G = \lambda_2$ to denote the second largest eigenvalue. Expansion can also be defined in terms of λ.

Definition 5.2.1. *((d, λ)-expander) A bipartite graph G on $L \sqcup R$ is a (d_L, d_R, λ)-expander if*

1. *Every vertex in L has degree d_L and every vertex in R has degree d_R.*
2. *$\lambda_G \leq \lambda$.*

This is a working definition that is far more due to our relatively better understanding of eigenvalues of matrices. The fact that a graph G is a (d_L, d_R, λ)-expander also means that all small sets expand. The following well known result which is one direction in the so called 'Expander Mixing Lemma' originally due to Haemers [Hae95]. This is similar to Lemma **??** and the proof is almost identical.

Lemma 5.2.2 (Expander Mixing Lemma)**.** *Let G be a (d_L, d_R, λ)-expander graph on $L \sqcup R$. Then for any subsets $S \subset L$ and $T \subset R$, such that $|S| = \alpha|L|$ and $|T| = \beta|R|$, we have*

$$\frac{e(S,T)}{e(L,R)} \leq \alpha\beta + \frac{\lambda}{\sqrt{d_L d_R}}\sqrt{\alpha\beta(1-\alpha)(1-\beta)}.$$

One example to help give more perespective here is: Let S be of size at most $o\left(\frac{|R|}{d_L}\right)$, and $T = N(S)$. It is easy to see that $|T| \leq d_L|S| = o(|R|)$, or, equivalently, $\beta \leq \alpha d_R = o(1)$. A straightforward application of Lemma 5.2.2 gives us

$$\alpha = \frac{e(S,T)}{e(L,R)} \leq \alpha\beta + \frac{\lambda}{\sqrt{d_L d_R}}\sqrt{\alpha\beta}$$

or, using that $\beta = o(1)$,

$$\beta \geq \alpha \frac{d_L d_R}{\lambda^2}(1 - o(1)).$$

One thing to observe here is that the smaller λ gets, the larger $|T| = |N(S)|$ is, and so if $\lambda = O(\sqrt{d_L})$, we would have that $|T| = \Omega(d_L|S|)$. This gives us an approach to show that certain graphs are unbalanced expanders, since as mentioned before, eigenvalues, especially of structured matrices is a relatively well studied subject with a plethora of extremely powerful tools.

The reason this approach is not viable is essentially that when λ cannot be so small. It is atleast of the order of $\Omega(\sqrt{d_R})$, which can be quite large when $|L| \gg |R|$.

Claim 5.2.3. *For a (d_L, d_R)-regular bipartite graph G on $L \sqcup R$ where $\max\{d_L, d_R\} = o(\min\{|L|, |R|\})$, we have that $\lambda(G) \geq \Omega(\sqrt{d_L} + \sqrt{d_R})$.*

Proof. Let M be the adjacency matrix of G. The largest eigenvalue of M in magnitude is $\sqrt{d_L d_R}$ and is given by the eigenvectors $v_1 = \frac{1}{2\sqrt{L}}\mathbb{1}_L + \frac{1}{2\sqrt{R}}\mathbb{1}_R$ and $v_2 = \frac{1}{2\sqrt{L}}\mathbb{1}_L - \frac{1}{2\sqrt{R}}\mathbb{1}_R$. Let $A \subset R$ be maximal such that $N(A) \leq |L|/2$ and define $A' := N(A)$. Likewise, let $B \subset L$ be maximal such that $|N(B)| \leq |R|/2$ and let $B' := N(B)$. Clearly, we have $|A| \geq \frac{|L|}{3d_R}$ and $|B| \geq \frac{|R|}{2d_L}$. Consider the vector

$$\begin{aligned} u = &\sqrt{\frac{|L| - |A'|}{|A'| \cdot |L|}}\mathbb{1}_{A'} - \sqrt{\frac{|A'|}{(|L| - |A'|) \cdot |L|}}\mathbb{1}_{L \setminus A'} \\ &+ \sqrt{\frac{|R| - |B'|}{|B'| \cdot |L|}}\mathbb{1}_{B'} - \sqrt{\frac{|B'|}{(|L| - |B'|) \cdot |L|}}\mathbb{1}_{L \setminus B'}. \end{aligned}$$

