Electronic Colloquium on Computational Complexity, Report No. 68 (2024)

Superpolynomial Lower Bounds for Smooth 3-LCCs and Sharp Bounds for Designs

Pravesh K. Kothari
kothari@cs.princeton.edu
IAS & Princeton University

Peter Manohar
pmanohar@cs.cmu.edu
Carnegie Mellon University

April 9, 2024

Abstract

We give improved lower bounds for binary 3-query locally correctable codes (3-LCCs) $C\colon \{0,1\}^k \to \{0,1\}^n$. Specifically, we prove:

(1) If C is a linear *design* 3-LCC, then $n \geq 2^{(1-o(1))\sqrt{k}}$. A design 3-LCC has the additional property that the correcting sets for every codeword bit form a perfect matching and every pair of codeword bits is queried an equal number of times across all matchings. Our bound is tight up to a factor $\sqrt{8}$ in the exponent of 2, as the best construction of binary 3-LCCs (obtained by taking Reed–Muller codes on $\mathbb{F}_4$ and applying a natural projection map) is a design 3-LCC with $n \leq 2^{\sqrt{8k}}$. Up to a $\sqrt{8}$ factor, this resolves the Hamada conjecture on the maximum $\mathbb{F}_2$-codimension of a 4-design.

(2) If C is a smooth, non-linear 3-LCC with near-perfect completeness, then, $n \geq k^{\Omega(\log k)}$.

(3) If C is a smooth, non-linear 3-LCC with completeness $1-\varepsilon$, then $n \geq \tilde{\Omega}(k^{\frac{1}{2\varepsilon}})$. In particular, when ε is a small constant, this implies a lower bound for general non-linear LCCs that beats the prior best $n \geq \tilde{\Omega}(k^3)$ lower bound of [AGKM23] by a polynomial factor.

Our design LCC lower bound is obtained via a fine-grained analysis of the Kikuchi matrix method applied to a variant of the matrix used in [KM23]. Our lower bounds for non-linear codes are obtained by designing a from-scratch reduction from nonlinear 3-LCCs to a system of "chain polynomial equations" — polynomial equations with similar structure to the *long chain derivations* that arise in the lower bounds for linear 3-LCCs [KM23].

Keywords: Locally Correctable Codes, Locally Decodable Codes, Kikuchi Matrices

ISSN 1433-8092

Contents

1 Introduction

A *locally correctable code* (LCC) is an error correcting code that admits, in addition, a *local* correction (a.k.a. *self-correction*) algorithm that can recover any symbol of the original codeword by querying only a small number of randomly chosen symbols from the received corrupted codeword. More formally, we say that a code $C\colon \{0,1\}^k \to \{0,1\}^n$ is q-locally correctable if for any codeword x, a corruption y of x, and input $u \in [n]$, the local correction algorithm reads at most q symbols (typically a small constant such as 2 or 3) of y and recovers the bit x_u with probability $1-\varepsilon$ whenever $\Delta(x,y) \coloneqq |\{v \in [n] : x_v \neq y_v\}| \leq \delta n$, where δ, the "distance" of the code, and ε, the decoding error, are constants. Such codes have had myriad applications (see the surveys [Tre04, Yek12, Dvi12]) starting with *program checking* [BK95], sublinear algorithms and property testing [RS96, BLR93], probabilistically checkable proofs [ALM+98, AS98], IP=PSPACE [LFKN90, Sha90], worst-case to average-case reductions [BFNW93], constructions of explicit rigid matrices [Dvi10], and q-server private information retrieval protocols [IK99, BIW10].

Reed–Muller codes, or codes based on the evaluations of multivariate degree $q-1$ polynomials over a finite field, provide a natural class of q-query locally correctable codes. For any constant $q \geq 2$, they imply binary q-LCCs with block length $n \leq 2^{O(k^{1/(q-1)})}$. These codes are in fact $\mathbb{F}_2$-linear if we view the code C as mapping $\mathbb{F}_2^k$ into $\mathbb{F}_2^n$. Despite significant effort over the past three decades, we do not know of a binary q-LCC with a smaller block length than Reed–Muller codes. This has motivated the conjecture that Reed–Muller codes are optimal q-LCCs for any constant q.

For $q = 2$, classical works [KW04, GKST06] on lower bounds on local codes confirm that the 2-LCC based on binary Hadamard codes (the special case of Reed–Muller codes when $q = 2$) achieves the smallest possible block length of $n = 2^k$ up to absolute constants in the exponent. For $q = 3$, a recent work of [KM23] improved on the best prior lower bound of $n \geq \tilde{\Omega}(k^3)$ [AGKM23], and showed that for any binary[1] linear code, $n \geq 2^{\Omega(k^{1/8})}$. As a corollary, they obtained a strong separation between 3-LCCs and the weaker notion of 3-query locally *decodable* codes — codes where the local correction algorithm only needs to succeeds for the k message bits and for which sub-exponential length constructions, i.e., $n = 2^{k^{o(1)}}$, are known [Yek08, Efr09]. The lower bound of [KM23] was very recently improved to $n \geq 2^{\Omega(k^{1/4})}$ in a follow-up work by Yankovitz [Yan24].

Despite this substantial progress, these results strongly exploit the linearity of the codes and do not yield *any* improvement over the prior cubic lower bound for non-linear codes of [AGKM23]. Even for the case of linear codes, these bounds, while exponential, still do not asymptotically match the block length of Reed–Muller codes. In this work, we make progress on both these fronts. Before discussing our results, we will take a brief detour to discuss a connection between 3-LCCs and a foundational question about the algebraic rank of combinatorial designs.

Connections to the Hamada Conjecture. Locally correctable codes have a deep connection — first formalized by Barkol, Ishai and Weinreb [BIW10] — to the widely open Hamada conjecture from the 1970s in combinatorial design theory (with deep connections to coding theory, see [AK92] for a classical reference). For positive integers m, s, λ, a 2-(m, s, λ)-design is a collection $\mathcal{B} \subseteq [m]$ of subsets (called *blocks*) of size s, such that every pair of elements in $[m]$ appears in exactly λ subsets

[1] Their results extends to codes over any field $\mathbb{F}$ of size $|\mathbb{F}| \leq k^{1-\eta}$ for a constant $\eta > 0$, more generally.

in $\mathcal{B}$. For any prime p, the p-rank of a design $\mathcal{B}$ is the rank, over $\mathbb{F}_p$, of the *incidence* matrix of $\mathcal{B}$: the 0-1 matrix with rows labeled by elements of $[m]$, columns labeled by elements of $\mathcal{B}$ and an entry (i, B) is 1 iff B contains i. A central question in algebraic design theory is understanding the smallest possible p-rank of a 2-(m, s, λ)-design.

In [BIW10], the authors showed that given any 2-(m, s, λ)-design $\mathcal{D}$ of p-rank $m - k$, the dual subspace to the column space of the incidence matrix of $\mathcal{D}$ yields a linear $(s - 1)$-query locally correctable code on $\mathbb{F}_p^m$ of dimension k. In particular, applying this transformation to the well-studied *geometric designs* yields the folklore construction of Reed–Muller locally correctable codes discussed earlier. Specifically, the 3-query locally correctable *binary* code obtained from Reed–Muller codes on $\mathbb{F}_4$ corresponds to a 2-$(n, 4, 1)$-design over $\mathbb{F}_2$ (see Appendix B).

In 1973, Hamada [Ham73] made a foundational conjecture (see [Jun11] for a recent survey) in the area that states[2] that affine geometric designs (i.e., duals to the Reed–Muller LCCs) minimize the p-rank among all algebraic designs of the same parameters. Over the past few decades, the conjecture has been confirmed in various special cases [HO75, DHV78, Tei80, Ton99] that all correspond to $s \leq 3$ or $s = n - 1$. In particular, the case of $s = 4$ (the setting of 3-LCCs) was widely open until the recent result of [KM23] for 3-LCC lower bounds. The connection between Hamada's conjecture and LCC lower bounds was suggested in [BIW10] as evidence for *the difficulty* of proving LCC lower bounds.

1.1 Theorem 1: Sharp lower bounds for design 3-LCCs.

In our first result, we obtain a bound that is sharp up to a $\sqrt{8}$ factor in the exponent on the blocklength of any binary linear 3-LCC where the local correction query sets form a 2-$(n, 4, 1)$-design. This is equivalent to asking for the local correction sets for correcting any bit of the codeword to be a *perfect* 3-uniform hypergraph matching and that every pair of codewords bits appears in exactly 2 triples across all matchings[3]. Specifically, for such design 3-LCCs, we prove:

Theorem 1. *Let $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ be a design 3-LCC. Then, $n \geq 2^{(1-o(1))\sqrt{k}}$. Here, the $o(1)$-factor is $O(\log k/\sqrt{k})$.*

Theorem 1 improves on the prior best lower bound of $n \geq 2^{\Omega(k^{1/3})}$ for designs recently obtained by Yankovitz [Yan24] building on the $n \geq 2^{\Omega(k^{1/6})}$ bound of [KM23].[4] We note that there is a technical bottleneck that prevents the proof of [Yan24] from beating a lower bound of $n \geq 2^{\Omega(k^{1/3})}$ even for the case of designs that Theorem 1 tackles (see Remark 2.4 for a more detailed explanation).

Reed–Muller codes, in particular, are design LCCs. In fact, in Appendix B we observe that the folklore best-known construction of binary 3-query LCCs — obtained by projecting Reed–Muller

[2] Hamada's original conjecture is that affine geometric designs, or, dual codes to Reed–Muller codes, are the unique optimal designs with the same parameters. This strong form has since then been disproved – there are non-affine geometric designs that achieve the *same* (but not better!) parameters [Jun84, Kan94, LLT00, LLT01, LT02, JT09]. The version of the problem we study here is called the *weak version* of Hamada conjecture.

[3] The reason that this is 2 instead of $\lambda = 1$ is because a 4-tuple (u, v, s, t) yields 2 decoding triples, (u, s, t) for v and (v, s, t) for u, that contain the pair (s, t).

[4] The stated result of [KM23] is $n \geq 2^{\Omega(k^{1/8})}$ for (non-design) linear 3-LCCs; however, their proof implicitly gives this slightly stronger bound for designs.

codes of degree-2 polynomials over $\mathbb{F}_4$ to $\mathbb{F}_2$ via the trace map — is a design 3-LCC with $n \leq 2^{\sqrt{8k}}$, or equivalently, a 2-$(n, 4, 1)$ design of rank $n - k$. Thus, the bound in Theorem 1 is *tight up to a factor of* $\sqrt{8}$ *in the exponent.* As a direct corollary, we also confirm the Hamada conjecture for 2-$(n, 4, 1)$-designs up to a factor of 8 in the co-dimension.

Towards obtaining a $k \leq O(\log^2 n)$ bound for all linear 3-LCCs. Given our almost sharp lower bound for design 3-LCCs, we can use our proof to attribute the "extra" $\log^2 n$ factor loss in [KM23, Yan24] to certain "irregularities" of general linear 3-LCCs. Concretely, there are two places where the proof of [KM23, Yan24] is lossy: (1) there is a "hypergraph decomposition" step to handle that pairs of codeword bits may appear in $\gg O(1)$ triples across all matchings (one $\log n$ loss), and (2) there is a "row pruning" step to argue that a certain graph is approximately regular (one $\log n$ loss). For the case of designs, [Yan24] proves a $k \leq O(\log^3 n)$ bound since there is no "hypergraph decomposition" needed for designs as each pair of codeword bits appears in a bounded number of triples. Our proof of Theorem 1 additionally shows that because the hypergraph matchings in the design are *perfect*, we can (via this work's modified approach, see Remark 2.4) mitigate the $\log n$ factor loss in the "row pruning" step.

The sharpness of our bound for design 3-LCCs may be somewhat surprising because removing an analogous "last" $\log n$ factor in the hypergraph Moore bound (also proved via the Kikuchi matrix method) and related problems remains a challenging problem [GKM22, HKM23, HKM+24]. In this setting, making extra structural assumptions about the hypergraph, analogous to the additional structure of design 3-LCCs, does not seem to help.

1.2 Theorem 2: Superpolynomial lower bounds for smooth 3-LCCs.

In our second result, we obtain improved lower bounds for smooth 3-LCCs with high completeness. These codes may be non-linear and may have adaptive correction algorithms.

A 3-LCC is said to be δ-*smooth* if no codeword bit is queried with probability more than $\frac{1}{\delta n}$ on any particular invocation of the decoder. Introduced by Katz and Trevisan [KT00], smooth codes provide a clean formalization of general locally correctable/decodable codes. We say that such a code has completeness $1 - \epsilon$, if, when running the δ-smooth local correction algorithm on an *uncorrupted* codeword, the algorithm succeeds with probability at least $1 - \epsilon$. Recall that the usual notion of completeness (e.g., in [KT00]) for LCCs is for an input with a δ-fraction of corruptions.

Our result shows that for any $(1 - \epsilon)$-complete δ-smooth code where δ is a constant, $n \geq k^{O(1/\epsilon)}$. In particular, when ϵ is subconstant, we obtain a superpolynomial lower bound on the block length. For technical reasons (explained in Section 2.3), the bound does not improve when ε becomes very small, namely $o(1/\sqrt{\log n})$.

Theorem 2. *There is an absolute constant $\gamma > 0$ such that the following holds. Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a δ-smooth (possibly non-linear) 3-LCC with completeness $1 - \varepsilon$ where $\varepsilon \leq \gamma/\sqrt{\log_2 n}$. Then, $n \geq (k')^{\Omega(\log k')}$, where $k' = \delta^3 k/\log(1/\delta)$.*

If instead $\varepsilon \geq \gamma/\sqrt{\log_2 n}$, then $n \geq \tilde{\Omega}((k'')^t)$ where $k'' = \delta^3 \varepsilon^4 k/\log(1/\delta)$ and $t = \lfloor \frac{1}{2\varepsilon} - \frac{1}{\log_2 n} \rfloor$. In particular, if ε is a small enough constant and $1/2\varepsilon$ is not an integer, then $n \geq \tilde{\Omega}((k')^{\frac{1}{2\varepsilon}})$.

As we shall discuss towards at the end of this section, Theorem 2 implies a lower bound for general $(3, \delta, \varepsilon)$-LCCs that beats the prior best $n \geq \tilde{\Omega}(k^3)$ lower bound of [AGKM23] by a polynomial factor when ε is a small constant. Moreover, in the case of near-perfect completeness, our result above obtains the first superpolynomial lower bound for (possibly adaptive and nonlinear) smooth 3-LCCs.

Our proof is based on the method of spectral refutation via Kikuchi matrices (first introduced in [WAM19] for an application to Gaussian tensor PCA and refutation of random k-XOR instances of even arity) developed in prior works [GKM22, HKM23, AGKM23, KM23]. The key idea in this method is to associate the existence of a combinatorial object (e.g., a 3-LCC) to the satisfiability of a family of XOR formulas and find a spectral *refutation* (i.e., certificate of unsatisfiability) for a randomly chosen member of the family. Unlike the works of [KM23, Yan24], which only prove lower bounds for linear codes with an argument that can be reformulated to be entirely combinatorial, the proof of Theorem 2 crucially uses the power of spectral refutation.

The proof of Theorem 2 requires new conceptual ideas. The first immediate observation is that the standard reduction of [KT00] to nonadaptive, linear decoders that succeeds in expectation loses a large factor in the completeness parameter. So, to prove Theorem 2 we need to come up with a new reduction *from scratch*. Our reduction is based on two new key ideas. First, we execute the "long chain derivation" strategy of [KM23] by *adaptively* forming chains by replacing the third query v_3 of the decoder by an invocation of the decoding algorithm for v_3 (and then iterating); we call such chains "adaptive chains". Second, we *exactly* encode the behavior of the LCC decoder on a particular input index u as a degree ≤ 3 polynomial. Effectively, the constraints we uncover from the decoder are "AND" constraints:

$$\text{``}x_{v_1} = a_1 \wedge x_{v_2} = a_2 \implies x_u = x_{v_3}\text{,''}$$

rather than the linear constraints $x_{v_1} + x_{v_2} + x_{v_3} = x_u$ encountered for linear codes. Combining these two ideas allows us to write a "chain polynomial" that plays the role of the "long chain derivations" in [KM23]. Refuting this "chain polynomial" using spectral bounds on Kikuchi matrices yields Theorem 2.

Smooth vs. general LCCs. Smooth LCCs (Definition 3.2) were defined in the work of [KT00], motivated by their connection to general LCCs (Definition 3.1). A simple reduction in [KT00] shows that any $(3, \delta, \varepsilon)$-LCC, i.e., an LCC with distance δ and completeness $1 - \varepsilon$, can be turned into a $(3, \delta/3, \varepsilon)$-smooth LCC, i.e., a $\delta/3$-smooth 3-LCC with completeness $1 - \varepsilon$. Conversely, any $(3, \delta, \varepsilon)$-smooth LCC is a $(3, \eta\delta, \varepsilon + \eta)$-LCC for any $\eta > 0$.

Thus, when ε is a small constant, Theorem 2 implies a lower bound for general $(3, \delta, \varepsilon)$-LCCs that beats the prior best $n \geq \tilde{\Omega}(k^3)$ lower bound of [AGKM23] by a polynomial factor.

However, in the setting of perfect completeness (and $\varepsilon = o(1)$ more generally), the comparison between smooth LCCs and general LCCs begins to break down. This is because, for a general LCC, δ is the fraction of errors one can tolerate while still decoding correctly with probability $1 - \varepsilon$; the parameters δ and ε are coupled! In particular, it is likely not possible to simultaneously have $\varepsilon = o(1)$, $\delta = O(1)$ and $q = O(1)$. On the other hand, for a smooth LCC, δ is the smoothness parameter, and $1 - \varepsilon$ is the probability that the decoder succeeds *on an uncorrupted codeword*. Thus, for smooth codes, it is perfectly sensible to set $\delta = O(1)$, $\varepsilon = 0$, and $q = O(1)$.

In retrospect, the definition of LCCs inherently couples δ and the completeness ε, whereas for smooth codes these parameters become independent. In particular, a smooth code allows us to seamlessly trade off between the fraction of errors $\eta\delta$ tolerated and the success probability $1 - \varepsilon - \eta$ of the decoder in the presence of this fraction of errors. For this reason, a smooth code is a stronger object, but also perhaps a more natural one.

Indeed, in some important applications of LDCs/LCCs, smooth LDCs/LCCs are the right notion to consider. For example, a *perfectly smooth* $(q, 1, \varepsilon)$-smooth LDC gives a q-server information-theoretically secure private information retrieval scheme with completeness $1 - \varepsilon$.

The subtle definitional issues above did not affect prior lower bound (or upper bound) techniques. Indeed, known constructions of q-LDCs and LCCs are *perfectly* smooth and satisfy *perfect* completeness, i.e., $(q, 1, 0)$-smooth LDCs/LCCs, and the lower bound techniques of [KT00, KW04, AGKM23] (that is, the best known lower bounds before the work [KM23]) succeed for smooth LDCs/LCCs even with *low* completeness.

Concurrent work. In concurrent and independent work, [AG24] proves an $n \geq 2^{\Omega(\sqrt{k/\log k})}$ lower bound for all linear 3-LCCs over $\mathbb{F}_2$, improving on the $2^{\Omega(k^{1/4})}$ shown in [Yan24]. This is incomparable to Theorem 1, as it proves a weaker (and possibly not tight) lower bound, as compared to the sharp statement in Theorem 1, but it applies for all linear 3-LCCs over $\mathbb{F}_2$, not just design 3-LCCs. The work of [AG24] does not prove any lower bound for nonlinear codes.

2 Proof Overview

In this section, we summarize the key conceptual ideas that we use in the proofs. We start by recalling the approach of [KM23] for proving lower bounds for *linear* 3-LCCs. Then, we explain the technical barriers to proving Theorem 1 encountered in the works of [KM23, Yan24]. Finally, we discuss our approach to handling the nonlinear case.

2.1 The approach of [KM23]

The proof of [KM23] gives a transformation that takes any linear 3-LCC $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ and turns it into a 2-LDC $\mathcal{L}' \colon \{0,1\}^k \to \{0,1\}^N$, where $N = n^{O(\mathrm{polylog}(n))}$. By applying known 2-LDC lower bounds (Fact 3.10), we can then conclude that $k \leq O(\mathrm{polylog}(n)) \cdot \log n$. Obtaining better lower bounds thus boils down to optimizing the polylog(n) factor here and/or removing the extra $\log n$ factor from Fact 3.10.

For intuition, let us think of N as $N = \binom{n}{s}^r$ for some choice of parameters s and r, and $\mathcal{L}'$ as the very simple transformation: for sets $(S_1, \dots, S_r)$, each in $\binom{[n]}{s}$, we set $\mathcal{L}'(b)_{(S_1,\dots,S_r)} = \sum_{h=1}^{r} \sum_{v \in S_h} x_v$, where $x = \mathcal{L}(b)$. Namely, we just take the XOR of the bits across all the sets. If we can show that $\mathcal{L}'$ is indeed a 2-LDC, then we can apply known 2-LDC lower bounds (Fact 3.10) to conclude that $k \leq O(\log N) = O(rs \log n)$. If we can then take $rs = O(\log n)$, or $r = s = O(\log n)$ while removing the extra $\log n$ factor from Fact 3.10, we will get an $O(\log^2 n)$ bound, i.e, $n \geq 2^{\Omega(\sqrt{k})}$.

Recall that in a linear 3-LCC, we are given 3-uniform hypergraph matchings $\{H_u\}_{u \in [n]}$, such that for each $C \in H_u$, $\sum_{v \in C} x_v = x_u$ for all codewords $x \in \mathcal{L}$. To show that $\mathcal{L}'$ is a 2-LDC, we need to find, for each $i \in [k]$, many pairs of vertices $((S_1, \dots, S_r), (T_1, \dots, T_r))$ such that $\sum_{h=1}^r \sum_{v \in S_h} x_v + \sum_{h=1}^r \sum_{v \in T_h} x_v = b_i$ for all $x = \mathcal{L}(b)$. The key idea of [KM23] is to build many such constraints by building *long chain derivations* out of the original constraints H_u.

Definition 2.1 (r-chains)**.** Let $H_1, \dots, H_n$ be the 3-uniform hypergraph matchings defined from the 3-LCC $\mathcal{L}$. An r-chain with *head* u_0 is an ordered sequence of vertices of length $3r + 1$, given by $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r)$ and for each $h = 0, \dots, r - 1$, it holds that $\{v_{2h+1}, v_{2h+2}, u_{h+1}\} \in H_{u_h}$. We let $\mathcal{H}_u^{(r)}$ denote the set of r-chains with head u.

We let $C_L = (v_1, v_3, v_5, \dots, v_{2(r-1)+1})$ denote the "left half" of the chain, and $C_R = (v_2, v_4, v_6, \dots, v_{2(r-1)+2})$ denote the "right half". We call u_r the "tail".

The number of chains $|\mathcal{H}_u^{(r)}|$ is at most $(6\delta n)^r$.