Note that $u \perp v_1$ and $u \perp v_2$, and so since the eigenvectors of real symmetric matrices are orthogonal, we have that $\lambda \geq \|Mv\|$. Using the fact that $M\mathbb{1}_{A'} = d_R\mathbb{1}_A$ and $M\mathbb{1}_{B'} = d_L\mathbb{1}_B$, we have

$$\begin{aligned} \lambda \geq \|Mu\| &\geq \sqrt{\frac{|L| - |A'|}{|A'| \cdot |L|}} \cdot \|M\mathbb{1}_{A'}\| + \sqrt{\frac{|R| - |B'|}{|B'| \cdot |L|}} \cdot \|M\mathbb{1}_{B'}\| \\ &= \sqrt{\frac{|L| - |A'|}{|A'| \cdot |L|}} d_R\sqrt{|A|} + \sqrt{\frac{|R| - |B'|}{|B'| \cdot |B|}} d_L\sqrt{|B|} \\ &\geq d_R\sqrt{\frac{|A|}{|L|}} + d_L\sqrt{\frac{|B|}{|R|}} \\ &\geq \sqrt{d_R/3} + \sqrt{d_L/3}. \end{aligned}$$

□

5.3 Proof of Theorem 5.1.2

The bulk of this section is the statement and proof of Theorem 5.3.1. After the proof of Theorem 5.3.1, we show how to derive Theorem 5.1.2 from it.

5.3.1 A probability inequality

We will use the following large deviation inequality for hypergeometric random variables (see [DP09]). Let X be a hypergeometric random variable with mean μ. Then for any $\delta \geq 1$,

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp(-\delta\mu/4). \tag{5.1}$$

5.3.2 A sketch of the proof

Here, we sketch the proof when $\rho = 0$, i.e., for *zero-error* list recovery. This contains most of the main ideas required for the general theorem. Let $S = \{x_1, \ldots, x_m\} \subset [n]$ be a randomly chosen evaluation set. The main observation is that if there are input lists $L_1, \ldots, L_m \subseteq [q]$, such that $(L_1 \times \cdots \times L_m)$ contains a large subset $\mathcal{D} \subseteq \mathcal{C}$ of codewords, then there is a small subset $\mathcal{C}' \subseteq \mathcal{D}$ $(\subseteq \mathcal{C})$ which agree on an unusually high number of coordinates. An appropriately sized random subset of $\mathcal{D}$ does this. Thus the event that a given puncturing is bad is contained witnessed by the event that there are few codewords that agree a lot on the coordinates chosen in S. The number of events of the latter kind are far fewer in number, thus giving us a relatively small(er) number of bad events to overcome for the union bound.

5.3.3 Proof of Theorem 5.1.2

We not prove Theorem 5.1.2. The calculations in the proof of Theorem 5.3.1 are all explicit, but we have not tried to optimize the constant terms.

Theorem 5.3.1. *Let $0 < \delta < 1$ and $0 \leq \rho < 1 - (1+\delta)^{-1/2}$ be real numbers. Let q, n, d, ℓ, and m be positive integers. Let $\mathcal{C} \subset [q]^n$ be a code of distance at least $n - nq^{-1} - d$. Denote $\gamma = (1+\delta)(1-\rho)^2 - 1$ and $\sigma = (1-\rho)(2-\rho)^{-1}$. Suppose that the following inequalities are satisfied:*

$$\begin{aligned} d &\geq nq^{-1}, \\ 4\gamma^{-1} \leq \ell &\leq 800^{-1}\,\sigma\gamma n d^{-1}, \\ \sigma m &\geq 1280\sqrt{\ell\gamma^{-1}}\log|\mathcal{C}|, \\ m &< n. \end{aligned}$$

Then, for $S \in \binom{[n]}{m}$ chosen uniformly at random, the probability that $\mathcal{C}_S$ is (ℓ, δ, ρ)-list recoverable is at least $1 - e^{-\sigma m/64}$.

Proof. For any $\mathcal{C}' \subseteq \mathcal{C}$, denote by $T(\mathcal{C}')$ the set of coordinates $i \in [n]$ such that there is a pair $c_1, c_2 \in \mathcal{C}'$ with $c_1[i] = c_2[i]$.

The basic outline of the proof is to first show that, for any S such that $\mathcal{C}_S$ is not (ℓ, δ, ρ)-list recoverable, there is a pair $S', \mathcal{C}'$ such that S' is large and $|T(\mathcal{C}') \cap S'|$ is unusually large. Taking a union bound over all candidates for $\mathcal{C}'$ then shows that there cannot be too many pairs of this sort.

Let $S \in \binom{[n]}{m}$ so that $\mathcal{C}_S$ is not (ℓ, δ, ρ)-list recoverable. We will show that there is a set $\mathcal{C}' \subset \mathcal{C}_S$ such that

$$|\mathcal{C}'| \leq 10\sqrt{\ell/\gamma}, \text{ and} \tag{5.2}$$

$$|T(\mathcal{C}') \cap S| \geq \sigma m/4. \tag{5.3}$$

Since $\mathcal{C}_S$ is not (ℓ, δ, ρ)-list recoverable, there are subsets $L_i \subseteq [q]$ for each $i \in S$ such that each $|L_i| \leq \ell$ and $|\{c \in \mathcal{C}_S : \Delta(c, \prod_{i \in S} L_i) \leq \rho n\}| > k(1 + \delta)$.

Let

$$\mathcal{D} = \{c \in \mathcal{C}_s : \Delta(c, \prod_{i \in S} L_i) \leq \rho n\}.$$

For $i \in S$, let

$$\mathcal{D}_i = \{c \in \mathcal{D} : c[i] \in L_i\}.$$

Let

$$I = \{(c, i) \in \mathcal{D} \times S : c \in \mathcal{D}_i\}.$$

From the definition of $\mathcal{D}$, we have

$$|I| \geq |\mathcal{D}|(1 - \rho)m. \tag{5.4}$$

Note that the average cardinality of the $\mathcal{D}_i$ is $(1 - \rho)|\mathcal{D}|$. Let

$$S' = \{i \in S : |\mathcal{D}_i| \geq (1 - \rho)^2|\mathcal{D}|\}.$$

If $\rho = 0$, then $\mathcal{D}_i = \mathcal{D}$ for each i, and hence $|S'| = m$. Next we show that, if $\rho > 0$, then $|S'| \geq (1 - \rho)(2 - \rho)^{-1}m = \sigma m$. Since $|\mathcal{D}_i \leq |\mathcal{D}|$ for each i, we have

$$|S'|\,|\mathcal{D}| \geq \sum_{i \in S'} |D_i| = |I| - \sum_{i \in S \setminus S'} \mathcal{D}_i. \tag{5.5}$$

Since $|D_i| < (1 - \rho)^2|\mathcal{D}|$ for each $i \in S \setminus S'$, we have

$$\sum_{i \in S \setminus S'} \leq (m - |S'|)(1 - \rho)^2|\mathcal{D}|. \tag{5.6}$$

A straightforward rearrangement of (5.4), (5.5), and (5.6) using the assumption that $\rho > 0$ leads to the claimed lower bound on $|S'|$:

$$|S'| \geq \sigma m. \tag{5.7}$$

Since $\sigma < 1$, the bound $|S'| \geq \sigma m$ holds for the case $\rho = 0$ as well.

For each $i \in S'$, choose a set $P_i \subset \binom{\mathcal{D}}{2}$ of $|P_i| \geq \gamma k/2$ disjoint pairs of codewords in $\mathcal{D}_i$ such that for each $\{c_1, c_2\} \in \mathcal{P}_i$, we have $c_1[i] = c_2[i]$. This is always possible since $|L_i| \leq \ell$ and $|\mathcal{D}_i| \geq (1+\rho)^2|\mathcal{D}| \geq (1+\gamma)\ell$.

Now choose $\mathcal{C}'$ randomly by including each element of $\mathcal{D}$ with probability $p = (\gamma\ell/2)^{-1/2}\ell(1+\delta)|\mathcal{D}|^{-1}$. Since $\ell \geq 4\gamma^{-1}$ by hypothesis and $|\mathcal{D}| \geq \ell(1+\delta)$ by the assumption that $\mathcal{C}_S$ is not (ℓ, δ, ρ)-list recoverable, we have $p < 1$. The expected size of $\mathcal{C}'$ is

$$\mathbf{E}[|\mathcal{C}'|] = p|\mathcal{D}| \leq (\gamma/(2\ell))^{-1/2}(1+\delta) \leq (8\ell/\gamma)^{1/2}.$$

For any fixed pair $c_1 \neq c_2$ of codewords in $\mathcal{D}$, the probability that both are included in $\mathcal{C}'$ is p^2. Since the pairs in P_i are disjoint, the events that two distinct pairs $\{c_1, c_2\}, \{c_3, c_4\} \in P_i$ are both included in $\mathcal{C}'$ are independent. Hence, the probability that no pair in P_i is included in $\mathcal{C}'$ is $(1-p^2)^{|P_i|} < e^{-p^2|P_i|} < 1/2$. Consequently, for each fixed $i \in S'$, the probability that $i \in T(\mathcal{C}')$ is greater than $1/2$. By linearity of expectation, $\mathbf{E}[|T(\mathcal{C}') \cap S'|] \geq |S'|/2 \geq \sigma m/2$.