The 2-chains can be interpreted as deriving constraints by taking two constraints $x_{v_1} + x_{v_2} + x_{u_1} = x_{u_0}$ in H_{u_0} and $x_{v_3} + x_{v_4} + x_{u_2} = x_{u_1}$ in H_{u_1}, and then adding them together to produce the constraint $x_{v_1} + x_{v_2} + x_{v_3} + x_{v_4} + x_{u_2} = x_{u_0}$; the set of r-chains is formed by repeating this operation. We have $(6\delta n)^r$ chains in $\mathcal{H}_u^{(r)}$ because there are $6\delta n$ *ordered* hyperedges in H_u.

The next idea of [KM23] is to use a Kikuchi graph to (1) define N and the map $\mathcal{L}'$, and (2) define the 2-LDC decoding constraints for $\mathcal{L}'$. For this overview, we will start with the following graph due to [Yan24], which is a bit simpler than the graphs used in [KM23] as it saves a use of the Cauchy-Schwarz inequality.

Definition 2.2 (Imbalanced Kikuchi graph)**.** Let s be a parameter, and let G_u be the graph with left vertex set $L = \binom{[n]}{s}^r \times [n]$ and right vertex set $R = \binom{[n]}{s}^r$. For a chain $C \in \mathcal{H}_u^{(r)}$, we add an edge between $((S_1, \dots, S_r), w)$ and $(T_1, \dots, T_r)$ in G_u "labeled" by C if $w = u_r$ and for each $h = 1, \dots, r$, we have $S_h = \{v_{2(h-1)+1}\} \cup U_h$ and $T_h = \{v_{2(h-1)+2}\} \cup U_h$ for some $U_h \subseteq [n] \setminus \{v_{2(h-1)+1}, v_{2(h-1)+2}\}$ of size $s - 1$. Two distinct chains may produce the same edge — we add edges with multiplicity.

To show that $\mathcal{L}'$, defined now as a map from $\{0,1\}^k \to \{0,1\}^{L \cup R}$ in an analogous way, is a 2-LDC, we need to show that for each u, G_u admits a large matching. An obvious barrier to this is that the graph is bipartite and imbalanced, and so the largest matching can only have size at most $|R| = |L|/n$. This be fixed with a simple trick: for each right vertex $(T_1, \dots, T_r)$, we can add n copies of the vertex to the graph and then split its edges evenly across the copies, thereby decreasing the degree by a factor of n.[5]

Extracting a large matching from G_u. Let us now explain the approach of [KM23] to show that G_u admits a large matching. We note that this is the *key technical difficulty* in the proof of [KM23], and in some sense, this has to be the difficult step because it will prove that $\mathcal{L}'$ is a 2-LDC!

Let $d_{u,L}$ denote the average left degree of G_u, and let $d_{u,R}$ denote the average right degree. For G_u to have a large matching, it should, at the very least, have at least $|L| = n\binom{n}{t}^r$ edges!

Some simple combinatorics shows that each chain $C \in \mathcal{H}_u^{(r)}$ contributes exactly $\binom{n-2}{s-1}^r$ edges to

[5] Technically, the degree might not be divisible exactly by n, so reduces the degree by a factor of $n(1 - o(1))$, which is sufficient.

the graph G_u. Therefore, as long as we have

$$d_{u,L} = |\mathcal{H}_u^{(r)}| \frac{\binom{n-2}{s-1}^r}{n\binom{n}{s}^r} = (1 \pm o(1))(6\delta n)^r \frac{1}{n} \left(\frac{s}{n}\right)^r = (1 \pm o(1))(6\delta s)^r \frac{1}{n} \gg 1\,,$$

then we can hope to find a large matching. Note that for this to hold, we *must* set $r = O(\log n / \log s)$.

One simple way to find a matching in G_u is to argue that G_u is approximately regular, meaning that most vertices have degree $\le O(d_{u,L})$. If this were the case, then (after the "vertex splitting trick") we get a matching of size $|E(G_u)|O(d_{u,L}) \ge \Omega(|L|)$, which would finish the proof. Unfortunately, this is not true: there can be left (right) vertices in G_u of degree $\gg d_{u,L}$ ($\gg d_{u,R}$). The "row pruning" strategy of [KM23] is to show that such vertices are rare so that by removing them we obtain a graph G'_u with $\Omega(|E(G_u)|)$ edges that has bounded left degree $\le O(d_{u,L})$ and bounded right degree $\le O(d_{u,R})$.

More formally, the proof of [KM23] uses a form of a Kim–Vu concentration inequality [KV00, SS12] for polynomials to argue that, with high probability, a *random* left (right) vertex has degree at most $O(d_{u,L})$ ($O(d_{u,R})$), which finishes the proof. This is the key technical "row pruning" step in the proof, and the proof uses the moment method with high moments. Unfortunately, to prove this, [KM23] requires $s = O(r^3 \log n)$, which loses several extra $\log n$ factors.

The clever trick of [Yan24], when phrased in the language of probability, can be interpreted as follows: we can achieve the same goals by only bounding the *second* moments of the degrees instead of the higher moments. This is similar in spirit to the "edge deletion" technique of [HKM23], which used a similar argument to remove some of the $\log n$ factors from a different "row pruning" argument of [GKM22] that used Kim–Vu concentration inequalities [KV00, SS12] in the context of CSP refutation.

Specifically, if $\deg_{u,L}(S_1, \dots, S_r, w)$ is the left degree of the vertex $(S_1, \dots, S_r, w)$ and $\deg_{u,R}(T_1, \dots, T_r)$ is the right degree of $(T_1, \dots, T_r)$, then [Yan24] shows that

$$\mathbb{E}_{S_1,\dots,S_r,w}[\deg_{u,L}(S_1, \dots, S_r, w)^2] \le (1 + o(1)) d_{u,L}^2\,,$$
$$\mathbb{E}_{T_1,\dots,T_r}[\deg_{u,R}(T_1, \dots, T_r)^2] \le (1 + o(1)) d_{u,R}^2\,,$$

where $d_{u,L}$ and $d_{u,R}$ are the first moments, and we only need $s = \Gamma r$ and $r = O(\log n)$, where Γ is a large enough constant, for this to hold.

Thus, applying Chebyshev's inequality, we can show that after removing a small number of vertices, we can find a subgraph G'_u of G_u where each vertex has left degree $\le (1 + o(1))d_{u,L}$ and right degree $\le (1 + o(1))d_{u,R}$, which, after a few more straightforward steps, finishes the proof.

In total, the final lower bound is $k \le O(rs \log n) = O(\log^3 n)$, as we have set $r, s = O(\log n)$.

Remark 2.3. This sketches the proof of Theorem 1.6 in [Yan24] for design 3-LCCs. We note that the reason [Yan24] obtains a weaker $k \le O(\log^4 n)$ bound for linear 3-LCCs is because a general 3-LCC can have "heavy pairs" — pairs of variables (v_1, v_2) that appear in many hyperedges across all the H_u's — and this loses the extra $\log n$ factor. Indeed, overcoming this issue to produce *any* lower bound better than the $k \le \tilde{O}(n^{1/3})$ of [AGKM23] is one of two key technical difficulties in [KM23] (the above row pruning argument is the other one). However, as Theorem 1 is only for designs, this issue does not arise.

2.2 Tight bounds for designs: proof sketch of Theorem 1

To beat the $O(\log^3 n)$ of [Yan24] for designs and get $O(\log^2 n)$, we need to find a $\log n$ factor to remove. At the very least, we know we cannot hope to take r much smaller than $\log n$, as we need $s^r \gg n$ for the entire approach to even make sense. So, there are two possibilities: either we can take $s = O(1)$, or the $O(rs \log n)$ bound coming from Fact 3.10 is not tight for the 2-LDC that we produce, and the truth is really $O(rs)$. Let us now investigate the first case, as if we could take $s = O(1)$ this would be the easiest route towards proving Theorem 1.

Second moments of degrees are large for $s \ll r$. Unfortunately, as we shall show, we cannot take $s = O(1)$, or even s much smaller than r, and still have the following moment bounds:

$$\mathbb{E}_{S_1,\dots,S_r,w}[\deg_{u,L}(S_1,\dots,S_r,w)^2] \leq O(d_{u,L}^2)\,,$$
$$\mathbb{E}_{T_1,\dots,T_r}[\deg_{u,R}(T_1,\dots,T_r)^2] \leq O(d_{u,R}^2)\,.$$

Indeed, let us compute $\mathbb{E}_{T_1,\dots,T_r}[\deg_{u,R}(T_1,\dots,T_r)^2]$. Recall that we will compare it to $d_{u,R}^2$, and we have already computed $d_{u,R} = n \cdot d_{u,L} = (1 \pm o(1))(6\delta s)^r$. To do the computation, we will need to use the number of pairs of chains $C, C' \in \mathcal{H}_u^{(r)}$ with $|C_R \cap C'_R| = t$ is at most $\binom{r}{t} \cdot 2^{2r}(3\delta n)^{2r-t}$. Let us denote $C_R = (v_2, v_4, v_6, \dots, v_{2(r-1)+2})$ and $C'_R = (v'_2, v'_4, v'_6, \dots, v'_{2(r-1)+2})$

$$\begin{aligned}
\mathbb{E}_{T_1,\dots,T_r}[\deg_{u,R}(T_1,\dots,T_r)^2] &\leq \sum_{t=0}^{r} \sum_{C,C' \in \mathcal{H}_u^{(r)} : |C_R \cap C'_R| = t} \Pr[v_{2h+2}, v'_{2h+2} \in T_{h+1}\ \forall h \in \{0,\dots,r-1\}] \\
&\leq \sum_{t=0}^{r} \sum_{C,C' \in \mathcal{H}_u^{(r)} : |C_R \cap C'_R| = t} \frac{\binom{n}{s-1}^t \binom{n}{s-2}^{r-t}}{\binom{n}{s}^t} \text{ (if } v_{2h+2} = v'_{2h+2}\text{, then } T_h \text{ has} \leq \binom{n}{s-1} \text{ choices, else} \leq \binom{n}{s-2} \text{ choices)} \\
&\leq \sum_{t=0}^{r} \sum_{C,C' \in \mathcal{H}_u^{(r)} : |C_R \cap C'_R| = t} (1 + o(1)) \left(\frac{s}{n}\right)^t \left(\frac{s}{n}\right)^{2r-2t} \quad \text{(by binomial coefficient estimates)} \\
&\leq (1 + o(1)) \sum_{t=0}^{r} \binom{r}{t} \cdot 2^{2r}(3\delta n)^{2r-t} \left(\frac{s}{n}\right)^{2r-t} \quad \text{(from the bound on number of pairs } (C, C')\text{)} \\
&\leq (1 + o(1))(6\delta s)^{2r} \sum_{t=0}^{r} \binom{r}{t} (3\delta s)^{-t} \\
&\leq (1 + o(1)) d_{u,R}^2 \sum_{t=0}^{r} \binom{r}{t} (3\delta s)^{-t}\,.
\end{aligned}$$

The problem with the moment bound is now readily apparent. For small t, $\binom{r}{t}$ is roughly r^t, and so we need $s \gg r/3\delta$ so that $(3\delta s)^{-t} r^t \ll 1$. Hence, if we take $s = o(r)$, the second moment is large.

Still, one may wonder if our estimate on the second moment here is tight. Perhaps it is simply that our bound on the moment is large, but the true second moment is not. It turns out that when the H_u's are near-perfect matchings (which is the case for design 3-LCCs!), i.e., $3\delta = 1 - o(1)$, then this bound is tight up to a $1 + o(1)$ factor. This implies that the right degrees truly have high variance (a similar calculation shows this for left degree also). So, it is unlikely that one can find a subgraph G'_u with $\Omega(|E(G_u)|)$ edges and, say, right degree bounded by $O(d_{u,R})$.

Remark 2.4. In Theorem 1, our notion of designs requires the matchings H_u to be perfect matchings, which is a stronger definition from the one used in [Yan24]. One might be worried that our improvement in the lower bound for designs is therefore primarily due to the initial assumption being a bit stronger, rather than for any real technical improvements. The above observation that the second moment bound is tight for perfect matchings not only shows that this is not the case, but also that perfect matchings are *the* case where the second moments are too large.

It is still potentially possible that the graph G_u admits a large matching, even though the second moments are large. While we have not formally ruled this out, the second moment calculation informally tells us that one probably needs a substantially different approach to prove this.

Removing a $\log n$ factor from Fact 3.10? We have argued that we cannot take $s = O(1)$ so that $O(rs \log^2 n)$. What about the other approach, where we try to shave off the extra $\log n$ coming from Fact 3.10 to get a bound of $O(rs)$? This is not something that can be done *generically* for all 2-LDCs, as of course the Hadamard code is a 2-LDC with $k = \log_2 n$.

Shaving this $\log n$ factor is closely related to removing the $\log n$ factor from Matrix Khintchine (Fact 3.11), a task studied in many different contexts [BSS14, MSS15, BJM23].

One such example is the hypergraph Moore bound: the task of showing that a q-uniform hypergraph on n vertices with $(n/\ell)^{q/2}\ell$ must have a cycle (also called an even cover) of length $O(\ell \log(n/\ell))$. The best bound for this problem is due to the methods of [GKM22, HKM23], which uses Kikuchi graphs similar to Definition 2.2 to show the existence of a length $O(\ell \log n)$ cycle when the hypergraph has $(n/\ell)^{q/2}\ell \cdot \log n$ hyperedges, a $\log n$ factor larger than the conjectured threshold. A more complicated argument manages to reduce the $\log n$ factor to $(\log n)^{\frac{1}{q+1}}$ when q is odd [HKM+24], and this proof requires a rather technical modification of the Kikuchi graph.

One might thus naturally surmise that a rather complicated modification of the graphs G_u, perhaps along the lines of [HKM+24], is necessary for us to obtain a sharp lower bound via the Kikuchi matrix method.

Our new Kikuchi graph. To our surprise, and perhaps contrary to the intuition developed above, it turns out that the following *simple* modification of the graphs succeeds: instead of indexing the vertices by collections of sets $(S_1, \dots, S_r)$ or $(T_1, \dots, T_r)$, each of size s, we index by "big" sets S and T of size ℓ (Definition 4.3). This graph is essentially the same as the previous graph if we $\ell = rs$ — the advantage of this new graph is that if we take $s = O(1)$, then the sets have size $\ell = O(r)$, which is still large, in contrast to the previous graph where the constituent sets S_h would only have size $s = O(1)$. We shall call this new graph the "uncolored graph", as we think of the graphs in Definition 2.2 as consisting of a big set $S = S_1 \cup \cdots \cup S_r$, where S_h contains vertices from $[n]$ with *color* h, which makes the S_h's disjoint by fiat as they use different colors.

In some sense, the above graph is *more* natural than even the ones appearing in [KM23, Yan24]. Of course, the reason those works use the colored graph rather than the uncolored one is that the colored graph makes some of the combinatorial analysis more tame! Indeed, this is because the r colors specify which vertices should appear in the r "links" of the chain, namely, vertices that are in S_h appear in the "h-th link", and correspond to choices only of $v_{2(h-1)+1}$. One can note (as observed in Definition 4.3) that the uncolored graph does not have edges for all chains $C \in \mathcal{H}_u^{(r)}$. Indeed, if the left half C_L or the right half C_R contains duplicate vertices, then C does not contribute *any* edges! And, if they share vertices, then they contribute many more edges than a chain where all vertices in C_L,C_R are distinct. Fortunately, as we observe in Definition 4.1, there are at least $(1-o(1))|\mathcal{H}_u^{(r)}|$ chains such that the vertices in C_L and C_R are distinct, so we can ignore these issues by working with this large subset of chains.

In spite of these technical challenges, we can make the analysis work for the uncolored graph. This allows us to take $\ell = 2r$ for $r = \frac{1}{2}\log_2 n + O(\log\log n)$, and gives us a final bound of $k \leq (1+o(1))2r\log_2 n$, i.e., $((1-o(1))k)^2 \leq \log_2 n$, which gives us Theorem 1. We note that to get the sharp constant in Theorem 1, we have to show that G_u admits a *near-perfect* matching.

The calculations involved here are sensitive and require sharp bounds on binomial coefficients. To give the reader a sense of how precise these bounds are, we observe that our proof shows that

$$\mathbb{E}_{(S,w)}[\deg_{u,L}(S,w)^2] = (1 \pm o(1))d_{u,L}^2 \cdot (3\delta)^{-1}\sum_{t=0}^{r-1}(3\delta)^{-t}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}},$$

$$\mathbb{E}_T[\deg_{u,R}(T)^2] = (1 \pm o(1))d_R^2\sum_{t=0}^{r}(3\delta)^{-t}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}}.$$

How do we bound the sum $\sum_{t=0}^{r}(3\delta)^{-t}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}}$? First of all, $\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}}$ is the probability mass function of a hypergeometric distribution, and so $\sum_{t=0}^{r}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$. Note that the mean of this distribution is r^2/ℓ and it has good concentration, so we should expect to lose a factor of $(3\delta)^{-t^*}$ where $t^* = r^2/\ell$.

In particular, if 3δ is bounded away from 1, say, 1/2, then we need to take $\ell = O(r^2)$ to mitigate this factor, and we so get *no improvement*. But, if we have a *design*, then $3\delta = 1 - \frac{1}{n}$, and so $(3\delta)^{-r}$ is only a $1 + o(1)$ factor and thus does not matter!

Finally, we note that since $\sum_{t=0}^{r}\frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$, we do need to be precise in our estimates above. In particular, standard estimates on binomial coefficients such as $\binom{\ell}{r} \geq \left(\frac{\ell}{r}\right)^r$ are insufficient.

We give the full proof of Theorem 1 in Section 4.

2.3 Superpolynomial lower bounds for nonlinear smooth 3-LCCs

We now explain the key ideas in the proof of Theorem 2. In this section, we will let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a nonlinear code, namely we will use $\{-1,1\}$-notation rather than $\{0,1\}$-notation, as it is more convenient for the proof.

Existing reductions do not work with the long chain derivation method of [KM23]. The standard starting point for lower bounds for nonlinear q-LDCs or q-LCCs is a reduction from the original work of Katz and Trevisan [KT00]. For 3-LCCs, this reduction takes any δ-smooth code C with completeness even as low as $\frac{1}{2} + \eta$ and outputs 3-uniform hypergraph matchings H_u for $u \in [n]$ with the following property: for every $u \in [n]$ and hyperedge $C \in H_u$, $\mathbb{E}_x[x_u \prod_{v \in C} x_v] \geq \Omega(\eta)$, where the expectation is over a uniformly random codeword $x \in \mathcal{C}$. That is to say, *every* hyperedge decodes correctly with some constant advantage *in expectation* over a random codeword.

We can now form chains (Definition 2.1) on the hypergraphs H_u. This gives us a "chain XOR instance" $\Psi_u(x)$ defined as $\sum_{C \in \mathcal{H}_u^{(r)}} x_u x_w \prod_{v \in C_L} x_v \prod_{v \in C_R} x_v$, where C_L and C_R are the left and right halves of the chain C and w is the tail of C. Unlike in the linear case, it is not guaranteed that these equations are all satisifiable, and so the "reduction-based approach of [KM23, Yan24] to 2-LDCs breaks down.[6] However, the "spectral refutation approach" of [KM23] is sufficiently general and resilient enough that, if we could show that $\mathbb{E}_{x \in \mathcal{C}}[\Psi_u(x)] \geq \eta |\mathcal{H}_u^{(r)}|$ for η even as small as $\eta \gg \frac{1}{n^{1/3}}$, then we could at the very least beat the cubic lower bound of [AGKM23]. And, if we could take η to be constant, we would get an exponential lower bound.

Unfortunately, we cannot show any lower bound on $\mathbb{E}_x[\Psi_u(x)]$. Indeed, let us even consider the simple case of length 2-chains, and let us try to bound, for $C = \{v_1, v_2, u_1\} \in H_u$ and $C' = \{v_3, v_4, u_2\} \in H_{u_1}$, the quantity $\mathbb{E}_{x \in \mathcal{C}}[x_u x_{v_1} x_{v_2} x_{v_3} x_{v_4} x_{u_2}]$. This is clearly $\mathbb{E}_{x \in \mathcal{C}}[(x_u x_{v_1} x_{v_2} x_{u_1})(x_{u_1} x_{v_3} x_{v_4} x_{u_2})]$, and while we know that the expectation of each term is $\geq \eta$, this does not imply anything on the expectation of the product.

This now suggests the following simple way to recover a lower bound: simply assume that the completeness is $1 - \varepsilon$, with the intuition being that η is related to $1 - \varepsilon$, and if η is close to 1 then we can apply a union bound. However, the reduction of [KT00] is lossy with respect to the completeness parameter. Indeed, this is because the reduction first makes the decoder nonadaptive by simulating the adaptive decoder by giving it random answers, and this takes $1 - \varepsilon$ completeness to $\frac{1}{2} + \eta$ for $\eta = \frac{1}{8}(1 - \varepsilon)$, which is too small for the union bound strategy to succeed.

Key idea 1: adaptive chains and the adaptive chain decoder. As explained above, the reduction of [KT00] is lossy, so we need to rethink the whole approach. Our intuition is that our reduction should try to remember as much information about the decoder as possible. In particular, this means that we need to remember information about the (possibly adaptive) decoder, and cannot use standard reductions to convert adaptive decoders into nonadaptive ones. At a high level, our new strategy is to form chains *before* applying the reduction of [KT00].

The first insight we have is that we can form chains *adaptively* by invoking the adaptive decoder $\mathrm{Dec}^x(u)$ in a structured way. Namely, define the "adaptive chain decoder" $\mathrm{Dec}_r^x(u)$ to be the decoder that works as follows:

(1) Simulate $\mathrm{Dec}^x(u)$ to generate the first query v_1. Then, read $a_1 = x_{v_1}$ and respond with a_1 to the simulated $\mathrm{Dec}^x(u)$ instance.

(2) The simulated $\mathrm{Dec}^x(u)$ generates a second query v_2. Then, read $a_2 = x_{v_2}$ and respond with a_2 to the simulated $\mathrm{Dec}^x(u)$ instance.

(3) The simulated $\mathrm{Dec}^x(u)$ generates a third query u_1, but $\mathrm{Dec}_r x(u)$ *does not* make this query. Instead, $\mathrm{Dec}_r x(u)$ invokes $\mathrm{Dec}^x(u_1)$, and then proceeds starting from Step (1).

[6] We note here that one of $\log n$ factors saved by [Yan24] is in optimizing the "hypergraph decomposition" step in [KM23], but this optimization is specific to linear codes. In this paper, we give a slightly tighter analysis of the decomposition in [KM23] that also saves this $\log n$ factor, and additionally generalizes to the nonlinear setting.

After r iterations of the loop, $\mathrm{Dec}_r^x(u)$ makes the final query u_r to receive a_r, and then "feeds answers backwards". Namely, the simulated $\mathrm{Dec}^x(u_{r-1})$ now outputs some guess a_{r-1} for $x_{u_{r-1}}$, which $\mathrm{Dec}_r^x(u)$ uses to answer the query u_{r-1} made by the simulated $\mathrm{Dec}^x(u_{r-2})$, etc. Finally, $\mathrm{Dec}_r^x(u)$ outputs the same answer as the first simulated $\mathrm{Dec}^x(u)$.