Let

$$Y = |T(\mathcal{C}') \cap S'| - \frac{\sigma m}{4}\frac{|\mathcal{C}'|}{\mathbf{E}[|\mathcal{C}'|]}.$$

By linearity of expectation, $\mathbf{E}[Y] \geq \sigma m/4$, hence there is some specific choice of $\mathcal{C}'$ for which $Y \geq \sigma m/4$. This can hold only if $|T(\mathcal{C}') \cap S| \geq |T(\mathcal{C}') \cap S'| \geq m/4$ and $|\mathcal{C}'| \leq 3\mathbf{E}(|\mathcal{C}'|)$ simultaneously, which establishes (5.2) and (5.3).

Next we bound the probability that, for a fixed choice of $\mathcal{C}'$ and random S, we have have $|T(C') \cap S|$ large. Let $\mathcal{C}' \subset \mathcal{C}$ be an arbitrary set of $|\mathcal{C}'| \leq 10\ell^{1/2}\gamma^{-1/2}$ codewords. Since the distance of $\mathcal{C}'$ is at least $n - nq^{-1} - d$ and $d \geq nq^{-1}$, we have

$$|T(\mathcal{C}')| \leq (nq^{-1} + d)\binom{|\mathcal{C}'|}{2} < d|\mathcal{C}'|^2. \tag{5.8}$$

For $S \in \binom{[n]}{m}$ chosen uniformly at random, $|T(\mathcal{C}') \cap S|$ follows a hypergeometric distribution. Specifically, we are making m draws from a population size of n of which $|T(\mathcal{C}')| \leq d|\mathcal{C}'|^2$ contribute to $|T(\mathcal{C}') \cap S|$. Using the assumption that $\ell \leq \gamma\sigma n(800d)^{-1}$, the expected value of $|T(\mathcal{C}') \cap S|$ is

$$\mathbf{E}\left[|T(\mathcal{C}') \cap S|\right] \leq d|\mathcal{C}'|^2 n^{-1} m \leq 100\frac{d\ell}{\gamma n}m \leq \frac{\sigma m}{8}. \tag{5.9}$$

Combining this with standard tail bounds for the hypergeometric distribution (5.1),

$$\mathbb{P}(|T(\mathcal{C}') \cap S| \geq \sigma m/4) \leq \exp\left(-\frac{\sigma m}{32}\right). \tag{5.10}$$

Finally, we take a union over all candidates for $\mathcal{C}'$. Let X be the event that $\mathcal{C}_S$ is not (ℓ, δ, ρ) list recoverable, with $S \in \binom{[n]}{m}$ uniformly at random. Using the assumption that $\sigma m \geq 1280\sqrt{\ell/\gamma}\log|\mathcal{C}|$, we have

$$\begin{aligned}
\mathbb{P}(X) &\leq \sum_{\mathcal{C}' \subset \mathcal{C}_S : |\mathcal{C}'| \leq 10\sqrt{\ell/\gamma}} \mathbb{P}(|T(\mathcal{C}' \cap S)| \geq \sigma m/4) \\
&\leq \binom{|\mathcal{C}|}{\lceil 10\sqrt{\ell/\gamma}\rceil + 1} \exp\left(-\frac{m}{32}\right) \\
&< \exp\left(20\sqrt{\ell/\gamma}\log|\mathcal{C}| - \sigma m/32\right) \\
&\leq \exp(-\sigma m/64),
\end{aligned}$$

as claimed. □

We now show how to derive Theorem 5.1.2 from Theorem 5.3.1.

Proof of Theorem 5.1.2. Suppose we have δ, ρ, n, q, and ϵ as in the hypotheses of Theorem 5.1.2. Let $m = \lceil 1280\epsilon^{-1}\log|C|\rceil$. The singleton bound combined with the assumption that $\epsilon < c$ for a suitably chosen absolute constant c implies that $m < n$. Choose $S \in \binom{[n]}{m}$ uniformly at random. The rate of $\mathcal{C}_S$ is

$$R = \log|\mathcal{C}|(m\log q)^{-1} = \Omega(\epsilon(\log q)^{-1}).$$

It is straightforward to check that the hypotheses of Theorem 5.3.1 are satisfied if we take $\ell = \epsilon^{-2}\sigma^2\gamma$, and hence we have that $\mathcal{C}_S$ is $(\epsilon^{-2}\sigma^2\gamma, \delta, \rho)$-list recoverable with high probability. □

5.4 Upper bound

Here we show the aforementioned upper bound for the rate to which a degree-d Reed-Solomon code over $\mathcal{F}_q$ can be randomly punctured to be $(q/2, 1/2)$-zero-error list-recoverable.