We can think of the decoder $\mathrm{Dec}_r^x(u)$ as generating *adaptive chains*, which are sequences of the form $(u_0, (v_1, a_1), (v_2, a_2), u_1, (v_3, a_3), (v_4, a_4), \dots)$.

Key idea 2: representing the decoder as a polynomial. Let us assume that the decoder succeeds with probability 1 for simplicity. Now that we have adaptive chains, we need to define a polynomial $\Psi_u(x)$ using the adaptive chains C in $\mathcal{H}_u^{(r)}$ (now redefined to be the set of adaptive chains) so that $\Psi_u(x)x_u = 1$ for all $x \in \mathcal{C}$. Our key idea is to encode the behavior of $\mathrm{Dec}^x(u)$ as a certain polynomial. Then, forming chains corresponds to taking certain "chain products" of these polynomials, which defines a "chain polynomial" $\Psi_u(x)$ that we will refute.

We can represent $\mathrm{Dec}^x(u)$ as a decision tree, and for simplicity, let us assume that $\mathrm{Dec}^x(u)$ makes exactly 3 queries and has perfect completeness. First, it has some distribution that it uses to generate the first query v_1. Then, it has a branch for each answer $a_1 \in \{-1, 1\}$ that it could receive. After the branch, it has another distribution to generate the second query v_2, and then another branch for each answer $a_2 \in \{-1, 1\}$. It then has a final distribution for the query v_3, and then it has a "decoding function" $f_{v_1,a_1,v_2,a_2,v_3}(a_3)$. Notice that we are allowed to have a different decoding function for each choice of "adaptive constraint" $C = (v_1, a_1, v_2, a_2, v_3)$, and so $f_{v_1,a_1,v_2,a_2,v_3}(a_3)$ need only depend on a_3. Because $\mathrm{Dec}^x(u)$ decodes with probability 1, we must have that for any $x \in \mathcal{C}$ with $x_{v_1} = a_1, x_{v_2} = a_2$, it holds that $f_{v_1,a_1,v_2,a_2,v_3}(x_{v_3}) = x_u$. Additionally, this implies that f_{v_1,a_1,v_2,a_2,v_3} is deterministic, and so it is one of the following 4 functions: $1, -1, a_3, -a_3$. For simplicity, let us pretend that all such decoding functions are simply a_3.

The above analysis effectively gives us constraints of the form

$$\text{“}x_{v_1} = a_1 \wedge x_{v_2} = a_2 \implies x_u = x_{v_3}\,.\text{”}$$

We can represent these constraints as polynomials by the AND polynomial (Definition 5.1): we then have polynomial constraints $\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{v_3} x_u = \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$.

Unlike in the linear case, the constraints come with weights, corresponding to the probability that the decoder $\mathrm{Dec}^x(u)$ makes the queries $C = (v_1, a_1, v_2, a_2, v_3)$ where x satisfies $x_{v_1} = a_1, x_{v_2} = a_2$. Let $\mathrm{wt}_u(C)$ be the weight of such a constraint. We then have for any $x \in \mathcal{C}$:

$$\sum_{C=(v_1,a_1,v_2,a_2,v_3)} \mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = 1$$
$$\sum_{C=(v_1,a_1,v_2,a_2,v_3)} \mathrm{wt}_u(C) = 4\,.$$

The first equation sums the probabilities of querying certain C's, which must sum to 1 because this is the query distribution of $\mathrm{Dec}^x(u)$. The second equation observes that $\mathrm{wt}(C)/4$ is the probability that $\mathrm{Dec}^y(u)$ queries C when y is chosen uniformly at random. As we will explain, the fact that this sums to 4 and not 1 is the critical reason why we only obtain a superpolynomial lower bound instead of an exponential one in the perfect completeness case.

Finally, we have

$$\sum_{C=(v_1,a_1,v_2,a_2,v_3)} \mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{v_3} x_u = \sum_{C=(v_1,a_1,v_2,a_2,v_3)} \mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = 1\,,$$

using the fact that the polynomial constraints are satisfied and the first of the 2 previous equations. We also note that, more generally, the left-hand side exactly computes $\mathbb{E}[\mathrm{Dec}_r^x(u) x_u]$ for a fixed x, where the expectation is over the internal randomness of the adaptive decoder.

Forming and refuting chain polynomials. The next step of the proof is to use the polynomial representation of $\mathrm{Dec}^x(u)$ above to represent the behavior of the chain decoder $\mathrm{Dec}_r^x(u)$ as a polynomial as well. Concretely, the polynomial for $\mathrm{Dec}_2^x(u)$ is

$$x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \left(\sum_{C'=(v_3,a_3,v_4,a_4,u_2)} \mathrm{wt}_{u_1}(C')\mathrm{AND}(a_3 x_{v_3}, a_3 x_{v_3}) x_{u_2} \right)\right)\,,$$

and we let $\Psi_u(x)$ denote the polynomial for the length r-chains, defined analogously. Because of perfect completeness, $\Psi_u(x) = 1$ for all $x \in \mathcal{C}$. Notice that $\Psi_u(x)$ exactly computes $\mathbb{E}[\mathrm{Dec}_r^x(u) x_u]$ where $\mathrm{Dec}_r^x(u)$ is the adaptive chain decoder and the expectation is over the internal randomness of $\mathrm{Dec}_r^x(u)$.

We then follow the strategy of [KM23] and let $\Psi_b(x) = \sum_{i=1}^k b_i \Psi_i(x)$. Standard reductions (Fact 3.4) allow us to assume that the code $\mathcal{C}$ is *systematic* with only a small loss in parameters, so that for a random $x \in \mathcal{C}$, the bits $x_1, \dots, x_k \in \{-1, 1\}$ are independent. To prove a lower bound, it suffices to then argue that $\mathbb{E}_{b \in \{-1,1\}^k}[\max_{y \in \{-1,1\}^n} \Psi_b(y)] < k$, as $\Psi_b(\mathcal{C}(b)) = k$.

The spectral refutation of [KM23], built on the CSP refutation algorithm of [GKM22], does not directly bound $\mathbb{E}_{b \in \{-1,1\}^k}[\max_{y \in \{-1,1\}^n} \Psi(y)]$, because the polynomials constructed here are quite a bit more general than the case handled in [KM23]. However, because the spectral methods of [GKM22, KM23] are sufficiently resilient, one can succeed in using the techniques to bound $\mathbb{E}_{b \in \{-1,1\}^k}[\max_{y \in \{-1,1\}^n} \Psi(y)]$ (Lemmas 5.9 and 6.6). This requires a more complicated version of the (already somewhat technical) proof in [KM23]; we will comment on the generalizations we require later.

We obtain a bound of $\mathbb{E}_{b \in \{-1,1\}^k}[\max_{y \in \{-1,1\}^n} \Psi(y)] \leq W \cdot O(\sqrt{k \ell r \log n})$ where ℓ, r are parameters with $\ell^r \gg n$, and W is the total "weight" of the Ψ_u, i.e., the sum of the weights of the coefficients. The quantity W should be considered some normalized analog of the "number of XOR constraints" present in the polynomial.

Because the sum of the weights in a single $\mathrm{Dec}(u)$ is 4, the total weight in Ψ_u is at most 4^r. This gives us a bound of $4^r \cdot O(\sqrt{k \ell r \log n})$; we then set $r = \sqrt{\log n}$, and $\ell = 2^{O(\sqrt{\log n})}$. Rearranging then implies that $k \leq 2^{O(\sqrt{\log n})}$, i.e., $n \geq k^{\Omega(\log k)}$.

The reason for the factor of 4 loss (which causes the 4^r loss and prevents us from taking $r = O(\log n)$ to get an exponential lower bound) can be observed by looking at how this reduction behaves when the code $\mathcal{C}$ is actually a linear 3-LCC. For each linear constraint $x_{v_1} x_{v_2} x_{v_3} = x_u$, we produce 4 constraints

$$\text{“}x_{v_1} = a_1 \wedge x_{v_2} = a_2 \implies x_u = a_1 a_2 x_{v_3}\,,\text{”}$$

one for every $a_1, a_2 \in \{-1,1\}^k$. So, we have produced a factor of 4 more equations than was needed. Indeed, the reason that this loss does not appear for linear codes is because of the equality

$$\sum_{a_1,a_2} \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) a_1 a_2 x_{v_3} = x_{v_1} x_{v_2} x_{v_3} .$$

However, in the adaptive case, the query v_3 can depend on the answers, and we also need not have $f_{v_1,a_1,v_2,a_2}(a_3) = a_1 a_2 a_3$, so this does not necessarily hold.

Additional technical complications. Let us now discuss the generalizations that we need of the theorems in [KM23] in order to prove Lemmas 5.9 and 6.6. First of all, we need to now handle weighted hypergraphs that are not necessarily matchings, rather than just matchings where all hyperedges have equal weight. The definitions and methods in [KM23] are sufficiently general that this can be done with some effort. A key part of the generalization is relying on the smoothness of the decoder, which conceptually generalizes the concept of hypergraph matchings to weighted hypergraphs. A second issue encountered is that $\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$ is not a homogeneous degree 2 polynomial and the decoding function $f_{v_1,a_1,v_2,a_2}(a_3)$ might have a negative sign and be $-a_3$. This means that Ψ_b is not a homogeneous degree $2r$ polynomial with nonnegative coefficients. Fortunately, this issue can again be circumvented by adding extra dummy variables y_{-v}, $y_{1^{(v)}}$, and $y_{-1^{(v)}}$ for each $v \in [n]$, where we expect these variables to take the values $y_{-v} = -x_v$, $y_{1^{(v)}} = 1$, and $y_{-1^{(v)}} - 1$.

A third and perhaps more pressing issue encountered is that the adaptive chains of $\mathrm{Dec}_r^x(u)$ are not necessarily all of length r. For example, suppose the initial invocation of $\mathrm{Dec}^x(u)$ by $\mathrm{Dec}_r^x(u)$ leads to the first 2 queries and answers being $C = (v_1, a_1, v_2, a_2)$. Then, $\mathrm{Dec}^x(u)$ generates a third query u_1. However, the decoding function $f_{v_1,a_1,v_2,a_2,u_1}(a_3)$ might not depend on $a_3 = x_{u_1}$, i.e., it could be the constant function 1 or the constant function -1. In this case, we do not have a way to continue the chain as normal, and $\mathrm{Dec}_r^x(u)$ does not make any more queries.

It turns out that the polynomials for the chains that stop early are (modulo some tricks) homogeneous degree $2t$ polynomials, where $2t$ is the number of queries, which is *even*. This should be compared to the polynomial Ψ_b, which is (modulo some tricks) a homogeneous degree $2r+1$ polynomial, which is *odd*. It turns out that the methods of [KM23], and indeed the more general case of refuting XOR instances [GKM22], are easier to analyze for even degree instances. This allows us to refute these "early stop" instances, and thereby show our key refutation lemmas: Lemmas 5.9 and 6.6.

Handling imperfect completeness. Finally, we explain how to handle the case when the decoder only succeeds with probability $1-\varepsilon$. By union bound, it follows that $\mathrm{Dec}_r^x(u)$ therefore succeeds with probability at least $1 - r\varepsilon$, and thus has $\mathbb{E}[x_u \mathrm{Dec}_r^x(u)] \geq 1 - 2r\varepsilon$. Now, we can only take $r \approx 1/2\varepsilon$ length chains while still showing a lower bound on $\mathbb{E}_{b \in \{-1,1\}^k}[\max_{y \in \{-1,1\}^n} \Psi_b(y)]$. In this parameter regime, when ε is a small constant, say, our final bound will be about $k \leq n^{1/r} \approx n^{2\varepsilon}$.

One may wonder why we cannot obtain a better lower bound on $\mathbb{E}[x_u \mathrm{Dec}_r^x(u)]$. Indeed, if $\mathbb{E}[x_u \mathrm{Dec}^x(u)]$ only depends on x, or u, but not both, then we could make the analysis work. The problem is that this is not something that we can enforce without loss of generality, as $\mathbb{E}[x_u \mathrm{Dec}^x(u)]$ could depend on both x and u.

The reason this causes an issue can be seen from the 2-chains. Consider the 2-chain polynomial from before:

$$x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \left(\sum_{C'=(v_3,a_3,v_4,a_4,u_2)} \mathrm{wt}_{u_1}(C')\mathrm{AND}(a_3 x_{v_3}, a_3 x_{v_3}) x_{u_2} \right)\right) .$$

We have

$$x_{u_1} \sum_{C'=(v_3,a_3,v_4,a_4,u_2)} \mathrm{wt}_{u_1}(C')\mathrm{AND}(a_3 x_{v_3}, a_3 x_{v_3}) x_{u_2} = \mathbb{E}[x_{u_1} \mathrm{Dec}^x(u_1)] = p_{x,u_1} \geq 1 - \varepsilon \,,$$

so that

$$\begin{aligned} &x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \left(\sum_{C'=(v_3,a_3,v_4,a_4,u_2)} \mathrm{wt}_{u_1}(C')\mathrm{AND}(a_3 x_{v_3}, a_3 x_{v_3}) x_{u_2} \right)\right) \\ &= x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{u_1} p_{x,u_1}\right) . \end{aligned}$$

Now,

$$x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{u_1}\right) = p_{x,u} \geq 1 - 2\varepsilon \,,$$

for all $x \in \mathcal{C}$, and so we would like conclude that

$$= x_u \sum_{C=(v_1,a_1,v_2,a_2,u_1)} \left(\mathrm{wt}_u(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) x_{u_1} p_{x,u_1}\right) \geq (1-2\varepsilon)^2 \,.$$

Unfortunately, the reweightings caused by the different p_{x,u_1}'s cannot be ignored because the terms in the sum are not nonnegative. In particular, it could be that $p_{x,u_1} = 1$ for the negative terms and $p_{x,u_1} = 1 - 2\varepsilon$ for the positive terms, which would then (if $p_{x,u} = 1 - 2\varepsilon$) make the sum $(1-\varepsilon)(1-2\varepsilon) - \varepsilon = (1-2\varepsilon)^2 - 2\varepsilon^2$, for example.

2.4 Roadmap

The rest of the paper is organized as follows. First, in Section 3, we introduce some notation and recall basic facts about LCCs that we shall use in the proof. In Section 4, we prove Theorem 1. In Sections 5, 6 and 8 to 10, we prove Theorem 2 in the case of perfect completeness. This proof is broken into two stages: the reduction from the adaptive smooth decoder to the chain polynomials and chain XOR instances is done in Sections 5 and 6, and in Sections 8 to 10 we refute these instances by proving Lemmas 5.9 and 6.6.

Finally, in Appendix A we prove the case of imperfect completeness in Theorem 2 and in Appendix B we recall the folklore construction of design 3-LCCs from Reed–Muller codes.

3 Preliminaries

3.1 Basic notation

We let $[n]$ denote the set $\{1, \dots, n\}$. For two subsets $S, T \subseteq [n]$, we let $S \oplus T$ denote the symmetric difference of S and T, i.e., $S \oplus T := \{i : (i \in S \wedge i \notin T) \vee (i \notin S \wedge i \in T)\}$. For a natural number $t \in \mathbb{N}$, we let $\binom{[n]}{t}$ be the collection of subsets of $[n]$ of size exactly t. Given variables $x_1, \dots, x_n$ and a subset $C \subseteq [n]$, we let $x_S := \prod_{v \in S} x_v$.

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

3.2 XOR formulas

A (weighted) XOR instance ψ on n variables $x_1, x_2, \dots, x_n$ taking values in $\{-1, 1\}$ is a collection of constraints of the form $\{x_C = b_C\}$ where $C \in \mathcal{H}$ where $\mathcal{H} \subseteq 2^{[n]}$ is the *constraint hypergraph*, along with weights $\mathrm{wt}(C) \geq 0$ for each $C \in \mathcal{H}$. The *arity* of a constraint $\{x_C = b_C\}$ equals $|C|$. The arity of ψ is the maximum arity of any constraint in it. The XOR formula associated with ψ is the expression $\psi(x) = \sum_{C \in \mathcal{H}} \mathrm{wt}(C) b_C x_C$ seen as a polynomial over $\{-1, 1\}^n$. Notice that $\psi(x) = \sum_{C \in \mathcal{H}} \mathrm{wt}(C)$ if x satisfies all the constraints of ψ and in general, evaluates to (weight of constraints satisfied by x) - (weight of constraints violated by x). The *value* $\mathrm{val}(\psi)$ of a XOR instance ψ (or, of the associated formula $\psi(x)$) is the maximum of $\psi(x)$ as x ranges over $\{-1, 1\}^n$. More generally, for a function $f(x)$, we shall let $\mathrm{val}(f) := \max_{x \in \{-1,1\}^n} f(x)$.

3.3 Locally correctable codes

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

Definition 3.1 (Locally correctable code)**.** A map $C \colon \{-1, 1\}^k \to \{-1, 1\}^n$ is a (q, δ, ε)-locally correctable code if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ that takes input an oracle access to some $y \in \{-1, 1\}^n$ and a $u \in [n]$, and has the following properties:

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

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

We say that C is *linear* if the map C, when viewed as a map from $\{0, 1\}^k \to \{0, 1\}^n$ via the mapping $0 \leftrightarrow 1$ and $1 \leftrightarrow -1$, is a linear map. We note that for linear codes, $k = \dim(\mathcal{V})$, where $\mathcal{V}$ is the image of $\{0, 1\}^k$ under the map C. We will typically let $\mathcal{L}$, as opposed to C, denote a linear code, and view $\mathcal{L}$ as a map $\mathcal{L} \colon \{0, 1\}^k \to \{0, 1\}^n$.

We say that C is systematic if for every $b \in \{-1, 1\}^k$, $C(b)|_{[k]} = b$.

For a code $C \colon \{-1, 1\}^k \to \{-1, 1\}^n$, we will write $x \in C$ to denote an $x = C(b)$ for some $b \in \{-1, 1\}^k$.

Definition 3.2 (Smooth LCCs [KT00])**.** A map $\mathcal{C}\colon \{-1,1\}^k \to \{-1,1\}^n$ is a δ-smooth q-locally correctable code with completeness $1-\varepsilon$ if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ that takes input an oracle access to some $y \in \{-1,1\}^n$ and a $u \in [n]$, and has the following properties:

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

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

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

We will call such codes (q,δ,ε)-smooth LCCs.

Remark 3.3. Any δ-smooth q-LCC with completeness $1-\varepsilon$ is a $(q,\eta\delta,\varepsilon+\eta)$-LCC for any $\eta > 0$. Indeed, this follows because if we let $y \in \{-1,1\}^n$ be a corruption of a codeword $x \in \mathcal{C}$ with $\eta\delta n$ errors, then the probability that the smooth decoder queries a corrupted entry is at most η.

Fact 3.4 (Systematic Nonlinear Codes, Lemma A.5, Thm A.6 in [BGT17])**.** *Let $\mathcal{C}\colon \{-1,1\}^k \to \{-1,1\}^n$ be a δ-smooth q-LCC with completeness $1-\varepsilon$. Then, there is a* systematic *code $\mathcal{C}'\colon \{-1,1\}^{k'} \to \{-1,1\}^n$ that is a δ-smooth q-LCC with completeness $1-\varepsilon$, where $k' = \Omega(k/\log(1/\delta))$.*

We next discuss a combinatorial characterization of *linear* locally correctable codes. To begin with, we recall basic notions about hypergraphs.

Definition 3.5. A weighted (and undirected) hypergraph $\mathcal{H}$ on vertex set $[n]$ is a weight function $\mathrm{wt}_{\mathcal{H}}\colon 2^{[n]} \to \mathbb{R}_{\geq 0}$, i.e., a function from unordered sets $C \subseteq [n]$ to $\mathbb{R}_{\geq 0}$. The hypergraph is $\leq q$-uniform if $|C| > q$ implies that $\mathrm{wt}_{\mathcal{H}}(C) = 0$ and q-uniform if $|C| \neq q$ implies that $\mathrm{wt}_{\mathcal{H}}(C) = 0$.

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

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

LCCs admit a standard combinatorial characterization (formalized in the definition below).

Definition 3.6 (Linear LCC in normal form)**.** A linear code $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ is (q,δ)-normally correctable if for each $u \in [n]$, there is a q-uniform hypergraph matching $\mathcal{H}_u$ with at least δn hyperedges such that for every $C \in \mathcal{H}_u$ and $b \in \{-1,1\}^k$, it holds that $\prod_{v \in C} x_v = x_u$ where $x = \mathcal{C}(b)$.

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

Fact 3.7 (Reduction to LCC normal form, Theorem 8.1 in [Dvi16])**.** *Let $\mathcal{L}\colon \{0,1\}^k \to \{0,1\}^n$ be a linear code that is (q,δ,ε)-locally correctable. Then, there is a linear code $\mathcal{L}'\colon \{0,1\}^k \to \{0,1\}^{2n}$ that is (q,δ')-normally correctable, with $\delta' \geq \delta/2q$.*

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

Definition 3.8 (Design 3-LCCs)**.** Let $H \subseteq \binom{[n]}{4}$ denote a collection of subsets of n of size exactly 4. We say that H is a *design* if, for every pair of vertices $u \neq v \in [n]$, there exists *exactly* one $C \in H$ with $\{u, v\} \subseteq C$.

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

Remark 3.9 (Connection between Definition 3.8 and Definition 3.6)**.** Given a design 3-LCC H, we can construct the hypergraphs H_u for $u \in [n]$ in Definition 3.8 by letting $H_u := \{C \setminus \{u\} : C \in H \text{ and } u \in C\}$ be the set of $C \in H$ that contain u (and then remove u). Because H is a design, for every pair $u \neq v \in [n]$, there exists $C \in H$ containing u and v. So, there is exactly one $C' \in H_u$ containing v, which implies that H_u is a perfect 3-uniform hypergraph matching on $[n] \setminus \{u\}$, i.e., $|H_u| = \frac{n-1}{3}$.

Finally, we recall the lower bound for linear 2-LDCs from [GKST06].

Fact 3.10 (Lemma 3.3, Lemma 3.5 in [GKST06])**.** *Let $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ be a linear map, and let $G_1, \dots, G_k$ be matchings on n vertices such that for every $b \in \{0,1\}^k$ and every $i \in [k]$ and every $(u, v) \in G_i$, it holds that $x_u + x_v = b_i$, where $x = \mathcal{L}(b)$. Suppose that $\frac{1}{k}\sum_{i=1}^k |G_i| \geq \delta n$. Then, $2\delta k \leq \log_2 n$.*

3.4 Concentration inequalities

We will use the following non-commutative Khintchine inequality [LP91].

Fact 3.11 (Rectangular Matrix Khintchine inequality, Theorem 4.1.1 of [Tro15])**.** *Let $X_1, \dots, X_k$ be fixed $d_1 \times d_2$ matrices and $b_1, \dots, b_k$ be i.i.d. from $\{-1, 1\}$. Let $\sigma^2 \geq \max(\|\sum_{i=1}^k X_i X_i^\top]\|_2, \|\sum_{i=1}^k X_i^\top X_i]\|_2)$. Then*

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

3.5 A fact about binomial coefficients

Fact 3.12. *Let n, r, t, ℓ be integers with $t \leq r$ and $\ell \geq r$. Then, it holds that*

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

Proof. First, we have that

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

We now observe that

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

which finishes the proof. □

4 Proof of Theorem 1

In this section, we prove Theorem 1. The proof is substantially simpler than the proof for general linear codes ([KM23]) or Theorem 2. The proof here will be self-contained, and will also serve as a partial warmup to Theorem 2.