First, we recall a bit of standard and relevant sumset notation. For a group G and subsets $A, B \subseteq G$, we denote the sumset $A+B = \{a+b \mid a \in A,\ b \in B\}$. Clearly, we have $|A+B| \leq |A|\cdot|B|$. If $G = \mathbb{Z}_p$, then for $n < p/2$, we have that $[n] + [n] = \{2, \ldots, 2n\}$. We are now ready to state and prove the upper bound.

Theorem 5.4.1. *Let $m = o(\log q)$, and $X = \{x_0, \ldots, x_m\}$ be a uniformly random subset of $\mathbb{F}_q$ where q is a prime. Then every $d \geq 1$, the degree-d Reed-Solomon code with the evaluation set at X is, with high probability, not $(q/2, 1/2)$-zero-error list-recoverable.*

Proof. Let $X = \{x_0, \ldots, x_m\}$. Let n be a large number such that $n^m = o(\sqrt{q})$. We are using the fact that $m = o(\log q)$ for the existence of such an n. W.L.O.G assume $x_0 = 0$ and $x_1 = 1$ (if $0, 1 \notin S$,

then adding them to S only makes the lower bound stronger). Consider the two sets

$$X_0 = \frac{1}{1-x_2}[n] + \cdots \frac{1}{1-x_{m-1}}[n]$$

and

$$X_1 = \frac{1}{x_2}[n] + \cdots \frac{1}{x_{m-1}}[n].$$

Claim 5.4.2. *With high probability over the choice of X, we have that $|X_0|, |X_1| = \Omega(\left(n^{m-2}\right))$.*

Proof. We do the proof for X_0, the case for X_1 follows analogously. Let P be the set of "collisions" in X_0. Formally:

$$P := \left\{ (a_2, \ldots, a_{m-2}, b_2, \ldots, b_{m-2}) \mid \sum_{i=2}^{m-2} a_i x_i = \sum_{i=2}^{m-2} b_i x_i \right\}.$$

So the number of distinct elements in X_0 is at least $n^{m-2} - |P|$. We observe that

$$\begin{aligned}
\mathbf{E}[|P|] &= \sum_{\substack{a_2,\ldots,a_{m-2}\in[n] \\ b_2,\ldots,b_{m-2}\in[n]}} \mathbb{P}\left(\sum_{i=2}^{m-2} a_i x_i = \sum_{i=2}^{m-2} b_i x_i \right) \\
&\leq \frac{1}{p} n^{2m-4} \\
&= o(n^{m-2}).
\end{aligned}$$

So by Markov's Inequality, with high probability, $|X_0| \sim n^{m-2}$. □

Consider $\mathcal{D}$, the set of degree-1 Reed-Solomon codes given by the lines

$$\{Y = aX + b\}_{b\in X_0, a\in X_1}.$$

First, we note that $|Y| = \Omega(n^{2m-4})$. Geometrically, $\mathcal{D}$ is just the set of all lines passing through some point of $\{0\} \times X_0$ and $\{1\} \times X_1$. Clearly, $\{c[0] \mid c \in \mathcal{C}\} = X_0$ and $\{c[1] \mid c \in \mathcal{D}\} = X_1$. For $i \neq 0, 1$, let us similarly define $X_i := \{c[x_i] \mid c \in \mathcal{D}\}$.We have that

$$\begin{aligned}
X_i &= \{a(1-x_i) + bx_i\}_{b\in X_0, a\in X_1} \\
&= (1-x_i)\left(\frac{1}{1-x_2}[n] + \cdots \frac{1}{1-x_{m-1}}[n] \right) + x_i \left(\frac{1}{x_2}[n] + \cdots \frac{1}{x_{m-1}}[n] \right) \\
&= \left([n] + \sum_{2\leq j\leq m,\ j\neq i} \frac{1-x_i}{1-x_j}[n] \right) + \left([n] + \sum_{2\leq j\leq m,\ j\neq i} \frac{x_i}{x_j}[n] \right) \\
&= \{2, \ldots, 2n\} + \sum_{2\leq j\leq m,\ j\neq i} \frac{1-x_i}{1-x_j}[n] + \sum_{2\leq j\leq m,\ j\neq i} \frac{x_i}{x_j}[n].
\end{aligned}$$

Thus, $|X_i| \leq (2n) \times n^{2m-6} \leq 2n^{2m-5}$.