The proof presented follows the overall blueprint of the proof in [KM23]. Namely, we will use the design 3-LCC $\mathcal{L}$ to construct a 2-query linear locally decodable code, and then we will apply the lower bound of [GKST06].[7] As mentioned in Section 2.2, we will incorporate the clever second moment method proof of the row pruning step due to [Yan24], which is very similar to the edge deletion method of [HKM23] done in the context of semirandom and smoothed CSP refutation [GKM22]. The key reason that we save the final $\log n$ factor is by using a more carefully chosen Kikuchi graph, a sharp accounting of binomial coefficients, and the crucial use of the fact that in the design case, the hypergraph matchings are perfect.

Let us now proceed with the proof. Let $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ be a design 3-LCC. Namely, there exists a 4-uniform hypergraph design $H \subseteq \binom{[n]}{4}$ such that for all $C \in H$, $\sum_{v \in C} x_v = 0$ for all $x \in \mathcal{L}$. Without loss of generality, we may assume that $\mathcal{L}$ is systematic, i.e., for each $b \in \{0,1\}^k$, $\mathcal{L}(b)_i = b_i$. To bound k, we will give another linear map $\mathcal{L}' \colon \{0,1\}^n \to \{0,1\}^{2nN}$, where $N = \binom{n}{\ell}$ for some parameter $\ell = (1+o(1))\log_2 n$, and we will show that $\mathcal{L}' \circ \mathcal{L} \colon \{0,1\}^k \to \{0,1\}^N$ is a 2-query linear locally decodable code with $\delta = \frac{1}{2}(1-o(1))$. We can then apply Fact 3.10 to conclude that $(1-o(1))k \leq 2\delta k \leq \log_2 N \leq (\ell+1)\log_2 n$ where $\ell = (1+o(1))\log_2 n$.

For each $u \in [n]$, we let H_u denote the 3-uniform hypergraph defined from H as specified in Remark 3.9, i.e., $H_u = \{C : C \cup \{u\} \in H\}$. As shown in Remark 3.9, H_u is a matching of size $\delta n = \frac{n-1}{3}$, i.e., $\delta := \frac{1}{3} - \frac{1}{3n}$.

Step 1: forming long chain derivations. In the first step of the proof, we use the initial system of constraints H to define a larger system of constraints, called long chain derivations. This is the key idea of [KM23] that yields the first exponential lower bound for linear 3-LCCs, and is the starting point of our proof.

[7] The proof in [KM23] is presented using the perspective of spectral refutation and matrix concentration bounds, even though the final proof eventually is a reduction to a 2-LDC. Here, we present the proof as a reduction as it is a more accessible and combinatorial analysis, although we note that one could prove the same result using matrix concentration as well.

Definition 4.1. Let $H_1, \dots, H_n$ be the 3-uniform hypergraph matchings defined from the 4-design H. An r-chain with *head* u_0 is an ordered sequence of vertices of length $3r+1$, given by $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r)$, such that all the v_h's are *distinct*[8] and for each $h = 0, \dots, r-1$, it holds that $\{v_{2h+1}, v_{2h+2}, u_{h+1}\} \in H_{u_h}$. We let $\mathcal{H}_u^{(r)}$ denote the set of r-chains with head u.

We let $C_L = (v_1, v_3, v_5, \dots, v_{2(r-1)+1})$ denote the "left half" of the chain, and $C_R = (v_2, v_4, v_6, \dots, v_{2(r-1)+2})$ denote the "right half". We call u_r the "tail".

We observe that $\mathcal{H}_u^{(r)}$ has size at most $(6\delta n)^r$ and size at least $(6\delta n - 4r)^r$. Indeed, the upper bound follows because, given a partial chain $(u_0, v_1, v_2, \dots, u_h)$, there are exactly $6\delta n$ choices of $(v_{2h+1}, v_{2h+2}, u_{h+1})$ (which we note are ordered), and the lower bound follows because there are always at least $6\delta n - 4h \geq 6\delta n - 4r$ choices, as each vertex v can appear in either the first or second spot in at most 2 *ordered* hyperedges in $H_{u'}$ for any $u' \in [n]$.

The following observation asserts that the system of linear equations given by the chains are satisfied by every $x \in \mathcal{L}$.

Observation 4.2. Let $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$ be an r-chain, with left half C_L and right half C_R. Then, for any $x \in \mathcal{L}$, it holds that $x_{u_r} + \sum_{v \in C_L} x_v + \sum_{v \in C_R} x_v = x_{u_0}$.

Proof. For any chain C, we have that for all $h = 0, \dots, r-1$, it holds that $\{v_{2h+1}, v_{2h+2}, u_{h+1}\} \in H_{u_h}$, which implies that $x_{v_{2h+1}} + x_{v_{2h+2}} + x_{u_{h+1}} = x_{u_h}$ for all $x \in \mathcal{L}$. By taking the product over all these equations, Observation 4.2 follows. □

Step 2: defining the Kikuchi graphs. In this step, we will define two linear maps $\mathcal{L}_1 \colon \{0,1\}^n \to \{0,1\}^L$ and $\mathcal{L}_2 \colon \{0,1\}^n \to \{0,1\}^R$, where $L = \binom{[n]}{\ell} \times [n]$, $R = \binom{[n]}{\ell}$, and ℓ is a parameter, as follows. Let $\mathcal{L}_1(x)_{(S,v)} := x_v + \sum_{v' \in S} x_{v'}$, and let $\mathcal{L}_2(x)_T := \sum_{v' \in T} x_{v'}$. Note that $|L| = nN$ and $|R| = N$, where $N = \binom{n}{\ell}$.

Now, for each $u \in [n]$, we will use the set of r-chains $\mathcal{H}_u^{(r)}$ to define a bipartite graph G_u with left vertices L and right vertices R such that, for each edge $((S,v),T)$ in G_u, it holds that $\mathcal{L}_1(x)_{(S,v)} + \mathcal{L}_2(x)_T = x_u$. This graph G_u will be the following Kikuchi graph.

Definition 4.3 (Kikuchi graph). Let ℓ be a parameter, to be determined later, and let G_u be the graph with left vertex set $L = \binom{[n]}{\ell} \times [n]$ and right vertex set $R = \binom{[n]}{\ell}$. For a chain $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$ with left half C_L and right half C_R, we add an edge $((S,w),T)$ to G_u "labeled" by C if $S = C_L \cup U$, $T = C_R \cup U$ where $|U| = \ell - r$[9] and $w = u_r$. Two distinct chains may produce the same edge — we add edges with multiplicity.

We now make the following simple observations about the graph G_u.

Observation 4.4. For any chain $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r) \in \mathcal{H}_u^{(r)}$, the number of edges in G_u "labeled" by C is exactly $\binom{n-2r}{\ell-r}$.

In particular, the average left degree of G_u, denoted by $d_{u,L}$ is $\binom{n-2r}{\ell-r}/nN$, and the average right degree, denoted by $d_{u,R}$ is $\binom{n-2r}{\ell-r}/N$.

[8] In this section only, we will enforce that all the v_h's are distinct, as this will be slightly more convenient.

[9] Note that here we will use that all the v_h's are distinct, so that $|C_L| = |C_R| = r$ and $|C_L| + |C_R| = 2r$.

Proof. Let C_L be the left half of C and let C_R be the right half. Because all the v_h's are distinct, we have $|C_L| = |C_R| = r$ and $|C_L \cup C_R| = 2r$. It follows that the number of pairs $((S, w), T)$ such that $((S, w), T)$ is an edge in G_u labeled by C is simply the number of choices for the set U, which is a subset of $[n] \setminus (C_L \cup C_R)$ of size $\ell - r$. Thus, there are exactly $\binom{n-2r}{\ell-r}$ choices. □

Observation 4.5. For every edge $((S, w), T)$ in G_u and $x \in \mathcal{L}$, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}_2(x)_T = x_u$.

Proof. Suppose that $((S, w), T)$ in G_u is an edge labeled by the chain C, which has left half C_L and right half C_R. We then have that $w = u_r$, $u = u_0$, and $S = C_L \cup U$, $T = C_R \cup U$. Therefore,

$$\begin{aligned}\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}_2(x)_T &= x_{u_r} + \sum_{z \in S} x_z + \sum_{z \in T} x_z \\ &= x_{u_r} + \sum_{z \in C_L} x_z + \sum_{z \in C_r} x_z + \sum_{z \in U} (x_z + x_z) = x_{u_r} + \sum_{z \in C_L} x_z + \sum_{z \in C_r} x_z = x_u \,,\end{aligned}$$

where the last equality uses Observation 4.2. □

The plan for the remainder of the proof. Let us now take a brief moment to outline the steps for the remainder of the proof. To construct a 2-LCC, it suffices to show that G_u admits a matching M_u of size $\Omega(N)$. Indeed, if this were the case, then the matching M_u would be the matching that we require to invoke Fact 3.10 and thus finish the proof.

To show that G_u has a large matching, it suffices bound the maximum degree of the graph by d, as then G_u must admit a matching of size at least $|E(G_u)|/d$. However to do this, there are two issues to resolve. The most obvious issue is that the bipartite graph is unbalanced, i.e., $|L| = n|R|$, and so this prevents us from obtaining a matching of size $\Omega(|L|)$. This issue can be easily fixed by the following trick:[10] for each right vertex $T \in R$, we can create n copies of T, denoted by $T^{(1)}, \dots, T^{(n)}$, and split the edges adjacent to T evenly across the copies. This decreases the average (and maximum) right degree by a factor of $(1 - o(1))n$, and fixes the issue.

The second, and much more challenging problem, is that the graph G_u need not be approximately biregular. Indeed, if the graph G_u was exactly *biregular*, then apply the above "splitting trick" would imply that the resulting graph has a *perfect* matching of size $nN/2$.

This irregularity issue is a common problem for Kikuchi matrices and has arisen in many prior works [GKM22, HKM23, AGKM23, KM23, Yan24]. The way to handle this issue is to show that G_u admits a subgraph G'_u that is *approximately* biregular and still contains a significant fraction of the edges of G_u, i.e., $|E(G'_u)| \geq \Omega(|E(G_u)|)$. We follow the terminology of prior work and call this step the "row pruning" step, which is so named because it involves pruning rows (and columns) of the adjacency matrix of G_u. This row pruning step is the crucial, and by far the most technical, component of the proof.

Step 3: Finding a near-perfect matching in G_u. We now argue that G_u admits a degree-bounded subgraph G'_u containing $(1 - o(1))|E(G_u)|$ edges. The strategy in [KM23] is to use the moment method to argue that with high probability, a random left (or right) vertex of the graph has

[10] This is a nice trick of [Yan24] that, while it does not affect the final bounds, saves a use of the Cauchy–Schwarz inequality and thus makes the graph G_u a bit simpler to describe.

degree at most $O(d_{u,L})$ (or $O(d_{u,R})$) with high probability. Here, we will diverge from the technical implementation of the proof in [KM23] and follow the approach of [HKM23, Yan24], which is the observation that it suffices to compute first and second moments only. Indeed, it is computing higher moments that causes the loss of several extra $\log n$ factors in the proof of [KM23], as compared to [Yan24].

The key reason we shall save the final $\log n$ factor is because the matchings H_u are nearly perfect, i.e., they have size δn where $\delta = \frac{1}{3} - \frac{1}{3n}$. This, combined with the careful choice of the matrix (see Remark 2.4) allows us to take $\ell = O(r)$ instead of $\ell = O(r^2)$, which saves a $\log n$ factor. We note that in order to get the sharp constant achieved in Theorem 1, we need to show that G_u contains a *near-perfect* matching.

Let $\deg_{u,L}(S, w)$ denote the left degree of (S, w) in G_u, and let $\deg_{u,R}(T)$ denote the right degree of T in G_u. In the following lemma, we compute the first[11] and second moments of the degree functions. This lemma is the key technical lemma of the proof, and immediately implies the existence of a degree-bounded subgraph of G_u of comparable density, as we shall shortly see.

Lemma 4.6 (Second moment bounds for the left and right degree)**.** *Let ℓ be a parameter with $\ell \geq r$ such that $r, \ell = o(n^{1/4})$. Let G_u be the graph defined in Definition 4.3. Then, it holds that*

$$\mathbb{E}_{(S,w)}[\deg_L(S,w)^2] \leq (1 + o(1) + \eta)\mathbb{E}_{(S,w)}[\deg_L(S,w)]\,,$$
$$\mathbb{E}_T[\deg_R(T)^2] \leq (1 + o(1))\mathbb{E}_T[\deg_R(T)]\,.$$

Here, the $o(1)$ is $O(\ell^2)/n$ and $\eta = n/\binom{\ell}{r}$.

We note that when we apply Lemma 4.6, we will take $r = \frac{1}{2}\log_2 n + O(\log\log n)$ and $\ell = 2r - 1$, which will end up satisfying the conditions with $\eta = 1/\mathrm{polylog}(n)$.

We postpone the proof of Lemma 4.6 to Section 4.1. Let us now use Lemma 4.6 to extract a near-perfect matching from G_u. We will assume that ℓ, r are chosen so that $\eta \leq 1/O(\log^2 n) = o(1)$, which will be the case when we choose parameters.

Using Lemma 4.6, we apply Chebyshev's inequality to observe that for the graph G_u:

1. There are at least $(1 - o(1))|L|$ left vertices with degree $d_{u,L}(1 \pm o(1))$. Let L'_u denote these left vertices.

2. There are at least $(1 - o(1))|R|$ right vertices with degree $d_{u,R}(1 \pm o(1))$. Let R'_u denote these right vertices.

Let $G'_u = G_u[L'_u, R'_u]$ be the induced subgraph. First, we observe that $|E(G'_u)| \geq (1 - o(1))|E(G_u)|$. This is because there are at least $(1 - o(1))d_{u,L}|L'_u| \geq (1 - o(1))(1 - o(1))d_{u,L}|L| \geq (1 - o(1))|E(G)|$ edges in $G[L', R]$ and at least $(1 - o(1))d_{u,R}|R'_u| \geq (1 - o(1))(1 - o(1))d_{u,R}|R| \geq (1 - o(1))|E(G)|$ edges in $G[L, R']$, and therefore $G[L', R']$ must have at least $(1 - o(1))|E(G)|$ edges. Furthermore, each left vertex in G' has degree at most $(1 + o(1))d_{u,L}$, and similarly each right vertex has degree at most $(1 + o(1))d_{u,R}$.

[11] Note that Observation 4.4 computes the first moments already.

Recall that $n \cdot d_{u,L} = d_{u,R}$ and $|L| = |R| \cdot n$. Therefore, by making n copies $T^{(1)}, \dots, T^{(n)}$ of each vertex T in R and splitting the edges equally across all copies (and doing the same induced transformation on G'_u), we can create a new bipartite graph G''_u with left vertex set L and right vertex set $R \times [n]$ where G''_u has max left (or right!) degree $(1 + o(1))d_{u,L}$ and at least $(1 - o(1))|E(G)|$ edges. Therefore, G''_u contains a matching M_u of size at least $(1 - o(1))|E(G)|d_{u,L} \geq (1 - o(1))|L|$. Note that this matching is *nearly perfect*, as the graph G''_u has $2|L|$ vertices, $|L|$ left vertices and $|L|$ right vertices.

Step 4: proving the final bound. Recall that we began with a linear map $\mathcal{L} \colon \{0,1\}^k \to \{0,1\}^n$ that is a design 3-LCC. We then built the maps $\mathcal{L}_1 \colon \{0,1\}^n \to \{0,1\}^L$ and $\mathcal{L}_2 \colon \{0,1\}^n \to \{0,1\}^R$, where $L = \binom{[n]}{\ell} \times [n]$ and $R = \binom{[n]}{\ell}$, and the matchings M_u for each $u \in [n]$ on the left vertex set L and the right vertex set $R \times [n]$. To do this, we needed to apply Lemma 4.6, which requires that $\ell, r = o(n^{1/4})$. We thus set $r = \lceil \frac{1}{2} \log_2 n + \Gamma \log_2 \log_2 n \rceil$ for a sufficiently large constant Γ and $\ell = 2r - 1$, which satisfies the conditions. We additionally have $\eta = 1/\log_2^2 n$, as

$$\binom{\ell}{r} = \binom{2r-1}{r} \geq \frac{2^{2r-1}}{2r} \geq \frac{n \cdot 2^{\Gamma \log_2 \log_2 n}}{O(\log n)} \geq n \cdot (\log_2 n)^{\Gamma - 1 - o(1)} \geq n(\log_2^2 n)\,,$$

where we use that $\binom{2r-1}{t}$ is maximized at $t = r$ and $t = r - 1$.

Let $\mathcal{L}'_2 \colon \{0,1\}^n \to \{0,1\}R \times [n]$ be the map where $\mathcal{L}'_2(x)_{T^{(h)}} = \mathcal{L}_2(x)_T$, where $T^{(h)}$ is the h-th copy of T in $R \times [n]$. A simple corollary of Observation 4.5 is that, for any $x \in \mathcal{L}$, $u \in [n]$, and edge $((S, w), T^{(h)})$ in M_u, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}'_2(x)_{T^{(h)}} = x_u$. In particular, since $\mathcal{L}$ is systematic, for any $i \in [k]$, edge $((S, w), T^{(h)})$ in M_u, and $b \in \{0,1\}^k$, it holds that $\mathcal{L}_1(x)_{(S,w)} + \mathcal{L}'_2(x)_{T^{(h)}} = x_i = b_i$.

Let $\mathcal{L}' \colon \{0,1\}^n \to \{0,1\}^{L \cup (R \times [n])} \cong \{0,1\}^{2nN}$ be the map where $\mathcal{L}'(x)_{(S,w)} = \mathcal{L}_1(x)$ and $\mathcal{L}'(x)_{T^{(h)}} = \mathcal{L}'_2(x)_{T^{(h)}}$. We have that $\mathcal{L} \circ \mathcal{L}'$ is linear map from $\{0,1\}^k \to \{0,1\}^{2nN}$ and that M_i is a matching of size $\geq (1 - o(1))nN = \frac{1}{2}(1 - o(1)) \cdot 2nN$ that decodes b_i. Therefore, by Fact 3.10, we conclude that $(1 - o(1))k \leq \log_2 N \leq (\ell + 1)(\log_2 n) = 2r \log_2 n = (1 + o(1))(\log_2 n)^2$, which proves Theorem 1.

4.1 Bounding the second moment of the left and right degrees: proof of Lemma 4.6

In this subsection, we compute upper bounds on the second moments of degree functions. This constitutes the main technical component of the proof.

As one can imagine, computing second moments requires counting the number of chains $C \in \mathcal{H}_u^{(r)}$ where the left half C_L (or right half C_R) contains a particular set Z. Because of this, we first prove the following claim.

Claim 4.7 (Ideal smoothness of chains from designs). Let H be a design 3-LCC and let $H_1, \dots, H_n$ be the 3-uniform hypergraphs defined in Remark 3.9. Let $r \geq 1$ be an integer, and let $Z \subseteq [n]$ be a subset of size t, for some $0 \leq t \leq r$. Then, the number of chains $C \in \mathcal{H}_u^{(r)}$ with $Z \subseteq C_R$ is at most $\binom{r}{t} t!(3\delta n)^{r-t} \cdot 2^r$. And, for any $w \in [n]$, the number of chains $C \in \mathcal{H}_u^{(r)}$ with tail w and $Z \subseteq C_L$ is at most $\binom{r}{t} t!(3\delta n)^{r-t-1} \cdot 2^r$ if $t \leq r - 1$ and $r! \cdot 2^r$ if $|Z| = r$.

Proof. First, let us count the number of chains $C \in \mathcal{H}_u^{(r)}$ with $Z \subseteq C_R$. We compute this in a similar way to our upper bound on $|\mathcal{H}_u^{(r)}|$. First, we pick the $\binom{r}{t}$ locations in C_R (recall that C_R is implicitly ordered by the order that the vertices appear in the chain) that will contain Z, and then we pick one of the $t!$ ways of ordering the entries of Z in these locations. Formally, we view this as fixing an ordered tuple $Q \in \{[n] \cup \star\}^r$, where the set of non-$\star$ elements of Q is equal to Z. The notation $Q_h = \star$ means that the element $v_{2(h-1)+2}$ in the chain C is "free", and $Q_h = v$ means that we must have $v_{2(h-1)+2} = v$.

Next, we count the number of chains as follows. We start with $u_0 = u$, and then we choose an ordered constraint $(v_1, v_2, u_1) \in H_{u_0}$ as follows. If $Q_1 \neq \star$, then we clearly have at most 2 choices, as we have forced $v_2 = v$ for where $v = Q_1$, which leaves at most one (unordered) $C \in H_{u_0}$ that contains v, and then we have 2 ways to order C. If this is not one of the locations where we have placed an entry of Z, i.e., $Q_1 = \star$, then we have at most $6\delta n$ choices. In total, we pay at most $\binom{r}{t} t! (6\delta n)^{r-|Z|} 2^{|Z|} = \binom{r}{t} t! (3\delta n)^{r-|Z|} 2^r$.

Now, we fix $w \in [n]$ and count the number of chains $C \in \mathcal{H}_u^{(r)}$ with tail w and $Z \subseteq C_L$. We first observe that if $|Z| = r$, then we have at most $2^r \cdot r!$ choices. Indeed, this means that $Z = C_L$, so we first pick an ordering on Z (to determine the ordering of the vertices in C_L), and then we pay a factor of 2 per step in the chain (as in the analysis in the previous paragraph). In total, there are $2^r \cdot r!$ choices.

Next, suppose that $|Z| \leq r - 1$. As before, we pay $\binom{r}{t} \cdot t!$ to determine Q, i.e., the locations and ordering of Z within the (ordered) set C_L. Let us now consider a fixed choice of the locations and ordering. We have two cases.

In the first case, suppose that $Q_r = \star$, i.e., the vertex of C_L in the "last link" (namely, $v_{2(r-1)+1}$), is not one of the locations chosen. Then, we can proceed as in the case of C_R, where we pay a factor of 2 to choose a link where v_{2h+1} is determined by Q, and a factor of $6\delta n$ on the other steps. There is one exception, which is the last step of the chain. Now, because we have also fixed the tail w, there are again only 2 choices for this step, even though $Q_r = \star$. Thus, in total, we have paid at most $2^{|Z|+1} (6\delta n)^{r-|Z|-1} = (3\delta n)^{r-|Z|} \cdot 2^r$.

In the second case, suppose that $Q_r \neq \star$, so that the vertex $v_{2(r-1)+1}$ is one of the locations chosen. Let h^* denote the index of the last $\star$ in Q, so $Q_{h^*} = \star$ and $Q_h \neq \star$ for all $h^* < h \leq r$. We now start *at the tail* of the chain and work our way backwards until we reach the h-th link in the chain. In the first step, we have already fixed the tail w and the vertex $v_{2(r-1)+1}$, and so because H is a *design*, there are at most 2 *ordered* tuples $(v, v', v_{2(r-1)+1}, w)$ where $\{v, v', v_{2(r-1)+1}, w\} \in H$, as there is one such unordered tuple and then we can swap the locations of v and v'. We continue backwards along the chain in this way until we reach the location h^*, so that $v_{2(h^*-1)+1}$ is not determined by Q since $Q_{h^*} = \star$. In particular, we have completely determined u_{h^*}, along with the all elements *after* u_{h^*} in the chain, namely $(v_{2h^*+1}, v_{2h^*+2}, \ldots, u_r)$.

Next, we proceed from the start of the chain, again paying 2 for each non-$\star$ entry and $6\delta n$ for each $\star$ entry, until we reach the h^*-th link. We have thus determined the chain up until (and including) u_{h^*-1}, i.e., $(u_0, v_1, v_2, \ldots, u_{h^*-1})$. For the final 2 vertices $(v_{2(h^*-1)+1}, v_{2(h^*-1)+2})$, we have at most 2 choices, because there is at most one hyperedge in $H_{u_{h^*-1}}$ that contains u_{h^*}, and then we have 2 ways to order the vertices. In total, we have paid $(6\delta n)^{r-|Z|-1} \cdot 2^{|Z|+1} = (3\delta n)^{r-|Z|-1} \cdot 2^r$, the same as in the other case.

In total, when $|Z| = t \leq r - 1$, we have at most $\binom{r}{t} t! (3\delta n)^{r-|Z|-1} \cdot 2^r$ choices. □

With Claim 4.7 in hand, we are almost ready to compute the second moments. To begin, we will first compute good upper bounds on the first moments $\mathbb{E}_{(S,w)}[\deg_{u,L}(S,w)]$ and $\mathbb{E}_T[\deg_{u,R}(T)]$. For the remainder of the proof, we may omit the subscript u in some places for convenience.

We have

$$\frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n-4r)^r \le d_R = \mathbb{E}_T[\deg_R(T)] \le \frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r\,,$$
$$\frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n-4r)^r \le d_L = \mathbb{E}_{(S,v)}[\deg_L(S,v)] \le \frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r\,.$$

This is because each chain C contributes $\binom{n-2r}{\ell-r}$ edges to the graph G, and we have already computed $(6\delta n-4r)^r \le |\mathcal{H}_u^{(r)}| \le (6\delta n)^r$. We also clearly have $(6\delta n-4r)^r \ge (6\delta n)^r(1-O(r^2/n))$, and so we have:

$$\left(1-\frac{O(r^2)}{n}\right)\frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r \le d_R = \mathbb{E}_T[\deg_R(T)] \le \frac{1}{\binom{n}{\ell}}\binom{n-2r}{\ell-r}(6\delta n)^r\,, \tag{1}$$
$$\left(1-\frac{O(r^2)}{n}\right)\frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r \le d_L = \mathbb{E}_{(S,v)}[\deg_L(S,v)] \le \frac{1}{n\cdot\binom{n}{\ell}}\binom{n-2r}{\ell-r}\cdot(6\delta n)^r\,. \tag{2}$$

Computing second moment of the right degree. We now compute the second moments. We will begin with $\mathbb{E}_T[\deg_R(T)^2]$, as this case is simpler. We have

$$\begin{aligned}
&\mathbb{E}_T[\deg_R(T)^2]\\
&\le \sum_{\substack{C=(C_L,C_R,w)\\ C'=(C'_L,C'_R,w')}} \Pr[C_R, C'_R \subseteq T] \;\; (T \text{ adjacent to edge labeled by } C \text{ implies } C_R \subseteq T)\\
&= \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r} \sum_{\substack{C'=(C'_L,C'_R,w')\\ |C_R\cap C'_R|=t}} \Pr[C_R, C'_R \subseteq T]\\
&= \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r} \sum_{\substack{C'=(C'_L,C'_R,w')\\ |C_R\cap C'_R|=t}} \frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}} \;\; (\text{as } C_R\cup C'_R \subseteq T \text{ and } |C_R\cup C'_R| = 2r-t)\\
&\le \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r} \binom{r}{t}\cdot\binom{r}{t} t!(3\delta n)^{r-t}\cdot 2^r\cdot\frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}} \;\; \left(\text{by Claim 4.7 and } \binom{r}{t} \text{ to pick } Z\subseteq C_R \text{ where } C_R\cap C'_R = Z\right)\\
&\le \sum_{t=0}^{r}(6\delta n)^r\binom{r}{t}\binom{r}{t} t!(3\delta n)^{r-t}\cdot 2^r\cdot\frac{\binom{n}{\ell-(2r-t)}}{\binom{n}{\ell}}\\
&\le \left(1+\frac{O(r^2)}{n}\right) d_R^2 \sum_{t=0}^{r}\binom{r}{t}\binom{r}{t} t!(3\delta n)^{-t}\frac{\binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \;\; (\text{by Eq. (1)}\,.
\end{aligned}$$

Now, we apply Fact 3.12 to conclude that

$$\mathbb{E}_T[\deg_R(T)^2] \le \left(1+\frac{O(\ell^2)}{n}\right) d_R^2 \sum_{t=0}^{r}\binom{r}{t}(3\delta n)^{-t} n^t\,\frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}}$$

$$= \left(1 + \frac{O(\ell^2)}{n}\right) d_R^2 \sum_{t=0}^{r} (3\delta)^{-t} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} .$$

Now, we observe that $\sum_{t=0}^{r} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$, as this is the probability mass function of a hypergeometric distribution, and that $3\delta = 1 - \frac{1}{n}$ (as H is a *design*), and so $(3\delta)^{-t} \leq (3\delta)^{-r} \leq \left(1 + \frac{O(r)}{n}\right)$. Thus,

$$\mathbb{E}_T[\deg_R(T)^2] \leq \left(1 + \frac{O(\ell^2)}{n}\right) d_R^2 ,$$

which gives the desired bound on the second moment.

Computing second moment of left degree. We now compute $\mathbb{E}_{(S,v)}[\deg_L(S,v)^2]$. We have

$$\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] \leq \sum_{C=(C_L,C_R,w),C'=(C'_L,C'_R,w)} \Pr[C_L, C'_L \subseteq S \wedge v = w] \text{ (both chains have same fixed tail } w)$$

$$= \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r} \sum_{\substack{C'=(C'_L,C'_R,w)\\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w]$$

$$= \left(\sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w)\\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w] \right) + \frac{\binom{n-2r}{\ell-r}}{n \cdot \binom{n}{\ell}} \cdot (6\delta n)^r \cdot r!2^r ,$$

where the last equality is because when $t = r$, then $C_L = C'_L$, and so $\Pr[C_L \subseteq S \wedge v = w] = \frac{\binom{n-2r}{\ell-r}}{n\cdot\binom{n}{\ell}}$, and by Claim 4.7, there are $r!2^r$ choices for C'.

Let us quickly handle this second term. We have by Eq. (2),

$$\frac{\binom{n-2r}{\ell-r}}{n \cdot \binom{n}{\ell}} \cdot (6\delta n)^r \cdot r!2^r \leq \left(1 + \frac{O(r^2)}{n}\right) d_L \cdot r!2^r .$$

We now compare d_L and $r!2^r$. By Eq. (2), we have

$$d_L \geq \left(1 - \frac{O(r^2)}{n}\right) \frac{\ell!}{n^{\ell+1}} \cdot \frac{(n-2r)^{\ell-r}}{(\ell-r)!} \cdot (6\delta n)^r \geq \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) (6\delta)^r \cdot \frac{1}{n} \cdot \frac{\ell!}{(\ell-r)!} .$$

Therefore,

$$\frac{d_L}{2^r r!} \geq \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) (3\delta)^r \cdot \frac{1}{n} \cdot \frac{\ell!}{(\ell-r)!r!} = \left(1 - \frac{O(r^2)}{n} - \frac{O(r\ell)}{n}\right) \left(1 - \frac{1}{n}\right)^r \cdot \frac{1}{n} \cdot \binom{\ell}{\ell-r}$$
$$= \left(1 - \frac{O(r\ell)}{n}\right) \left(1 - \frac{1}{n}\right)^r \cdot \frac{1}{n} \cdot \binom{\ell}{\ell-r} .$$

As $\binom{\ell}{\ell-r} = \eta n$ is the definition of η in Lemma 4.6, we conclude that

$$\frac{d_L}{2^r r!} \geq \eta \left(1 - \frac{O(r\ell)}{n}\right) ,$$

and so the second term is $\eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right)$.

We now return to the main calculation. We have

$$\begin{aligned}
\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] &\le \left(\sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w)\\ |C_L \cap C'_L|=t}} \Pr[C_L, C'_L \subseteq S \wedge v = w]\right) + \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \sum_{\substack{C'=(C'_L,C'_R,w)\\ |C_L \cap C'_L|=t}} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \quad \text{(as } C_L \cup C'_L \subseteq S \text{ and } |C_L \cup C'_L| = 2r - t\text{)} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{C=(C_L,C_R,w)} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! 2^r (3\delta n)^{r-t-1} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \quad \text{(by Claim 4.7 and } \binom{r}{t} \text{ to pick } Z = C_L \cap C'_L\text{)} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \sum_{t=0}^{r-1} (6\delta n)^r \binom{r}{t}\binom{r}{t} t! 2^r (3\delta n)^{r-t-1} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \frac{(6\delta n)^{2r}}{3\delta n} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! (3\delta n)^{-t} \frac{\binom{n}{\ell-(2r-t)}}{n\binom{n}{\ell}} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(r^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} \binom{r}{t}\binom{r}{t} t! (3\delta n)^{-t} \frac{\binom{n}{\ell}\binom{n}{\ell-(2r-t)}}{\binom{n-2r}{\ell-r}\binom{n-2r}{\ell-r}} \quad \text{(by Eq. (2))} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(\ell^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} \binom{r}{t} (3\delta n)^{-t} n^t \frac{\binom{\ell-r}{r-t}}{\binom{\ell}{r}} \quad \text{(by Fact 3.12)} \\
&\le \eta d_L^2 \left(1 + \frac{O(r\ell)}{n}\right) + \left(1 + \frac{O(\ell^2)}{n}\right) d_L^2 \cdot (3\delta)^{-1} \sum_{t=0}^{r-1} (3\delta)^{-t} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} .
\end{aligned}$$

Now, we have $\sum_{t=0}^{r} \frac{\binom{r}{t}\binom{\ell-r}{r-t}}{\binom{\ell}{r}} = 1$ as this is the probability mass function of a hypergeometric distribution. As $3\delta = 1 - 1/n$, it follows that $(3\delta)^{-t-1} \le (3\delta)^{-r} \le 1 + O(r/n)$, and therefore we conclude that $\mathbb{E}_{(S,v)}[\deg_L(S,v)^2] \le \left(1 + \frac{O(\ell^2)}{n} + \eta\right) d_L^2$.

5 From Adaptive Decoders to Chain Polynomials

In this section, we begin the proof of Theorem 2. We will transform a 3-LCC with an adaptive decoder into a system of satisfiable polynomial constraints that we call "chain polynomials". The polynomials will be products of AND polynomials, which we recall below.

Definition 5.1 (AND polynomial)**.** Let AND: $\{-1,1\}^2 \to \{0,1\}$ be the function where $\mathrm{AND}(\sigma,\sigma') = 1$ if $\sigma = \sigma' = 1$, and 0 otherwise. We note that $\mathrm{AND}(\sigma,\sigma') = \frac{1}{2}(1+\sigma) \cdot \frac{1}{2}(1+\sigma')$.

The key structure that we shall extract from the 3-LCC is captured by the following lemma.

Lemma 5.2. *Let $C\colon \{-1,1\}^k \to \{-1,1\}^n$ be a 3-LCC with an adaptive decoder $\mathrm{Dec}(\cdot)$. Then, for every $u \in [n]$, there are weight functions $\mathrm{wt}_{H_u}\colon [n]\times\{-1,1\}\times[n]\times\{-1,1\}\times[n] \to \mathbb{R}_{\geq 0}$ and $\mathrm{wt}_{G_u}\colon [n]\times\{-1,1\}\times[n]\times\{-1,1\} \to \mathbb{R}_{\geq 0}$ and bits $\sigma_{(u,v_1,a_1,v_2,a_2,v_3)} \in \{-1,1\}$, $\sigma_{(u,v_1,a_1,v_2,a_2)} \in \{-1,1\}$ such that for every $x \in C$,*

$$\sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\right) = 4\,, \tag{3}$$

$$\sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\right)\cdot \mathrm{AND}(a_1x_{v_1}, a_2x_{v_2}) = 1\,, \tag{4}$$

$$x_u \sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C)\sigma_{(u,C)} + \sum_{v_3\in[n]} \mathrm{wt}_{H_u}(C,v_3)\sigma_{(u,C,v_3)}x_{v_3}\right)\cdot \mathrm{AND}(a_1x_{v_1}, a_2x_{v_2}) = \mathbb{E}[\mathrm{Dec}^x(u)x_u]\,, \tag{5}$$

where the expectation $\mathbb{E}[\mathrm{Dec}^x(u)x_u]$ is over the internal randomness of the decoder. In particular, if Dec *has perfect completeness, then* $\mathbb{E}[\mathrm{Dec}^x(u)x_u] = 1$.

Furthermore, if $\mathrm{Dec}(\cdot)$ *is δ-smooth, then for any $v \in [n]$, we have*

$$\sum_{\substack{(C,v_3)=(v_1,a_1,v_2,a_2,v_3)\\ v_1=v\vee v_2=v\vee v_3=v}} \mathrm{wt}_{H_u}(C,v_3) + \sum_{\substack{C=(v_1,a_1,v_2,a_2)\\ v_1=v\vee v_2=v}} \mathrm{wt}_{G_u}(C) \leq \frac{4}{\delta n}\,.$$

We prove Lemma 5.2 in Section 5.1.

We now continue and use the above collection of polynomials to construct *polynomial chains*, a generalization of chain XOR instances defined in [KM23].

Definition 5.3 (t-chain hypergraph $\mathcal{H}_u^{(t)}$)**.** Let $t \geq 1$ be an integer. For any $u \in [n]$, let $\mathcal{H}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{H}_u^{(t)}}\colon \left([n]\times([n]\times\{-1,1\})^2\right)^t \times [n] \to \mathbb{R}_{\geq 0}$, i.e., from tuples of the form $C = (u_0, v_1, a_1, v_2, a_2, u_1, v_3, a_3, v_4, a_4, u_2, \dots, u_{t-1}, v_{2(t-1)+1}, a_{2(t-1)+1}, v_{2(t-1)+2}, a_{2(t-1)+1}, u_t)$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \prod_{h=0}^{t-1} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, a_{2h+1}, v_{2h+2}, a_{2h+2}, u_{h+1})\,.$$

For a t-chain C, we call u_0 the head, the u_h's the *pivots* for $1 \leq h \leq t-1$, and u_t the *tail* of the chain C. We call $C_L = (v_1, a_1, v_3, a_3, \dots, v_{2(t-1)+1}, a_{2(t-1)+1})$ the *left half* of the chain and $C_R = (v_2, a_2, v_4, a_4, \dots, v_{2(t-1)+2}, a_{2(t-1)+2})$ the *right half.*

The h-th link in defined to be $(u_h, v_{2h+1}, a_{2h+1}, v_{2h+2}, a_{2h+2}, u_{h+1})$.

Definition 5.4 (t-chain hypergraph $\mathcal{G}_u^{(t)}$)**.** Let $t \geq 1$ be an integer. For any $u \in [n]$, let $\mathcal{G}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{G}_u^{(t)}}\colon \left([n]\times([n]\times\{-1,1\})^2\right)^t \to \mathbb{R}_{\geq 0}$, i.e., from tuples of the form $C =$

$(u_0, v_1, a_1, v_2, a_2, u_1, v_3, a_3, v_4, a_4, u_2, \dots, u_{t-1}, v_{2(t-1)+1}, a_{2(t-1)+1}, v_{2(t-1)+2}, a_{2(t-1)+1})$[12] to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) = \mathrm{wt}_{G_{u_{t-1}}}(v_{2(t-1)+1}, a_{2(t-1)+1}, v_{2(t-1)+2}, a_{2(t-1)+2}) \cdot \prod_{h=0}^{t-2} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, a_{2h+1}, v_{2h+2}, a_{2h+2}, u_{h+1}) .$$

As before, we call $C_L = (v_1, a_1, v_3, a_3, \dots, v_{2(t-1)+1}, a_{2(t-1)+1})$ the *left half* of the chain and $C_R = (v_2, a_2, v_4, a_4, \dots, v_{2(t-1)+2}, a_{2(t-1)+2})$ the *right half*.

Note that the chains in $\mathcal{G}^{(t)}$ have no tail vertex u_t. We call the t-chain hypergraph $\mathcal{G}_u^{(t)}$ "graph-tailed", as the "last link" has 2 vertices only.

Remark 5.5 (Iterative view of the chain construction). We can view the chains as being constructed iteratively in the following way. We start with a fixed u_0. Then, we add (v_1, a_2, v_2, a_2). We now have 2 choices. We can either stop (and the chain is then in $\mathcal{G}_u^{(1)}$), or we can add u_1 to the end of the chain (and then the chain is in $\mathcal{H}_u^{(1)}$). For each chain in $\mathcal{H}_u^{(1)}$, we can then continue by adding on a "link" at u_1. On the other hand, there is no way to continue a chain in $\mathcal{G}_u^{(1)}$ in this way, as it does not contain u_1.

Definition 5.6 (Chain Polynomials)**.** Let $C = (u_0, v_1, a_1, v_2, a_2, u_1, \dots, u_t)$ be a t-chain in $\mathcal{H}_u^{(t)}$. The chain polynomial, denoted by f_C is a polynomial in the variables $x|_{C_L}, x|_{C_R}, x_{u_t}$ (where C_L and C_R are the right and left halves of the chains), defined as

$$f_C(x|_{C_L}, x|_{C_R}, x_{u_t}) = x_{u_t} \prod_{h=0}^{t-1} \mathrm{AND}(a_{2h+1} x_{v_{2h+1}}, a_{2h+2} x_{v_{2h+2}}) \sigma_{(u_{h-1}, v_{2h+1}, a_{2h+1}, v_{2h+2}, a_{2h+2}, u_h)} .$$

For a chain $C \in \mathcal{G}_u^{(t)}$, we let

$$f_C(x|_{C_L}, x|_{C_R}) = \prod_{h=0}^{t-1} \mathrm{AND}(a_{2h+1} x_{v_{2h+1}}, a_{2h+2} x_{v_{2h+2}}) \sigma_{(u_{h-1}, v_{2h+1}, a_{2h+1}, v_{2h+2}, a_{2h+2})} .$$

We are now ready to state the key facts about the chain polynomials.

Claim 5.7 (Key facts of chain polynomials). Let $\mathcal{C} \colon \{-1,1\}^k \to \{-1,1\}^n$ be a systematic 3-LCC with a (potentially adaptive) decoder. Fix $r \geq 0$, and for $1 \leq t \leq r+1$, let $\mathcal{G}_u^{(t)}, \mathcal{H}_u^{(t)}$ for $u \in [n]$ be the chains defined in Definitions 5.3 and 5.4, constructed from the polynomial system of equations in Lemma 5.2. Then, for each $u \in [n]$, the following holds:

(1) The chain polynomials correctly decode x_u. Namely, for each $x \in \mathcal{C}$, it holds that

$$x_u \left(\sum_{C \in \mathcal{H}_u^{(r+1)}} \mathrm{wt}_{\mathcal{H}_u^{(r+1)}}(C) f_C(x|_{C_L}, x|_{C_R}, x_{u_{r+1}}) + \sum_{t=1}^{r} \sum_{C \in \mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) f_C(x|_{C_L}, x|_{C_R}) \right) = 1 ,$$

[12] Note the difference with Definition 5.3: there is no final tail vertex here.

(2) The total weight of the chains of length at most $r+1$ is

$$\sum_{C \in \mathcal{H}_u^{(r+1)}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) + \sum_{t=1}^{r+1} \sum_{C \in \mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) \leq 4^{r+1}\,.$$

Proof. Let us first show the first equation. We prove this by induction on r. The base case of $r = 0$ is simple, as we have

$$\begin{aligned}
&x_u \left(\sum_{C \in \mathcal{H}_u^{(1)}} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(C) f_C(x|_{C_L}, x|_{C_R}, x_{u_1}) + \sum_{C \in \mathcal{G}_u^{(1)}} \mathrm{wt}_{\mathcal{G}_u^{(1)}}(C) f_C(x|_{C_L}, x|_{C_R}) \right) \\
&= x_u \left(\sum_{C=(v_1,a_1,v_2,a_2,v_3)} \mathrm{wt}_{H_u}(C) \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \sigma_{(u,C)} x_{v_3} + \sum_{C=(v_1,a_1,v_2,a_2)} \mathrm{wt}_{G_u}(C) \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \sigma_{(u,C)} \right) \\
&= \mathbb{E}[x_u \mathrm{Dec}^x(u)] = 1 \quad \text{(by Eq. (5))}\,.
\end{aligned}$$

We now prove the induction step. Suppose that

$$x_u \left(\sum_{C \in \mathcal{H}_u^{(r)}} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) f_C(x|_{C_L}, x|_{C_R}, x_{u_r}) + \sum_{t=1}^{r} \sum_{C \in \mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) f_C(x|_{C_L}, x|_{C_R}) \right) = 1\,.$$

We then have that for each $C \in \mathcal{H}_u^{(r)}$ with tail u_r and $\mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) > 0$,

$$f_C(x|_{C_L}, x|_{C_R}, x_{u_r}) = \left(\prod_{h=0}^{r-1} \mathrm{AND}(a_{2h+1} x_{v_{2h+1}}, a_{2h+2} x_{v_{2h+2}}) \sigma_{(u_{h-1},v_{2h+1},a_{2h+1},v_{2h+2},a_{2h+2},u_h)} \right) \cdot x_{u_r}\,,$$

and we have (via the base case) that

$$\begin{aligned}
x_{u_r} = &\sum_{C=(v_{2r+1},a_{2r+1},v_{2r+2},a_{2r+2},u_{r+1})} \mathrm{wt}_{H_u}(C) \mathrm{AND}(a_{2r+1} x_{v_{2r+1}}, a_{2r+2} x_{v_{2r+2}}) \sigma_{(u_r,C,u_{r+1})} x_{u_{r+1}} \\
&+ \sum_{C=(v_{2r+1},a_{2r+1},v_{2r+2},a_{2r+2})} \mathrm{wt}_{G_u}(C) \mathrm{AND}(a_{2r+1} x_{v_{2r+1}}, a_{2r+2} x_{v_{2r+2}}) \sigma_{(u_r,C)}\,.
\end{aligned}$$

We now simply multiply the two polynomials and sum over $C \in \mathcal{H}_u^{(r)}$ to finish the proof of Item (1).