This shows that there are lists $X_0, X_1, \ldots, X_m$ each of size at most $\ell := 2n^{2m-5}$ such that there are at least $\Omega(n^{2m-4}) = \ell^{1+\frac{1}{k}}$ codewords, namely $\mathcal{D}$, contained in $X_0 \times \cdots \times X_m$. □

For a fixed d, the above theorem rules out hope of randomly puncturing degree-d Reed-Solomon codes to rate $\omega\left(\frac{1}{\log q}\right)$ for the desired list recoverability. We believe that this is essentially the barrier. We state the concrete conjecture that we alluded to in Section 5.1.1.

Conjecture 5.4.3. *For any $\delta > 0$, the degree-d Reed-Solomon code with evaluation set $\mathbb{F}_q$ can be randomly punctured to rate $\Omega_d\left(\frac{1}{\log q}\right)$ so that is it $(\delta q, \delta)$-list recoverable with high probability.*

5.5 Discussion and open problems

The main open problem that we would like to showcase is Conjecture 5.4.3. This was probably believed to be true but we could not find it written down explicitly in the literature. List recoverable codes have connections to various other combinatorial objects (see [Vad07]) and if true, Conjecture 5.4.3 could lead to the construction of some other interesting combinatorial objects.

The second open problem is to derandomize Theorem 5.1.2, i.e., to find an *explicit* Reed-Solomon code which is list recoverable beyond the Johnson bound at least in the zero-error case. Understanding how these evaluation sets look like could lead to progress on Conjecture 5.4.3, or could be interesting in its own right.

Finally, the last open problem is that given a Reed-Solomon code $\mathcal{C} \subset [q]^m$ of rate R on a randomly chosen evaluation set S, find an efficient algorithm for list recovery, i.e., take input lists $L_1, \ldots, L_m$ of size $O(R^{-2}(\log q)^{-1})$, and output all the codewords contained in $L_1 \times \cdots \times L_m$ with high probability (over the choice of S and the randomness used by the algorithm). This would also likely require some understanding of the properties of the evaluation set.

References

[ABP20] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Improved inapproximability of rainbow coloring. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1479–1495. SIAM, 2020.

[ACC+18] Jai Aslam, Shuli Chen, Ethan Coldren, Florian Frick, and Linus Setiabrata. On the generalized erdskneser conjecture: Proofs and reductions. *Journal of Combinatorial Theory, Series B*, 2018.

[AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$-Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017.

[AKK+05] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Trans. Information Theory*, 51(11):4032–4039, 2005.

[AS00] Noga Alon and Joel H. Spencer. The probabilistic method, 2000.

[AS03] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[ASW15] Emmanuel Abbe, Amir Shpilka, and Avi Wigderson. Reed-muller codes for random erasures and errors. *IEEE Trans. Information Theory*, 61(10):5229–5252, 2015.

[Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 3–10, 2010.

[Bár78] Imre Bárány. A short proof of Kneser's conjecture. *Journal of Combinatorial Theory, Series A*, 25(3):325 – 326, 1978.

[Bec81] József Beck. Roth's estimate of the discrepancy of integer sequences is nearly sharp. *Combinatorica*, 1(4):319–325, 1981.

[Ber67] E. R. Berlekamp. Factoring polynomials over finite fields. *Bell System Tech. J.*, 46:1853–1859, 1967.

[BF81] József Beck and Tibor Fiala. "integer-making" theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

[BFSU98] Andrei Z. Broder, Alan M. Frieze, Stephen Suen, and Eli Upfal. Optimal construction of edge-disjoint paths in random graphs. *SIAM J. Comput.*, 28(2):541–573, 1998.

[BG16] Joshua Brakensiek and Venkatesan Guruswami. New Hardness Results for Graph and Hypergraph Colorings. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 14:1–14:27, 2016.

[BG17] Joshua Brakensiek and Venkatesan Guruswami. The Quest for Strong Inapproximability Results with Perfect Completeness. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 4:1–4:20, 2017.

[BH97] Debe Bednarchak and Martin Helm. A note on the beck-fiala theorem. *Combinatorica*, 17(1):147–149, Mar 1997.

[Bha18] Amey Bhangale. NP-Hardness of Coloring 2-Colorable Hypergraph with Poly-Logarithmically Many Colors. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107, pages 15:1–15:11, 2018.

[BKO19] Jakub Bulín, Andrei A. Krokhin, and Jakub Oprsal. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, (STOC 2019)*, pages 602–613, 2019.

[BKS+10] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497, 2010.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[Blu15] Avrim Blum. Lecture notes in foundations of machine learning and data science, November 2015.

[BM19] Nikhil Bansal and Raghu Meka. On the discrepancy of random low degree set systems. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2557–2564, 2019.