We now turn to Item (2). We will again prove this by induction, where the base case follows from Eq. (3). To prove the induction step, we observe that

$$\sum_{C \in \mathcal{H}_u^{(r+1)}} \mathrm{wt}_{\mathcal{H}_u^{(r+1)}}(C) + \sum_{C \in \mathcal{G}_u^{(r+1)}} \mathrm{wt}_{\mathcal{G}_u^{(r+1)}}(C) = 4 \sum_{C \in \mathcal{H}_u^{(r)}} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C)\,.$$

Indeed, this follows by (1) picking a length r-chain C, (2) extending it to a length $r+1$ chain (that is either in $\mathcal{H}_u^{(r+1)}$ or $\mathcal{G}_u^{(r+1)}$), and then applying Eq. (3). We then have by the induction hypothesis

$$\sum_{C \in \mathcal{H}_u^{(r+1)}} \mathrm{wt}_{\mathcal{H}_u^{(r+1)}}(C) + \sum_{t=1}^{r+1} \sum_{C \in \mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) = 4 \sum_{C \in \mathcal{H}_u^{(r)}} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) + \sum_{t=1}^{r} \sum_{C \in \mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C)$$

$$\leq 4\left(\sum_{C\in\mathcal{H}_u^{(r)}} \mathrm{wt}_{\mathcal{H}_u^{(r)}}(C) + \sum_{t=1}^{r}\sum_{C\in\mathcal{G}_u^{(t)}} \mathrm{wt}_{\mathcal{G}_u^{(t)}}(C)\right) \leq 4^{r+1}\,,$$

which finishes the proof of Item (2). □

We are now ready to define the chain polynomial instances.

Definition 5.8 (Chain polynomial instance)**.** Let $r \geq 1$ be an integer and let $b \in \{-1,1\}^k$. For each $1 \leq t \leq r$, we define the "graph-tailed" polynomial

$$\Phi_b^{(t)}(x) = \sum_{i=1}^{k}\sum_{C\in\mathcal{G}_i^{(t)}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C)\cdot b_i f_C(x)\,,$$

and we also define the "hypergraph-tailed" polynomial

$$\Psi_b(x) = \sum_{i=1}^{k}\sum_{C\in\mathcal{H}_i^{(r+1)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)\cdot b_i f_C(x)\,.$$

We will omit the subscript b when it is clear from context. We note that in the above definitions, each f_C is the chain polynomial as defined in Definition 5.6.

With the above setup in hand, we can now state the main technical lemmas.

Lemma 5.9 (Refuting the chain polynomial instances)**.** *Let ℓ, d, r be parameters such that $d^r \geq n$, $\ell \geq 6dr/\delta$, and $\ell r = o(n)$.*[13] *Furthermore, suppose that $k \geq 1/\delta$. Then, for every $1 \leq t \leq r+1$, it holds that*

$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Phi_b^{(t)})] \leq 4^t O(\sqrt{k\ell r\log n})\,,$$

$$\mathbb{E}_{b\leftarrow\{-1,1\}^k}[\mathrm{val}(\Psi_b)] \leq 4^r\left(\frac{k(r+1)}{\delta}O(\sqrt{k\ell r\log n})\right)^{1/2}\,.$$

We now use Lemma 6.6 to finish the proof.

Proof of Theorem 2 from Lemma 6.6. By construction of the chain polynomials, i.e., Claim 5.7, we have that for every $b \in \{-1,1\}^k$ and $x = C(b)$, $\Psi_b(x) + \sum_{t=1}^{r+1}\Phi_b^{(t)}(x) = k$. This is because $b_i = x_i$ (as the code is *systematic*), and by Claim 5.7, the "b_i part" of the polynomial is equal to x_i when x is a codeword. Therefore, there must exist t such that $\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)})] \geq k/(r+2)$, or else $\mathbb{E}_b[\mathrm{val}(\Psi_b(x))] \geq k/(r+2)$.

Let us take $r = O(\sqrt{\log n})$, $d = 2^{O(\sqrt{\log n})}$, and $\ell = O(dr/\delta) = \delta^{-1}2^{O(\sqrt{\log n})}\sqrt{\log n}$. We clearly have that all the conditions of Lemma 6.6 are satisfied.

If $\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)})] \geq k/(r+2)$ for some t, then we have

$$\frac{k}{r+2} \leq \mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)})] \leq 4^t O(\sqrt{k\ell r\log n})$$

[13] Note that this is achievable by setting $\ell = 6dr/\delta$ and $r = O(\log n/\log d)$.

$$
\begin{aligned}
&\implies k \le r^2 4^{O(t)} O(\ell r \log n) \le r^2 4^{O(r)} O(\ell r \log n) \\
&\le 2^{O(\sqrt{\log n})} \cdot \frac{r^4 \log n}{\delta} \le 2^{O(\sqrt{\log n})} \\
&\implies (\log \delta k) \le O(\sqrt{\log n}) \,,
\end{aligned}
$$

or equivalently, $n \ge 2^{\Omega((\log \delta k)^2)} = (\delta k)^{\Omega(\log(\delta k))}$.

Otherwise, if $\mathbb{E}_b[\mathrm{val}(\Psi_b(x))] \ge k/(r+2)$, then we have

$$
\begin{aligned}
&\frac{k}{r+2} \le \mathbb{E}_b[\mathrm{val}(\Psi_b(x))] \le 4^r \left(\frac{k(r+1)}{\delta} O(\sqrt{k \ell r \log n})\right)^{1/2} \\
&\implies k \le \frac{r^6}{\delta^2} 2^{O(r)} O(\ell r \log n) = \frac{1}{\delta^3} 2^{O(\sqrt{\log n})} \\
&\implies \delta^3 k \le 2^{O(\sqrt{\log n})} \\
&\implies n \ge (\delta^3 k)^{\Omega(\log \delta^3 k)} \,.
\end{aligned}
$$

This finishes the proof; we note that the additional $\log(1/\delta)$-factor comes from Fact 3.4, we loses a factor of $\log(1/\delta)$ in k when one makes the code systematic. □

The remainder of paper is dedicated to proving Lemmas 5.2 and 5.9. First, we show Lemma 5.2 in Section 5.1. Then, in Section 6, we generalize the chain XOR instances of [KM23] to weighted, directed, and nonuniform hypergraphs, and we show (in Section 6.1 that refuting these "nicer" polynomials (Lemma 6.6) suffices to prove Lemma 5.9. We then break the proof of Lemma 6.6 across Sections 7 to 10, which will complete the proof of Theorem 2.

5.1 Constructing Polynomials from Adaptive Smoothed Decoders

In this subsection, we prove Lemma 5.2. Let $C \colon \{-1,1\}^k \to \{-1,1\}^n$ be a systematic 3-LCC an adaptive decoder. For each $u \in [n]$, we use the decoding algorithm $\mathrm{Dec}(u)$ to weight functions wt_{H_u} and wt_{G_u}. In what follows, we consider a fixed $u \in [n]$.

First, without loss of generality, we may assume that the decoder $\mathrm{Dec}(u)$ makes *exactly* 3 queries. We can view the decoder as a decision tree: first, $\mathrm{Dec}(u)$ generates the first query v_1 from some distribution. Then, $\mathrm{Dec}(u)$ receives a bit $a_1 \in \{-1,1\}$, the answer to the query v_1. This answer selects the branch of the decision tree, which determines the distribution of the next query v_2. Then, the decoder receives another answer $a_2 \in \{-1,1\}$, which selects the branch of the decision tree, and gives the distribution of the final query v_3. Finally, the decoder receives an answer a_3, and then it computes a (deterministic) function $f_{(v_1,a_1,v_2,a_2)}$ of a_3 to produce its output. This function must be deterministic as it must always output x_u, by perfect completeness.[14] We note that there are exactly 4 valid deterministic functions: 1, -1, a_3, and $-a_3$, so $f_{(v_1,a_1,v_2,a_2)}$ must be one of these.

For each choice of $C = (v_1, a_1, v_2, a_2, v_3) \in ([n] \times \{-1,1\})^2 \times [n]$, we let $\mathrm{wt}_u(C)$ be the probability that the decoder makes the set of queries C (with the appropriate answers) when given oracle

[14] We note that if the function is not deterministic then it is simply a convex combination of deterministic functions, and we can also handle this case. See Appendix A.

access to *any* x that is consistent with C, meaning that $x_{v_1} = a_1$ and $x_{v_2} = a_2$. Indeed, this does not depend on the choice of x, as there is some probability p_{v_1} that the decoder queries v_1 (which does not depend on x), and then given $x_{v_1} = a_1$, there is a probability p_{v_2} that the decoder queries v_2, etc.

We now partition the query sets into two types. If C is such that $f_{(v_1,a_1,v_2,a_2)}$ is a constant function $\sigma \in \{-1, 1\}$ (so it does not depend on a_3), then we set $\mathrm{wt}_{G_u}(v_1, a_1, v_2, a_2) = \mathrm{wt}_u(C)$ and $\sigma_{(v_1,a_1,v_2,a_2)} = \sigma$. Otherwise, we have that C is such that $f_{(v_1,a_1,v_2,a_2)} = \sigma a_3$, and then we set $\mathrm{wt}_{H_u}(v_1, a_1, v_2, a_2, v_3) = \mathrm{wt}_u(C)$ and $\sigma_{(v_1,a_1,v_2,a_2,v_3)} = \sigma$.

We now show that this weight function has the desired properties. Indeed, we have essentially encoded the behavior of the arbitrary decoder as this system of polynomials.

First, let us show that

$$\sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3) \right) = 4\,.$$

Consider the decoder $\mathrm{Dec}'(u)$ that simulates Dec_u by generating random bits as the answers to the queries of $\mathrm{Dec}(u)$. It follows that the probability that $\mathrm{Dec}'(u)$ queries a particular C is $\mathrm{wt}(C)/4$, and hence Eq. (3) holds.

Next, let us show that for any $x \in \mathcal{C}$

$$\sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C) + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3) \right) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = 1\,.$$

Indeed, we observe that for any $x \in \mathcal{C}$ and any C, $\mathrm{wt}_{H_u}(C, v_3) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$ is 0 if C is inconsistent with x, and otherwise it is the probability that $\mathrm{Dec}^x(u)$ queries C, and the same statement holds for $\mathrm{wt}_{G_u}(C)\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2})$. Hence, the sum must be 1.

Finally, we have

$$x_u \sum_{C=(v_1,a_1,v_2,a_2)} \left(\mathrm{wt}_{G_u}(C)\sigma_{(u,C)} + \sum_{v_3 \in [n]} \mathrm{wt}_{H_u}(C, v_3)\sigma_{(u,C,v_3)} x_{v_3} \right) \cdot \mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = \mathbb{E}[\mathrm{Dec}^x(u) x_u]\,.$$

Indeed, this is because for any $C = (v_1, a_1, v_2, a_2, v_3)$ and any $x \in \mathcal{C}$, if $\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) = 1$ then the output of the decoding function (which is $\sigma_{(u,C,v_3)} x_{v_3}$) is equal to x_u, by perfect completeness. And, a similar statement holds for $C = (v_1, a_1, v_2, a_2)$ as well. This finishes the proof.

6 Chain XOR Polynomials and the Main Technical Lemma

In this section, we will introduce an abstract notion of chains that produces a polynomial that we call a "chain XOR instance" and state a technical lemma (Lemma 6.6) that bounds the value of such instances. Then, in Section 6.1 we show that this technical lemma implies Lemma 5.9. This notion of chain XOR instances is a generalization of chain XOR derivations constructed in [KM23]. The notions here handle the case of weighted and nonuniform hypergraphs.

We begin by defining a (δ-smoothed) 3-LCC hypergraph collection.

Definition 6.1 (3-LCC hypergraph collection)**.** A 3-LCC hypergraph collection on $[n]$ vertices is a collection of pairs (H_u, G_u), one for each $u \in [n]$, where G_u is a (weighted and directed) 2-uniform hypergraph and H_u is a (weighted and directed) 3-uniform hypergraph[15] such that for every $u \in [n]$, $\sum_{C \in [n]^2} \mathrm{wt}_{G_u}(C) + \sum_{C \in [n]^3} \mathrm{wt}_{H_u}(C) = 1$.

We furthermore say that the hypergraph collection is δ-smooth if for every $u, v \in [n]$, $\sum_{C \in [n]^2 : v \in C} \mathrm{wt}_{G_u}(C) + \sum_{C \in [n]^3 : v \in C} \mathrm{wt}_{H_u}(C) \leq \frac{1}{\delta n}$

We now define the t-chain hypergraphs.

Definition 6.2 (t-chain hypergraph $\mathcal{H}_u^{(t)}$)**.** Let $t \geq 1$ be an integer, and let $(G_u, H_u)_{u \in [n]}$ denote a 3-LCC hypergraph collection. For any $u \in [n]$, let $\mathcal{H}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{H}_u^{(t)}} : [n]^{3t+1} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t+1$ tuples of the form $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2}, u_t)$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \prod_{h=0}^{t-1} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1}) \,.$$

For a t-chain C, we call u_0 the head, the u_h's the *pivots* for $1 \leq h \leq t-1$, and u_t the *tail* of the chain C. The monomial associated to C, which we denote by g_C, is defined to be $x_{u_t} \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$. We call the t-chain hypergraph $\mathcal{H}_u^{(t)}$ "hypergraph-tailed", as the last link uses one of the hypergraphs H_v.

We note that for any $u \in [n]$, $\mathcal{H}_u^{(1)}$ is equivalent to H_u, i.e., $\mathcal{H}_u^{(1)} = \{u\} \times H_u$.

Definition 6.3 (t-chain hypergraph $\mathcal{G}_u^{(t)}$)**.** Let $t \geq 1$ be an integer, and let $(G_u, H_u)_{u \in [n]}$ denote a 3-LCC hypergraph collection. For any $u \in [n]$, let $\mathcal{G}_u^{(t)}$ denote the weight function $\mathrm{wt}_{\mathcal{G}_u^{(t)}} : [n]^{3t} \to \mathbb{R}_{\geq 0}$, i.e., from length $3t$ tuples of the form $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \ldots, u_{t-1}, v_{2(t-1)+1}, v_{2(t-1)+2})$ to $\mathbb{R}_{\geq 0}$, where $\mathrm{wt}_{\mathcal{G}_u^{(t)}}(C) = 0$ if $u_0 \neq u$, and otherwise:

$$\mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) = \mathrm{wt}_{G_{u_{t-1}}}(v_{2(t-1)+1}, v_{2(t-1)+2}) \cdot \prod_{h=0}^{t-2} \mathrm{wt}_{H_{u_h}}(v_{2h+1}, v_{2h+2}, u_{h+1}) \,.$$

Note that the chains in $\mathcal{G}^{(t)}$ have no tail vertex u_t. The monomial associated to C, which we denote by x_C, is defined to be $g_C = \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}}$. We call the t-chain hypergraph $\mathcal{G}_u^{(t)}$ "graph-tailed", as the last link uses one of the graphs G_v.

We note that for any $u \in [n]$, $\mathcal{G}_u^{(1)}$ is equivalent to G_u, i.e., $\mathcal{G}_u^{(1)} = \{u\} \times G_u$.

We now make the following observation.

Observation 6.4. Let $(G_u, H_u)_{u \in [n]}$ denote a 3-LCC hypergraph collection. Then, for any $t \geq 1$ and $u \in [n]$, it holds that $\sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) + \sum_{t'=1}^{t} \sum_{C \in [n]^{3t'}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) = 1$.

[15] Note that Definition 3.5 requires that each tuple with nonzero weight has *distinct* vertices.

Proof. This follows by induction. The base case of $t = 1$ is simple, as by definition we have

$$\sum_{C \in [n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(C) + \sum_{C \in [n]^3} \mathrm{wt}_{\mathcal{G}_u^{(1)}}(C) = \sum_{(u,C) \in [n]^4} \mathrm{wt}_{\mathcal{H}_u^{(1)}}(C) + \sum_{(u,C) \in [n]^3} \mathrm{wt}_{\mathcal{G}_u^{(1)}}(C) = \sum_{C \in [n]^3} \mathrm{wt}_{H_u}(C) + \sum_{C \in [n]^2} \mathrm{wt}_{G_u}(C) \,.$$

We now show the induction step. Let $C \in [n]^{3t+1}$ have tail u_t. Let S_1 denote the set of tuples in $[n]^{3t+3}$ that extend C, i.e., the first $3t+1$ coordinates are C, and similarly let S_2 denote the set of tuples in $[n]^{3t+4}$ that extend C. We observe that $S_1 = C \times [n]^2$ and $S_2 = C \times [n]^3$. Moreover, we have

$$\sum_{C' \in S_1} \mathrm{wt}_{\mathcal{G}_u^{(t+1)}}(C') + \sum_{C' \in S_2} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C') = \sum_{C' \in [n]^2} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \mathrm{wt}_{G_{u_t}}(C') + \sum_{C' \in [n]^3} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \mathrm{wt}_{H_{u_t}}(C') = \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) \,.$$

Hence, it follows that

$$\begin{aligned}
&\sum_{C \in [n]^{3t+4}} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C) + \sum_{t'=1}^{t+1} \sum_{C \in [n]^{3t'}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) = \sum_{C \in [n]^{3t+4}} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C) + \sum_{C \in [n]^{3t+3}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) + \sum_{t'=1}^{t} \sum_{C \in [n]^{3t'}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) \\
&= \sum_{C \in [n]^{3t+4}} \mathrm{wt}_{\mathcal{H}_u^{(t+1)}}(C) + \sum_{C \in [n]^{3t+3}} \mathrm{wt}_{\mathcal{G}_u^{(t+1)}}(C) + \sum_{t'=1}^{t} \sum_{C \in [n]^{3t'}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) = \sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_u^{(t)}}(C) + \sum_{t'=1}^{t} \sum_{C \in [n]^{3t'}} \mathrm{wt}_{\mathcal{G}_u^{(t')}}(C) = 1 \,,
\end{aligned}$$

where the last step is by the induction hypothesis. □

We are now ready to define the chain XOR instances.

Definition 6.5 (Chain XOR instance)**.** Let $(G_u, H_u)_{u \in [n]}$ denote a 3-LCC hypergraph collection. Let $k \leq n$ and $r \geq 1$ be an integer. For each $1 \leq t \leq r$, we define the "graph-tailed" polynomial

$$\Phi_b^{(t)}(x) = \sum_{i \in K} \sum_{C \in [n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \cdot b_i g_C \,,$$

and we also define the "hypergraph-tailed" polynomial

$$\Psi_b(x) = \sum_{i \in K} \sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}_i^{(r)}}(C) \cdot b_i g_C \,.$$

We will omit the subscript b when it is clear from context. We note that in the above definitions, each g_C is the monomial associated with the chain C, as defined in Definitions 6.2 and 6.3.

With the above setup in hand, we can now state the main technical lemma.

Lemma 6.6 (Refuting the chain XOR instances)**.** *Let $(G_u, H_u)_{u \in [n]}$ denote a δ-smooth 3-LCC hypergraph collection and let $k \leq n$. Let ℓ, d, r be parameters such that $d^r \geq n$, $\ell \geq 6dr/\delta$, and $\ell r = o(n)$. Furthermore, suppose that $k \geq 1/\delta$. Then, for each $1 \leq t \leq r+1$, it holds that*

$$\begin{aligned}
&\mathbb{E}_{b \leftarrow \{-1,1\}^k}[\mathrm{val}(\Phi_b^{(t)})] \leq O(\sqrt{k \ell r \log n}) \,, \\
&\mathbb{E}_{b \leftarrow \{-1,1\}^k}\,[\mathrm{val}(\Psi_b)] \leq \left(\frac{k(r+1)}{\delta} O(\sqrt{k \ell r \log n}) \right)^{1/2} \,.
\end{aligned}$$

The proof of Lemma 6.6 has two steps. First, in Section 7, we refute the graph-tailed instances. Then, in Sections 8 to 10, we refute the hypergraph-tailed instances.

As we shall show in Section 6.1, Lemma 6.6 implies Lemma 5.9. For now, we devote the rest of this section to establishing some shared terminology which will be useful in the later sections.

Chains that fix some positions. We will often refer to the set of chains where some of the links, i.e., pairs (v_{2h+1}, v_{2h+2}) are forced to contain some $v \in [n]$. Towards this, we introduce the following terminology.

Definition 6.7 (Chains containing Q). Let t, r be integers with $t \leq r$. For any $Q = (Q_1, \dots, Q_t, Q_{t+1}) \in \{[n] \cup \star\}^{t+1}$, we say that a length $3r+1$ tuple $C = (u_0, v_1, v_2, u_1, v_3, v_4, u_2, \dots, u_{t-1}, v_{2(r-1)+1}, v_{2(r-1)+2}, u_r)$ contains Q, denoted by $Q \subseteq C$, if $Q_{t+1} \in \{\star, u_r\}$ and for $1 \leq h \leq t$, if $Q_h \neq \star$, then either $Q_h = v_{2(r-1-t+h)+1}$ or $Q_h = v_{2(r-1-t+h)+2}$.

We say that a Q is *contiguous* if there exists $s \leq t$ such that $Q_h \neq \star$ for every $h \geq s+1$ and $Q_h = \star$ for every $1 \leq h \leq s$, i.e., the first s entries are $\star$, and the remaining entries are non-$\star$. We note that by definition, $Q_{t+1} \neq \star$ always.

We say that Q is *complete* if Q does not contain any $\star$. We say that $Q' \supseteq Q$ if whenever $Q_h \neq \star$, $Q'_h = Q_h$. We define the size $|Q|$ to be the number of coordinates in Q that do not equal $\star$.

6.1 Relating the chain polynomials and chain XOR instances

Recall that the chain polynomials $\Phi_b^{(t)}$ and Ψ_b in Section 5 are products of AND functions, which means that (1) they are inhomogeneous polynomials, and (2) some of the coefficients can be negative. This is contrast to the chain XOR instances produced in Section 6, which are homogeneous and with positive coefficients, and this is very helpful in the proof of Lemma 6.6.

The goal of this section is to show that, given the output of Lemma 5.2, we can construct a 3-LCC hypergraph collection $(H'_u, G'_u)_{u \in [n]}$ such that the chain XOR instances $\Phi'^{(t)}_b$ and Ψ'_b produced from (H'_u, G'_u) are (up to a scaling factor) equivalent to the chain polynomial instances $\Phi_b^{(t)}$ and Ψ_b from Section 5.

First, we explain how to convert the polynomials $\Phi_b^{(t)}$ and Ψ_b into equivalent homogeneous polynomials $\tilde{\Phi}_b^{(t)}$ and $\tilde{\Psi}_b$ over a larger set of $4n$ variables. In particular, these new polynomials will have the following properties (1) $\mathrm{val}(\tilde{\Phi}_b^{(t)}) \geq \mathrm{val}(\Phi_b^{(t)})$ and $\mathrm{val}(\tilde{\Psi}_b) \geq \mathrm{val}(\Psi_b)$, (2) $\tilde{\Psi}_b$ is a degree $2(r+1)+1$ homogeneous polynomial and $\tilde{\Phi}_b^{(t)}$ is a degree $2t$ homogeneous polynomial. Then, we will construct a 3-LCC hypergraph collection $(H'_u, G'_u)_{u \in [4n]}$, and show that the chain XOR instances $\Phi'^{(t)}_b$ and Ψ'_b produced from this collection are equal to $4^{-t}\Phi_b^{(t)}$ and $4^{-r}\tilde{\Psi}_b$.

Defining the homogeneous polynomials. This transformation to produce $\Phi_b^{(t)}$ and $\tilde{\Psi}_b$ is straightforward. First, we define a map $\pi \colon \{-1,1\}^n$ to $\{-1,1\}^{4n}$ as follows. For each $x \in \{-1,1\}^n$ we define $y = \pi(x)$ by adding, for each $v \in [n]$, the 4 bits x_v, $-x_v$, 1 and -1 to y. We refer to these bits as $+v, -v, 1^{(v)}, -1^{(v)}$, i.e., $y_{+v} = x_v$, $y_{-v} = -x_v$, $y_{1^{(v)}} = 1$, and $y_{-1^{(v)}} = -1$. We think of $1^{(v)}$ as the v-th copy of 1, and similarly $-1^{(v)}$ is the v-th copy of -1.

Now, we transform the polynomials $\Phi_b^{(t)}$ and Ψ_b. Each term that contains a function $\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \cdot$

σ for $\sigma \in \{-1, 1\}$ is replaced by the 8 terms

$$\frac{1}{8}y_{\sigma^{(v_1)}}y_{1^{(v_2)}} + \frac{1}{8}y_{\sigma a_1 v_1}y_{1^{(v_2)}} + \frac{1}{8}y_{\sigma^{(v_1)}}y_{a_2 v_2} + \frac{1}{4}y_{\sigma a_1 v_1}y_{a_2 v_2} + \frac{1}{8}y_{1^{(v_1)}}y_{\sigma^{(v_2)}} + \frac{1}{8}y_{a_1 v_1}y_{\sigma^{(v_2)}} + \frac{1}{8}y_{1^{(v_1)}}y_{\sigma a_2 v_2} + \frac{1}{8}y_{a_1 v_1}y_{\sigma a_2 v_2}\,,$$

where, e.g., $y_{\sigma a_1 v_1}$ is y_{+v} if $\sigma a_1 = 1$ and y_{-v} if $a_1 = -1$, and $y_{\sigma^{(v)}}$ is either $y_{1^{(v)}}$ if $\sigma = 1$ or $y_{-1^{(v)}}$ if $\sigma = -1$. By construction, if $y = \pi(x)$, then $\mathrm{AND}(a_1 x_{v_1}, a_2 x_{v_2}) \cdot \sigma$ is equal to this new polynomial, and this polynomial is a homogeneous degree 2 polynomial in y with nonnegative coefficients. Furthermore, the coefficients of the new polynomial all sum to 1.

Defining the homogeneous polynomials via XOR chains on hypergraphs. In order to refute the homogeneous polynomials using Lemma 6.6, we will need to write them as "chain XOR instances" on a certain set of hypergraphs. Now, because the coefficients of the new AND polynomials all sum to 1, we can essentially replace each hyperedge $C = (v_1, a_1, v_2, a_2, v_3)$, e.g., with 8 new hyperedges each of weight $1/8\mathrm{wt}(C)$. This defines, for each $u \in [n]$, a pair (H'_u, G'_u) of weighted 3-uniform and 2-uniform hypergraphs.

We can then form the chain XOR instances $\Phi'^{(t)}_b$ and Ψ'_b from this hypergraph collection. It will be fairly immediate to observe that the resulting polynomials produced via this process are the same as $\tilde{\Phi}^{(t)}_b$ and $\tilde{\Psi}_b$ up to a scaling factor – in other words, the operations of "form chains" and "add extra variables" commute. As a result, the "chain XOR instances" $\Phi'^{(t)}_b$ and Ψ'_b we get from this process are equal to the polynomials $4^{-t}\tilde{\Phi}^{(t)}_b$ and $4^{-r}\tilde{\Psi}_b$ that we have just defined, and thus we can refute them using Lemma 6.6, which will imply Lemma 5.9.

In the remainder of this section, we define the pairs (H'_u, G'_u) of weighted 3- and 2-uniform hypergraphs. Then, we will finally observe that these XOR instances are equivalent to the polynomials $\tilde{\Phi}^{(t)}_b$ and $\tilde{\Psi}_b$.

Definition 6.8 (The pairs (H'_u, G'_u))**.** Let $u \in [n]$ and let $n' := 4n$. We identify $4n$ with the $4n$ vertices $+v, -v, 1^{(v)}$, and $-1^{(v)}$.

We define the weight function $\mathrm{wt}_{H'_u}$ and $\mathrm{wt}_{G'_u}$ as follows. For each $C = (v_1, a_1, v_2, a_2, v_3)$ with bit $\sigma_C \in \{-1, 1\}$, we set

(1) $\mathrm{wt}_{H'_u}(\sigma_C a_1 v_1, a_2 v_2) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(2) $\mathrm{wt}_{H'_u}(\sigma_C^{(v_1)}, a_2 v_2) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(3) $\mathrm{wt}_{H'_u}(\sigma_C a_1 v_1, 1^{(v_2)}) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(4) $\mathrm{wt}_{H'_u}(\sigma_C^{(v_1)}, 1^{(v_2)}) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(5) $\mathrm{wt}_{H'_u}(a_1 v_1, \sigma_C a_2 v_2) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(6) $\mathrm{wt}_{H'_u}(1^{(v_1)}, \sigma_C a_2 v_2) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(7) $\mathrm{wt}_{H'_u}(a_1 v_1, \sigma_C^{(v_2)}) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

(8) $\mathrm{wt}_{H'_u}(1^{(v_1)}, \sigma_C^{(v_2)}) = \frac{1}{8}\mathrm{wt}_{H_u}(C) \cdot \frac{1}{4}$,

and we do the analogous transformation to define $\mathrm{wt}_{G_u'}$ from wt_{G_u}.

Furthermore, if the original decoder $\mathrm{Dec}(\cdot)$ was δ-smooth, then for any new vertex $v' \in [n']$, it holds that

$$\sum_{C'=(v_1',v_2',v_3'):v'\in C'} \mathrm{wt}_{H_u'}(C') + \sum_{C'=(v_1',v_2'):v'\in C'} \mathrm{wt}_{G_u'}(C') \leq \frac{4}{\delta n'}\,.$$

Remark 6.9. So far, we have only defined a pair (H_u', G_u') for the original vertices u, not the new vertices $u' \in [n']$, so technically we have not defined a full hypergraph collection. However, we can easily define equivalent hypergraphs for all the new vertices, but this turns out to be unnecessary as the only hyperedges in H_u' with nonzero weight have $C' = (v_1', v_2', v_3)$ where $v_3 \in [n]$ is one of the original vertices, and so the chain XOR instances formed will never use the hypergraphs $(H_{u'}', G_{u'}')$ if u' is a "new vertex". So, we do not need to define $H_{u'}'$ and $G_{u'}'$ where u' is a new variable. This is also the reason for the upper bound of $\frac{16}{\delta n'}$ instead of $\frac{4}{\delta n'}$ – a fixed third vertex $v_3 \in [n]$ could have $\frac{4}{\delta n}$-fraction of the weight in the original decoder, which is now $\frac{16}{\delta n'} \cdot \frac{1}{4}$ (as we scale down all weights by $1/4$).

This now leads us to the following key observation.

Observation 6.10. Let $\tilde{\Phi}_b^{(t)}$ and $\tilde{\Psi}_b$ be the polynomials defined via the transformation to $\Phi_b^{(t)}$ and Ψ_b, and let $\Phi'^{(t)}_b$ and Ψ'_b be the chain XOR instances of the 3-LCC hypergraph collection $(H_u', G_u')_{u\in[n]}$. Then, $\tilde{\Phi}_b^{(t)} = 4^{-t}\Phi'^{(t)}_b$ and $\tilde{\Psi}_b = 4^{-r}\Psi_b'$.

This observation follows immediately from the definitions. Namely, for each $C = (v_1, a_1, v_2, a_2, v_3)$ in the original H_u, we have now added 8 different constraints of weight $1/32$ times the original weight of C, such that the XOR instance on the new constraints is equal to the AND polynomial of the previous constraints.

The above observation immediately shows that Lemma 6.6 implies Lemma 5.9, and so it remains to prove Lemma 6.6.

7 Refuting the Graph-Tail Instances

In this section, we prove the first equation of Lemma 6.6. Let $r \geq 1$ and let $1 \leq t \leq r+1$ be fixed. We begin by defining the Kikuchi matrices.

Definition 7.1. Let $r \geq 1$ and $1 \leq t \leq r+1$. Let $i \in [k]$. For a tuple $C = (i, v_1, v_2, u_1, v_3, v_4, \dots, v_{2(t-1)+1}, v_{2(t-1)+2}) \in [n]^{3t}$, we define the matrix $A_i^{(C)} \in \{0,1\}^N$ where $N = \binom{n}{\ell}^t$, to be the matrix indexed by tuples of sets $\vec{S} = (S_0, \dots, S_{t-1})$, where $A_i^{(C)}((S_0,\dots,S_{t-1}),(T_0,\dots,T_{t-1})) = 1$ if for all $h = 0,\dots,t-1$, $S_h \oplus T_h = \{v_{2h+1}, v_{2h+2}\}$ with $v_{2h+1} \in S_h, v_{2h+2} \in T_h$. If this does not hold, then the entry of the matrix is 0.

We let $A_i = \frac{1}{D_t}\sum_{C\in[n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) A_i^{(C)}$ and $A = \sum_{i=1}^k b_i A_i$. Here, $D_t = \binom{n-2}{\ell-1}^t$.

Next, we relate $\Phi^{(t)}(x)$ to a quadratic form on the matrix A.

Lemma 7.2. *Let $x \in \{-1,1\}^n$, and let $x' \in \{-1,1\}^N$, where $N = \binom{n}{\ell}^t$, denote the vector where the $(S_0, S_1, \dots, S_{t-1})$-th entry of x' is $\prod_{h=0}^{t-1} x_{S_h}$. Let $i \in [k]$ and $t \in \{0, \dots, r\}$. Then, for any $C = (i, v_1, v_2, u_1, v_3, v_4, \dots, v_{2(t-1)+1}, v_{2(t-1)+2}) \in [n]^{3t}$, it holds that*

$$x'^{\top} A_i^{(C)} x' = D_t \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}} ,$$

i.e., the product of the monomials associated to C, where $D_t = \binom{n-2}{\ell-1}^t$. Moreover, for any matrix $B_i^{(C)}$ obtained by "zeroing out" exactly αD_t entries of $A_i^{(C)}$, the equality holds with a factor of $1-\alpha$ on the right.

In particular, $x'^{\top} A x' = \Phi^{(t)}(x)$.

Proof. Let $\vec{S} = (S_0, S_1, \dots, S_{t-1})$ and $\vec{T} = (T_0, \dots, T_{t-1})$ be such that $A_i^{(C)}(\vec{S}, \vec{T}) = 1$. Then, we have that

$$x'_{\vec{S}} x'_{\vec{T}} = \prod_{h=0}^{t-1} x_{S_h} x_{T_h} = \prod_{h=0}^{t-1} x_{S_h \oplus T_h} = \prod_{h=0}^{t-1} x_{v_{2h+1}} x_{v_{2h+2}} \;,$$

which is equal to the product of monomials on the right-hand side of the equation we wish to show.

It thus remains to argue that $A_i^{(C)}$ has exactly D_t nonzero entries. We observe that, for each $h = 0, \dots, t-1$, there are exactly $\binom{n-2}{\ell-1}$ pairs (S_h, T_h) such that $S_h \oplus T_h = C_h$ with $v_{2h+1} \in S_h$ and $v_{2h+2} \in T_h$. Indeed, this is because by Definition 3.5, these vertices must be distinct, and then we must simply choose a set of size $\ell - 1$ that does not contain either of v_{2h+1} and v_{2h+2} and this determines S_h and T_h. Thus, $D_t = \binom{n-2}{\ell-1}^t$, as required. □

We would like to now apply matrix Khintchine (Fact 3.11) to bound $\mathbb{E}_b[\|A\|_2]$ and thus bound $\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)}(x))]$. However, to do this, we need good bounds on the $\|A_i\|_2$ of the individual matrices A_i. It turns out that the bounds we require for this approach to work are false, but one can find a submatrix B_i of A_i such that the bounds hold. To argue this, we will need the following first moment bounds.

Lemma 7.3 (First and conditional moment bounds)**.** *Fix $r \geq 1$, $1 \leq t \leq r+1$, and $i \in [k]$. Let A_i be the Kikuchi matrix defined in Definition 7.1.*

Let $\vec{S} = (S_0, \dots, S_{t-1}) \in \binom{[n]}{\ell}^t$ be a row of the matrix, and let $\deg_i(\vec{S})$ denote the ℓ_1-norm of the $\vec{S}$-th row of A_i. Then,

$$\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})] \leq \frac{1}{N} ,$$

where $N = \binom{n}{\ell}^t$.

Furthermore, let $C \in [n]^{3t}$ be a chain with head i. Let $\mathcal{D}_C$ denote the uniform distribution over rows of $A_i^{(C)}$ that contain a nonzero entry. Then, it holds that

$$\mathbb{E}_{\vec{S} \sim \mathcal{D}_C}[\deg_i(\vec{S})] \leq \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{4}{N} .$$

Finally, the same bounds hold for the columns of the matrix.

With Lemma 7.3, we can now do the following. Let Γ be a sufficiently large constant, let $\mathcal{B}_1 = \{\vec{S} : \deg_i(\vec{S}) \geq \Gamma/N\}$ be the set of rows with ℓ_1-norm at least Γ/N, and similarly let $\mathcal{B}_2$ be defined for the columns. We observe that by the conditional moment bounds in Lemma 7.3 and Markov's inequality, each $A_i^{(C)}$ has at least $1 - O(1/\Gamma)$-fraction of its nonzero rows not in $\mathcal{B}_1$, and similarly for columns and $\mathcal{B}_2$. It thus follows that after setting all the rows in $\mathcal{B}_1$ and columns in $\mathcal{B}_2$ to 0, the resulting matrix still has at least $1 - O(1/\Gamma)$-fraction of its original nonzero entries. By taking Γ large enough, we can ensure that this fraction is at least 1/2. Now, we let $B_i^{(C)}$ be the matrix where we have deleted all rows in $\mathcal{B}_1$ and columns in $\mathcal{B}_2$ from $A_i^{(C)}$, and we have additionally set more entries to 0 so that $B_i^{(C)}$ has *exactly* $D_t/2$ nonzero entries, where t is such that $C \in [n]^{3t}$.

Let us define: $B_i = \frac{1}{D_t}\sum_{C\in[n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) B_i^{(C)}$ and $B = \sum_{i=1}^k b_i B_i$. By Lemma 7.2 (and the "moreover" part), we have that for every $x \in \{-1,1\}^n$, there exists $x' \in \{-1,1\}^N$ such that $x'^\top B x' = \frac{1}{2}\Phi^{(t)}(x)$. By construction, we have that $\|B_i\|_2 \leq \Gamma/N$, as this is an upper bound on the ℓ_1-norm of any row/column in B_i.

Thus, applying matrix Khintchine (Fact 3.11), we obtain

$$\mathbb{E}_b[\mathrm{val}(\Phi_b^{(t)})] \leq \mathbb{E}_b[N\|B\|_2] \leq N \cdot \frac{\Gamma}{N} O(\sqrt{k \log N}) = O(\sqrt{k\ell r \log n}),$$

where we use that Γ is constant. This finishes the proof of the first equation in Lemma 6.6, up to the proof of Lemma 7.3.

Proof of Lemma 7.3. We will only prove the statement for the rows. One can observe from the proof that it will immediately hold for the columns also.

We begin by estimating the first moment, i.e., $\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})]$. By definition, we have that

$$\mathbb{E}_{\vec{S}}[\deg_i(\vec{S})] = \frac{1}{N}\frac{1}{D_t}\sum_{C\in[n]^{3t}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \cdot D_t \leq \frac{1}{N},$$

as the sum of the weights of all chains is at most 1.

We now fix $t \in \{1, \dots, r+1\}$, $C \in [n]^{3t}$ with head i. Let $\mathcal{D}_C$ denote the uniform distribution over rows of $A_i^{(C)}$ that contain a nonzero entry. We compute the conditional expectation as follows. First, we shall bound, for $C' \in [n]^{3t}$ with head i, the number of rows $\vec{S}$ such that $A_i^{(C)}$ and $A_i^{(C')}$ both have a nonzero entry in the $\vec{S}$-th row, *normalized* by the scaling factor $1/D_t$. This quantity will depend on some parameter z, which is the number of "shared vertices" between C and C'. Then, we will bound, for each z, the total weight of all $C' \in [n]^{3t}$ that has at least z "shared vertices" with C.

Step 1: bounding the normalized number of entries for a fixed C'. To begin, we define the number of "shared vertices" between two pairs of chains C and C'.

Definition 7.4 (Left vertices)**.** Let $C \in [n]^{3t}$. The tuple of *left vertices* of C is the sequence $L(C) = (v_1, v_3, v_5, \dots, v_{2(t-1)+1})$. We note that if $\vec{S}$ is a row such that $A_i^{(C)}$ has nonzero entry in the $\vec{S}$-th row, then $v_{2h+1} \in S_h$ for $h = 0, \dots, t-1$.

Definition 7.5 (Intersection patterns)**.** Let $C \in [n]^{3t}$ and $C' \in [n]^{3t}$.

The *intersection pattern* of C with C', given by $Z \in \{0,1\}^t$, is defined as $Z_h = 1$ if $L(C)_h = L(C')_h$, and it is 0 otherwise.

We now fix $C' \in [n]^{3t}$ and count the number of rows as a function of the intersection pattern Z. We observe that in order for a row $\vec{S}$ to have a nonzero entry for both pairs of chains, we must have $\{L(C)_h, L(C')_h\} \subseteq S_{h-1}$ for all $h = 1, \dots, t$.

We observe that for each intersection point, i.e., an h such that $L(C)_h = L(C')_h$, there are $\binom{n}{\ell-1}$ choices for the corresponding set, as it needs to only contain one vertex. For each nonintersection point, i.e., an $h \in \{1, \dots, t\}$ where $L(C)_h \neq L(C')_h$, we have $\binom{n}{\ell-2}$ choices, because the set needs to contain both vertices. In total, we have $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}$.

Now, this implies an upper bound of $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z} / D_t$ on the normalized number of entries, which we can compute as

$$\begin{aligned}
\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z} / D_t &= \frac{\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{t-z}}{\binom{n-2}{\ell-1}^t} = 2\left(\frac{\binom{n}{\ell-2}}{\binom{n}{\ell-1}}\right)^{t-z} \cdot \left(\frac{\binom{n}{\ell-1}}{\binom{n-2}{\ell-1}}\right)^t \\
&\leq \left(\frac{\ell-1}{n-\ell+2}\right)^{t-z} \cdot \left(\frac{n(n-1)}{(n-\ell+1)(n-\ell)}\right)^t \leq \left(\frac{\ell}{n}\right)^{t-z} \cdot \left(1 + \frac{O(\ell r)}{n}\right) .
\end{aligned}$$

Step 2: bounding the weight of C' with a fixed intersection pattern Z. Let us fix the intersection pattern Z. We observe that this determines a set of $|Z|$ vertices that must be contained in C'. We will abuse notation and let $Z \in [n] \cup \{\star\}^t$ denote this sequence of vertices (with $\star$'s for the unfixed entries). Let t'' denote the largest $h \in \{1, \dots, t\}$ for which $Z_{t''} \neq \star$. We then have

$$\begin{aligned}
&\sum_{C' \in [n]^{3t} : Z \subseteq C} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\mathrm{wt}_{\mathcal{G}_i^{(t'')}}(C'') + \sum_{C' \in [n]^{3(t-t'')}} \mathrm{wt}_{\mathcal{G}_i^{(t)}}(C'', C') \right) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\mathrm{wt}_{\mathcal{G}_i^{(t'')}}(C'') + \sum_{(u, C') \in [n]^{3(t-t'')}} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \right) \\
&= \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\mathrm{wt}_{\mathcal{G}_i^{(t'')}}(C'') + \sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \sum_{C' \in [n]^{3(t-t'')-1}} \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \right) \\
&\leq \sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\mathrm{wt}_{\mathcal{G}_i^{(t'')}}(C'') + \sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \right) .
\end{aligned}$$

Above, we use that $\sum_{C' \in [n]^{3(t-t'')-1}} \mathrm{wt}_{\mathcal{G}_u^{(t-t'')}}(u, C') \leq 1$, which follows by Observation 6.4.

We now clearly have that $\sum_{C'' \in [n]^{3t''} : Z \subseteq C''} \left(\mathrm{wt}_{\mathcal{G}_i^{(t'')}}(C'') + \sum_{u \in [n]} \mathrm{wt}_{\mathcal{H}_i^{(t'')}}(C'', u) \right) \leq (\delta n)^{-|Z|}$. This follows by δ-smoothness, as when we sum over a link with no fixed vertex, it has weight 1, and when we sum over a link where $Z_h \neq \star$, by δ-smoothness it must have weight at most $1/\delta n$. We thus have a bound of $(\delta n)^{-|Z|}$.

Putting it all together. By combining steps (1) and (2) (and paying an additional $\binom{t}{z}$ factor to choose the nonzero entries of Z), we thus obtain the final bound of

$$\begin{aligned}
&\mathbb{E}_{\vec{S}\sim\mathcal{D}_C}[\deg_i(\vec{S})] \le \frac{1}{D_t}\sum_{z=0}^{t}\binom{t}{z}\cdot 2\left(1+\frac{O(\ell r)}{n}\right)\cdot\left(\frac{\ell}{n}\right)^{t-z}\cdot(\delta n)^{-z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t}\left(\frac{\ell}{n}\right)^{t}\cdot\sum_{z=0}^{t}\cdot\left(\frac{t}{\delta\ell}\right)^{z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t}\left(\frac{\ell}{n}\right)^{t}\cdot\sum_{z=0}^{r}\cdot\left(\frac{r}{\delta\ell}\right)^{z} \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{D_t}\left(\frac{\ell}{n}\right)^{t},
\end{aligned}$$

where we use that $\ell \ge 2r/\delta$.

Finally, we need to compute D_t/N. We have

$$\begin{aligned}
\frac{D_t}{N} &= \frac{\binom{n-2}{\ell-1}^t\cdot\binom{n}{\ell}^{r+1-t}}{\binom{n}{\ell}^{r+1}} = \left(\frac{\binom{n-2}{\ell-1}}{\binom{n}{\ell}}\right)^t \\
&\left(\frac{\ell(n-\ell)}{n(n-1)}\right)^t \ge \left(\frac{\ell}{n}\right)^t\left(1-\frac{O(\ell r)}{n}\right).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\mathbb{E}_{\vec{S}\sim\mathcal{D}_C}[\deg_i(\vec{S})] \le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{D_t}\left(\frac{\ell}{n}\right)^t \\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{N},
\end{aligned}$$

which finishes the proof. □

8 Smooth Partitions of Chains

In this section, we begin the proof of the second equation in Lemma 6.6.

For notation, we let $\mathcal{H}^{(t)}$ be the union, over u, of $\mathcal{H}_u^{(t)}$, and $\mathrm{wt}_{\mathcal{H}^{(t)}}(\cdot) = \sum_{u\in[n]}\mathrm{wt}_{\mathcal{H}_u^{(t)}}(\cdot)$.