[Buk16] Boris Bukh. An improvement of the beck-fiala theorem. *Combinatorics, Probability and Computing*, 25(3):380?398, 2016.

[Cha00] Bernard Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, New York, NY, USA, 2000.

[Che] Xue Chen. personal communication.

[Chl07] Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 691–701. IEEE, 2007.

[CL06] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Math.*, 3(1):79–127, 2006.

[CS08] Eden Chlamtac and Gyanit Singh. Improved approximation guarantees through higher levels of SDP hierarchies. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 49–62. Springer, 2008.

[CZ18] Xue Chen and David Zuckerman. Existence of simple extractors. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:116, 2018.

[DG13] Irit Dinur and Venkatesan Guruswami. PCPs via Low-Degree Long Code and Hardness for Constrained Hypergraph Coloring. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 340–349, 2013.

[DGKR05] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered pcp and the hardness of hypergraph vertex cover. *SIAM Journal on Computing*, 34(5):1129–1146, 2005.

[DMR09] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.

[DP09] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[DRS02] Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. In *The 34rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[Dum17] Ilya Dumer. Recursive decoding and its performance for low-rate reed-muller codes. *CoRR*, abs/1703.05306, 2017.

[EL15] Esther Ezra and Shachar Lovett. On the beck-fiala conjecture for random set systems. In *APPROX-RANDOM*, 2015.

[FK98] Uriel Feige and Joe Kilian. Zero Knowledge and the Chromatic Number. *J. Comput. Syst. Sci.*, 57(2):187–199, 1998.

[FKS89] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue of random regular graphs. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 587–598, New York, NY, USA, 1989. ACM.

[FS18] C. Franks and M. Saks. On the Discrepancy of Random Matrices with Many Columns. *ArXiv e-prints*, July 2018.

[GHH+17] Venkatesan Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. Super-Polylogarithmic Hypergraph Coloring Hardness via Low-Degree Long Codes. *SIAM J. Comput.*, 46(1):132–159, 2017.

[GKdO+18] Sivakanth Gopi, Swastik Kopparty, Rafael Mendes de Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the gilbert-varshamov bound. *IEEE Trans. Information Theory*, 64(8):5813–5831, 2018.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989.

[GL15] Venkatesan Guruswami and Euiwoong Lee. Strong Inapproximability Results on Balanced Rainbow-Colorable Hypergraphs. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 822–836. SIAM, 2015.

[GR06] Venkatesan Guruswami and Atri Rudra. Limits to list decoding reed-solomon codes. *IEEE Trans. Information Theory*, 52(8):3642–3649, 2006.

[GS17] Venkatesan Guruswami and Rishi Saket. Hardness of Rainbow Coloring Hypergraphs. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017)*, pages 33:1–33:15, 2017.

[GS19] Venkatesan Guruswami and Sai Sandeep. Rainbow coloring hardness via low sensitivity polymorphisms. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019*, pages 15:1–15:17, 2019.

[Gur] Venkatesan Guruswami. personal communication.

[Gur06] Venkatesan Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. *J. ACM*, 56(4):20:1–20:34, 2009.

[Hae95] Willem H. Haemers. Interlacing eigenvalues and graphs. *Linear Algebra and its Applications*, 226-228:593 – 616, 1995. Honoring J.J.Seidel.

[Har70] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):84, 1970.

[HR19] Rebecca Hoberg and Thomas Rothvoss. A fourier-analytic approach for the discrepancy of random set systems. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2547–2556. SIAM, 2019.

[Hua13] Sangxia Huang. Improved Hardness of Approximating Chromatic Number. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 233–243, 2013.

[Hua15] Sangxia Huang. $2^{(\log N)^{1/10-o(1)}}$ Hardness for Hypergraph Coloring. *CoRR*, abs/1504.03923, 2015.

[Kle66] Daniel J. Kleitman. On a combinatorial conjecture of erdós. *Journal of Combinatorial Theory*, 1(2):209 – 214, 1966.

[KLP12] Greg Kuperberg, Shachar Lovett, and Ron Peled. Probabilistic existence of rigid combinatorial structures. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1091–1106, 2012.

[KLS00] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the Hardness of Approximating the Chromatic Number. *Combinatorica*, 20(3):393–415, 2000.

[KMSU15] Shrinivas Kudekar, Marco Mondelli, Eren Sasoglu, and Rüdiger L. Urbanke. Reed-muller codes achieve capacity on the binary erasure channel under MAP decoding. *CoRR*, abs/1505.05831, 2015.

[KNS01] Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *Journal of Algorithms*, 41(1):99–113, 2001.

[KO19] Andrei A. Krokhin and Jakub Oprsal. The complexity of 3-colouring h-colourable graphs. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1227–1239. IEEE Computer Society, 2019.

[KP15] Santhosh Kumar and Henry D. Pfister. Reed-muller codes achieve capacity on erasure channels. *CoRR*, abs/1505.05123, 2015.

[KP18] Swastik Kopparty and Aditya Potukuchi. Syndrome decoding of reed-muller codes and tensor decomposition over finite fields. In *SODA*, pages 680–691, 2018.

[KS14] Subhash Khot and Rishi Saket. Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1607–1625. SIAM, 2014.

[KS17] Subhash Khot and Rishi Saket. Hardness of Coloring 2-Colorable 12-Uniform Hypergraphs with $2^{(\log n)^{\Omega(1)}}$ Colors. *SIAM J. Comput.*, 46(1):235–271, 2017.

[LM15] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.*, 44(5):1573–1582, 2015.

[Lov78] László Lovász. Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, 25(3):319 – 324, 1978.

[LP] Ben Lund and Aditya Potukuchi. On the list recovery of randomly punctured codes. *(in preparation)*.

[LRA93] S. E. Leurgans, R. T. Ross, and R. B. Abel. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.

[LRR17] Avi Levy, Harishchandra Ramadas, and Thomas Rothvoss. Deterministic discrepancy minimization via the multiplicative weight update method. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 380–391, 2017.

[LZ07] Carsten E.M.C. Lange and Gnter M. Ziegler. On generalized kneser hypergraph colorings. *Journal of Combinatorial Theory, Series A*, 114(1):159 – 166, 2007.

[Mat95] Jiří Matoušek. Tight upper bounds for the discrepancy of half-spaces. *Discrete & Computational Geometry*, 13:593–601, 1995.

[Mat99] J. Matousek. *Geometric Discrepancy: An Illustrated Guide*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 1999.

[Mat07] Jiri Matousek. *Using the Borsuk-Ulam Theorem: Lectures on Topological Methods in Combinatorics and Geometry*. Springer Publishing Company, Incorporated, 2007.

[McC87] P. McCullagh. *Tensor methods in statistics*. Monographs on statistics and applied probability. Chapman and Hall, 1987.

[McD93] Colin McDiarmid. A Random Recolouring Method for Graphs and Hypergraphs. *Combinatorics, Probability & Computing*, 2:363–365, 1993.

[Mul54] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers*, EC-3(3):6–12, Sept 1954.

[Pot18] Aditya Potukuchi. Discrepancy in random hypergraph models, 2018.

[Pot19] Aditya Potukuchi. A spectral bound on hypergraph discrepancy. *CoRR*, abs/1907.04117, 2019. *To appear in ICALP 2020*.

[Rao03] KPS Bhaskara Rao. *Theory of generalized inverses over commutative rings*, volume 17. CRC Press, 2003.

[Raz13] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013.

[Ree54] I. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, September 1954.

[Rot17] Thomas Rothvoss. Constructive discrepancy minimization for convex sets. *SIAM J. Comput.*, 46(1):224–234, 2017.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RW14] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 764–773, 2014.

[RW15] Atri Rudra and Mary Wootters. It'll probably work out: Improved list-decoding through random operations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 287–296, 2015.

[Sar90] Karanbir S. Sarkaria. A generalized kneser conjecture. *J. Comb. Theory, Ser. B*, 49(2):236–240, 1990.

[Sho94] Victor Shoup. Fast construction of irreducible polynomials over finite fields. *J. Symb. Comput.*, 17(5):371–391, 1994.

[Spe85] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985.

[Spe88] Joel Spencer. Coloring the projective plane. *Discrete Mathematics*, 73(1):213 – 220, 1988.

[SSV17] Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Efficiently decoding reed-muller codes from random errors. *IEEE Trans. Information Theory*, 63(4):1954–1960, 2017.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Vad07] Salil P. Vadhan. The unified theory of pseudorandomness: guest column. *SIGACT News*, 38(3):39–54, 2007.

[Var16] Girish Varma. Reducing uniformity in Khot-Saket hypergraph coloring hardness reductions. *Chicago J. Theor. Comput. Sci.*, 2016, 2016.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

[Woj96] Jerzy Wojciechowski. Splitting Necklaces and a Generalization of the Borsuk-Ulam Antipodal Theorem. *The Journal of Combinatorial Mathematics and Combinatorial Computing*, 21:235–254, 1996.

[WZ20] Marcin Wrochna and Stanislav Zivny. Improved hardness for H-colourings of G-colourable graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1426–1435. SIAM, 2020.

[Zuc07] David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.