Lemma 8.1. *Let $t \ge 1$ and $d \ge 1$ be integers. There is a subset $P_t \subseteq [n]^{t+1}$ and disjoint sets $\mathcal{T}^{(Q)} \subseteq [n]^{3t+1}$ for $Q \in P_t$ such that (1) $Q \subseteq C$ for each $C \in \mathcal{T}^{(Q)}$, and (2) $\mathrm{wt}(Q) := \sum_{C\in\mathcal{T}^{(Q)}}\mathrm{wt}_{\mathcal{H}^{(t)}}(C) \ge nd^t\cdot(\delta n)^{-t-1}$.*

We say Q is heavy *if $Q \in P_t$. Note that if Q is heavy then Q is contiguous and complete by definition.*

Finally, as a trivial case, we let $P_0 = [n]$ and for $Q = (v) \in P_0$, we let $\mathcal{T}^{(Q)} = (v)$. Here, we let $\mathrm{wt}(Q) = 1$.

Proof. The proof follows by a simple greedy algorithm. Let $S = [n]^{3t+1}$. If there exists Q such that $\sum_{C \in S : Q \subseteq C} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) \geq nd^t \cdot (\delta n)^{-t-1}$, then we remove all such C from S and add them to $\mathcal{T}^{(Q)}$. We repeat until there is no such Q remaining. We note that Q cannot be used twice in this sequence, as when we pick a Q we remove all $C \in S$ containing Q. □

Definition 8.2 (Partitions of the chains)**.** Let $r \geq 1$ be an integer. For each $1 \leq t \leq r$ and heavy $Q \in P_t$, we let $\mathcal{H}^{(r,Q)}$ denote the set of tuples $C \in [n]^{3r+1}$ where:

1. C is extends a tuple in $\mathcal{T}^{(Q)}$ "backwards", i.e., $(C_{3(r-t)+1}, \dots, C_{3r+1}) \in \mathcal{T}^{(Q)}$;

2. Q is maximal: for any $t' > t$ and $Q' \in P_{t'}$, $(C_{3(r-t')+1}, \dots, C_{3r+1}) \notin \mathcal{T}^{(Q')}$.

Observation 8.3. We have that for each $t = 0, \dots, r$, it holds that $\sum_{Q \in P_t} \mathrm{wt}(Q) \leq n$, and so $\sum_{t=0}^{r} \sum_{Q \in P_t} \mathrm{wt}(Q) \leq (r+1)n$.

Proof. We observe that for any $t = 0, \dots, r$, it holds that

$$\sum_{Q \in P_t} \mathrm{wt}(Q) = \sum_{Q \in P_t} \sum_{C \in \mathcal{T}^{(Q)}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) \leq \sum_{C \in [n]^{3t+1}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C) = n \,. \quad \square$$

We note that Definition 8.2 gives a partition of the r-chains, but the polynomial $\Psi(x)$ uses a restricted set of $(r+1)$-chains, namely those that have their head in $[k]$. In the following definition, we use the partition of the r-chains to induce a partition of the special $(r+1)$-chains.

Definition 8.4 (Induced partition of $\mathcal{H}_i^{(r+1)}$)**.** Let $r \geq 1$ be an integer. For each $0 \leq t \leq r$ and each $Q \in P_t$, we let $\mathcal{H}_i^{(r+1,Q)}$ denote the set of length $3r+4$ tuples of the form (i, w_1, w_2, C) where $C \in \mathcal{H}^{(r,Q)}$.

Definition 8.5 (Bipartite XOR formulas from a contiguously regular partition)**.** Fix integers $r, d \geq 1$. For each $1 \leq t \leq r$ and $Q \in P_t$, we define $\Psi_{i,Q}$ as the following XOR formula with terms corresponding to $(r+1)$-chains in $\mathcal{H}^{(r+1,Q)}$ with x_Q "modded out" from the corresponding monomial.

$$\Psi_{i,Q}(x) = \sum_{C = (i, v_1, v_2, u_1, \dots, u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \cdot x_{v_1} x_{v_2} \prod_{h=1}^{r} x_{\{v_{2h+1}, v_{2h+2}\} \setminus Q_h} \,.$$

Here, we use the convention that if $Q_h = \star$, then $\{v, v'\} \setminus Q_h := \{v, v'\}$.

For each $0 \leq t \leq r$, let $\Psi^{(t)}(x, y) = \sum_{i=1}^{k} \sum_{Q \in P_t} b_i y_Q \Psi_{i,Q}(x)$. Finally, we let $\Psi(x, y) = \sum_{0 \leq t \leq r} \Psi^{(t)}(x, y)$; here, for every heavy $Q \in P_t$ for some $0 \leq t \leq r$ used in the contiguously regular partition, we introduce a new variable y_Q.

We next observe that $\Psi(x, y)$ is a relaxation of the polynomial $\Psi(x)$. Indeed, we have abused notation and labeled them both as "Ψ" for this reason. This follows from the observation is that $\Psi(x, y)$ is produced by simply replacing the monomial x_Q in $\Psi(x)$ with a new variable y_Q for each heavy Q. More formally, the following holds.

Lemma 8.6. *Fix $x \in \{-1, 1\}^n$. Then, there is a $y \in \{-1, 1\}^{\sum_{t=0}^{r} |P_t|}$ such that $\Psi(x, y) = \Psi(x)$.*

Proof. For each $0 \le t \le r$, set $y_Q = x_Q$ for every $Q \in P_t$, where $x_Q := \prod_{h:Q_h \neq \star} x_{Q_h}$. □

We finish this section by proving the following statement, which intuitively shows that the partitions of the chains are smooth.

Lemma 8.7 (Smoothness of partitioned chains)**.** *Fix $i \in [k]$ and $t \in \{0, \dots, r\}$. Let $Z \in ([n] \cup \{\star\})^{r+1} \times \{\star\}$ be a Z that has a $\star$ in the last entry. Then, $\sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le (\delta n)^{-|Z|}$.*

Let $Q \in P_t$ and $\mathcal{H}_i^{(r+1,Q)}$ be as defined in Definition 8.4. Let $Z \in ([n] \cup \{\star\})^{r+1} \times [n]$ be such that Z extends Q, i.e., $Z_{r-t+h} = Q_h$ for all $1 \le h \le t+1$. Then, $\sum_{C \in \mathcal{H}_i^{(r+1,Q)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)$ is at most $\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|}$ if $|Z| \le r+1$, and at most $(\delta n)^{-r-1}$ if $|Z| = r+2$. Furthermore, if $d^{r+1} \ge n$, then $(\delta n)^{-r-1} \le \mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|}$.

Remark 8.8. We remark that this is place where we need the assumption that $d^{r+1} \ge n$.

Proof. The first statement follows immediately by δ-smoothness of the original hypergraphs. Indeed, for any $u \in [n]$ and $v \in [n]$, we have that $\sum_{C \in [n]^3 : v \in C} \mathrm{wt}_{H_u}(C) \le 1/\delta n$. We now have that

$$\sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C)$$

$$\le \sum_{\substack{(v_1, v_2, u_1) \\ Z_1 \in \{v_1, v_2\}}} \mathrm{wt}_{H_i}(v_1, v_2, u_1) \cdot \Bigg(\sum_{\substack{(v_3, v_4, u_2) \\ Z_2 \in \{v_3, v_4\}}} \mathrm{wt}_{H_{u_1}}(v_3, v_4, u_2) \Bigg(\cdots \Bigg(\sum_{\substack{(v_{2r+1}, v_{2r+2}, u_{r+1}) \\ Z_r \in \{v_{2r+1}, v_{2r+2}\}}} \mathrm{wt}_{H_{u_r}}(v_{2r+1}, v_{2r+2}, u_{r+1}) \Bigg) \cdots \Bigg) \Bigg).$$

We notice that the h-th term is at most $1/\delta n$ if $Z_h \neq \star$, and otherwise it is at most 1. So, in total, we get a bound of $(\delta n)^{-|Z|}$.

We now prove the second part of the statement. Let $|Q| = t+1$. We have two cases.

Case 1: Z does not contain a $\star$ entry. This means that $|Z| = r+2$. Let $Z' \in [n]^{r+1} \times \{\star\}$ be Z with the last entry replaced by a $\star$, i.e., $Z'_h = Z_h$ for all $1 \le h \le r+1$, and $Z'_{r+2} = \star$. We observe that

$$\sum_{C \in \mathcal{H}_i^{(r+1,Q)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le \sum_{C \in \mathcal{H}_i^{(r+1)} : Z \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le \sum_{C \in \mathcal{H}_i^{(r+1)} : Z' \subseteq C} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \le (\delta n)^{-|Z'|} = (\delta n)^{-r-1},$$

where we use the first statement that we have already shown. To finish the argument in this case, we need to argue that $\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|} \ge (\delta n)^{-r-1}$. Indeed, we have by definition that $\mathrm{wt}(Q) \ge n d^{|Q|-1} (\delta n)^{-|Q|}$, and so

$$\mathrm{wt}(Q) d^{|Z|-|Q|} (\delta n)^{-|Z|-1+|Q|} \ge \frac{1}{\delta} d^{|Z|-1} (\delta n)^{-|Z|} = (\delta n)^{-r-1} \cdot \frac{1}{\delta^2 n} d^{r+1}.$$

Thus, the desired inequality holds if $d^{r+1} \ge n$.

Case 2: Z contains a $\star$ entry. This means that $|Z| \le r+1$. Then, we have that $Z = (Z^{(1)}, \star, Z^{(2)}, Q)$, where $Z^{(2)}$ contains no $\star$ entries.

We observe that each $C \in \mathcal{H}_i^{(r+1,Q)}$ with $Z \subseteq C$ can be split into 3 parts: $C = (i, C^{(1)}, C^{(2)}, C^{(3)})$, where $C^{(3)} \in \mathcal{T}^{(Q)}$ is a length t chain, $(i, C^{(1)})$ is a length $|Z^{(1)}|$ chain with head i, and $C^{(2)}$ is a length

$r - t - |Z^{(1)}|$ chain whose head is the tail of $C^{(1)}$ and whose tail is the head of $C^{(3)}$. By δ-smoothness, $\sum_{C^{(1)} : Z^{(1)} \subseteq C^{(1)}} \mathrm{wt}_{\mathcal{H}_i^{(|Z^{(1)}|)}}(i, C^{(1)}) \leq (\delta n)^{-|Z^{(1)}|}$.

We either have that $(Z^{(2)}, Q)$ is Q, i.e., $Z^{(2)}$ is empty, or that $(Z^{(2)}, Q)$ is not Q. In the first case, $\sum_{C^{(3)} \in \mathcal{T}^{(Q)}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C^{(3)}) = \mathrm{wt}(Q)$ by definition (note that if $t = 0$, then $C^{(3)}$ is just the single vertex v where $Q = (v)$, and we have defined $\mathrm{wt}(Q) = 1$). In the second case, we observe that by Definitions 8.2 and 8.4, $(Z^{(2)}, Q)$ cannot be heavy. Indeed, if it was, then either $C^{(3)} \in \mathcal{T}^{(Z^{(2)}, Q)}$, and so $C \in \mathcal{H}_i^{(r+1, (Z^{(2)}, Q))}$, or else there is some other Q' with $|Q'| = |Z^{(2)}| + t + 1$ with $C^{(3)} \in \mathcal{T}^{(Q')}$, in which case we would have $C \in \mathcal{H}_i^{(r+1, Q')}$. We note that here we must use that Z contains at least one $\star$, so that $|Z^{(2)}| + |Q| \leq r + 1$. This is because all heavy Q' have $|Q'| \leq r + 1$, as they are defined for the length r-chains.

Thus, $(Z^{(2)}, Q)$ cannot be heavy. It then follows that $\sum_{C^{(3)} : C^{(3)} \notin \mathcal{T}^{(Q')} \ \forall Q' \in P_{t + |Z^{(2)}|}} \mathrm{wt}_{\mathcal{H}^{(t)}}(C^{(3)}) \leq n d^{|Z^{(2)}| + t} (\delta n)^{-|Z^{(2)}| - t - 1} \leq \mathrm{wt}(Q) d^{|Z^{(2)}|} (\delta n)^{-|Z^{(2)}|}$. We note that any $C \in \mathcal{H}_i^{(r+1, Q)}$ must have $C^{(3)} \notin \mathcal{T}^{(Q')} \ \forall Q' \in P_{t + |Z^{(2)}|}$, as otherwise we would violate Item (2) in Definition 8.2 since $|Z^{(2)}| \geq 1$.

To finish the proof, we observe that once $C^{(1)}$ and $C^{(3)}$ are chosen, the total weight of all "valid" $C^{(2)}$, i.e., $C^{(2)}$'s that could complete the chain to form $C \in \mathcal{H}_i^{(r+1, Q)}$, is at most $1/\delta n$. Indeed, this is because the head of $C^{(2)}$ is the tail of $C^{(1)}$ and its tail is the head of $C^{(3)}$, and the total weight of all length h chains, for any h, with a fixed head u and fixed tail v is at most $1/\delta n$ by δ-smoothness. Thus, in total, we have shown that $\sum_{C \in \mathcal{H}_i^{(r+1, Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \leq (\delta n)^{-|Z^{(2)}|} \cdot (\delta n)^{-1} \cdot \mathrm{wt}(Q) d^{|Z^{(2)}|} (\delta n)^{-|Z^{(2)}|} = \mathrm{wt}(Q) \cdot d^{|Z| - |Q|} (\delta n)^{-|Z| - 1 + |Q|}$. □

9 Spectral Refutation via Kikuchi Matrices

In Section 8, we defined polynomials $\Psi^{(t)}(x, y)$ and a map from $x \mapsto y$ such that $\Psi(x) = \sum_{t=0}^{r} \Psi^{(t)}(x, y)$ when y is the image of x under this map. Thus, to prove Lemma 6.6, we need to upper bound $\mathbb{E}_b[\mathrm{val}(\sum_{t=0}^{r} \Psi^{(t)}(x, y))]$. In this section, we will use the Kikuchi matrix method to bound this quantity, proving the second half of Lemma 6.6.

9.1 Step 1: the Cauchy–Schwarz trick

First, we show that we can relate $\sum_{t=0}^{r} \Psi^{(t)}(x, y)$ to a certain "cross-term" polynomial obtained via applying the Cauchy–Schwarz inequality.

Lemma 9.1 (Cauchy–Schwarz trick). *Let M be a maximum directed matching*[16][17] *of $[k]$ and let f_M be the cross-term polynomial defined as*

$$f_M^{(t)} = \sum_{\{i,j\} \in M} b_i b_j \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \Psi_{i,Q}(x) \Psi_{j,Q}(x)\,,$$

$$f_M = \sum_{t=0}^{r} f_M^{(t)}\,.$$

[16] A directed matching is a matching, only the edges are additionally directed

[17] This is a perfect matching if k is even, and will leave one element of $[k]$ unmatched if k is odd.

Then for every x, y with ±1 values, it holds that

$$\left(\sum_{t=0}^{r} \Psi^{(t)}(x,y)\right)^2 \leq n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2k\mathbb{E}_M[f_M]\right),$$

where the expectation $\mathbb{E}_M$ is over a uniformly random maximum directed matching M.

Proof. We will first apply the Cauchy–Schwarz inequality to eliminate the y variables:

$$\begin{aligned}
\left(\sum_{t=0}^{r} \Psi^{(t)}(x,y)\right)^2 &= \left(\sum_{t=0}^{r} \sum_{Q \in P_t} y_Q \cdot \sqrt{\mathrm{wt}(Q)} \left(\sum_{i=1}^{k} b_i \frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)\right)^2 \\
&\leq \left(\sum_{t=0}^{r} \sum_{Q \in P_t} y_Q^2 \mathrm{wt}(Q)\right)\left(\sum_{t=0}^{r} \sum_{Q \in P_t} \left(\sum_{i=1}^{k} b_i \frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)^2\right) \\
&\leq ((r+1)n)\left(\sum_{t=0}^{r} \sum_{Q \in P_t} \left(\sum_{i=1}^{k} b_i \frac{\Psi_{i,Q}}{\sqrt{\mathrm{wt}(Q)}}\right)^2\right) \\
&\leq n(r+1)\left(\sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{i,j=1}^{k} b_i b_j \Psi_{i,Q}\Psi_{j,Q}\right) \\
&\leq n(r+1)\left(\sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{i=1}^{k} \Psi_{i,Q}^2 + \sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{i \neq j \in [k]} b_i b_j \Psi_{i,Q}\Psi_{j,Q}\right).
\end{aligned}$$

By Lemma 8.7, we have that

$$|\Psi_{i,Q}(x)| \leq \sum_{C \in \mathcal{H}_i^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \leq \mathrm{wt}(Q) \cdot (\delta n)^{-1},$$

Hence, $\sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{i=1}^{k} \Psi_{i,Q}^2 \leq \frac{k}{\delta^2 n^2} \sum_{t=0}^{r} \sum_{Q \in P_t} \mathrm{wt}(Q) \leq \frac{k(r+1)}{\delta^2 n}$.

To finish the proof, we observe that the probability that a pair (i, j) is contained in a directed matching M is at least $\frac{1}{2k}$. □

9.2 Step 2: defining the Kikuchi matrices

It thus remains to bound $\mathrm{val}(f_M)$ for an arbitrary directed maximum matching M.

We define the Kikuchi matrices that we consider below.

Definition 9.2. Let $i, j \in [k]$ and $t \in \{0, \ldots, r\}$. Let $Q \in P_t$.

Let $C = (i, v_1, v_2, u_1, v_3, v_4, \ldots, u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}$ and $C' = (j, v'_1, v'_2, u_1, v'_3, v'_4, \ldots, u_{r+1}) \in \mathcal{H}_j^{(r+1,Q)}$. We let $A_{i,j}^{(C,C',Q)} \in \{0,1\}^{\binom{[n]}{\ell}^{2r+2}}$ be the matrix with rows and columns by indexed by $(2r+2)$-tuples of sets $(S_0, \ldots, S_r, S'_0, \ldots, S'_r)$ of size exactly ℓ defined as follows.

We set $A_{i,j}^{(C,C',Q)}((S_0, \ldots, S_r, S'_0, \ldots, S'_r), (T_0, \ldots, T_r, T'_0, \ldots T'_r))$ equal to 1 if the following holds, and otherwise we set this entry to be 0. In what follows, we let $C_h = \{v_{2h+1}, v_{2h+2}\}$, and we note that $|C_h| = 2$ for any chain with nonzero weight, by Definition 3.5.

1. For $h = 0, \dots, r-t$, we have $S_h \oplus T_h = C_h$ and $v_{2h+1} \in S_h$, $v_{2h+2} \in T_h$.
2. For $h = 0, \dots, r-t$, we have $S'_h \oplus T'_h = C'_h$ and $v'_{2h+1} \in S'_h$, $v'_{2h+2} \in T'_h$,
3. For $h = 1, \dots, t$, the following holds. Let $w_h = C_{r-t+h} \setminus Q_h$, and $w'_h = C'_{r-t+h} \setminus Q_h$. We have $S_{r-t+h} = R \cup \{w_h\}$, $T_{r-t+h} = R \cup \{w'_h\}$, and $S'_{r-t+h} = T'_{r-t+h}$.[18]

We let $A^{(t)}_{i,j} = \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \cdot A^{(C,C',Q)}_{i,j}$ and $A_{i,j} = \sum_{t=0}^{r} \frac{1}{D_t} A^{(t)}_{i,j}$, where $D_t = \binom{n-2}{\ell-1}^{2r+2-t} \cdot \binom{n}{\ell}^t$. For any matching M on $[k]$, let $A_M = \sum_{(i,j) \in M} b_i b_j A_{i,j}$. We will abuse notation and let $A := A_M$.

The following lemma shows that we can express $f_M(x)$ as a (scaling of a) quadratic form on the matrix $A^{(t)}$.

Lemma 9.3. *Let $x \in \{-1,1\}^n$, and let $x' \in \{-1,1\}^N$, where $N = \binom{n}{\ell}^{2r+2}$, denote the vector where the $(S_0, S_1, \dots, S_r, S'_0, S'_1, \dots, S'_r)$-th entry of x' is $\prod_{h=0}^{r} x_{S_h} x_{S'_h}$. Let $i, j \in [k]$ and $t \in \{0, \dots, r\}$. Let $Q \in P_t$, and let let $C = (i, v_1, v_2, u_1, v_3, v_4, \dots, u_{r+1}) \in \mathcal{H}_i^{(r+1,Q)}$ and $C' = (j, v'_1, v'_2, u_1, v'_3, v'_4, \dots, u_{r+1}) \in \mathcal{H}_j^{(r+1,Q)}$. Then,*

$$x'^{\top} A^{(C,C',Q)}_{i,j} x' = D_t x_{v_1} x_{v_2} \prod_{h=1}^{r} x_{\{v_{2h+1}, v_{2h+2}\} \setminus Q_h} \cdot x_{v'_1} x_{v'_2} \prod_{h=1}^{r} x_{\{v'_{2h+1}, v'_{2h+2}\} \setminus Q_h} ,$$

i.e., the product of the monomials associated to C and C', modded out by Q_h, where $D_t = \binom{n-2}{\ell-1}^{2r+2-t} \cdot \binom{n}{\ell}^t$. Moreover, for any matrix $B^{(C,C',Q)}_{i,j}$ obtained by "zeroing out" exactly αD_t entries of $A^{(C,C',Q)}_{i,j}$, the equality holds with a factor of $1 - \alpha$ on the right.

In particular, $x'^{\top} A x' = f_M(x)$.

Proof. Let $\vec{S} = (S_0, S_1, \dots, S_r, S'_0, S'_1, \dots, S'_r)$ and $\vec{T} = (T_0, \dots, T_r, T'_0, \dots T'_r)$ be such that $A^{(\vec{C},\vec{C}',Q)}_{i,j}(\vec{S}, \vec{T}) = 1$. Then, we have that

$$\begin{aligned} x'_{\vec{S}} x'_{\vec{T}} &= \prod_{h=0}^{r} x_{S_h} x_{T_h} x_{S'_h} x_{T'_h} = \prod_{h=0}^{r-t} x_{S_h \oplus T_h} x_{S'_h \oplus T'_h} \prod_{h=1}^{t} x_{S_{r-t+h} \oplus T_{r-t+h}} x_{S'_{r-t+h} \oplus T'_{r-t+h}} \\ &= \prod_{h=0}^{r-t} x_{C_h} x_{C'_h} \prod_{h=1}^{t} x_{C_{r-t+h} \setminus Q_h} x_{C'_{r-t+h} \setminus Q_h} , \end{aligned}$$

which is equal to the product of monomials on the right-hand side of the equation we wish to show.

It thus remains to argue that $A^{(\vec{C},\vec{C}',Q)}_{i,j}$ has exactly D_t nonzero entries. We observe that, for each $h = 0, \dots, r-t$, there are exactly $\binom{n-2}{\ell-1}$ pairs (S_h, T_h) such that $S_h \oplus T_h = C_h$ with $v_{2h+1} \in S_h$ and

[18] It is possible that one could have $w_h = w'_h$ here. In that case, we pick a canonical extra vertex v, and require that $v \notin R$ as well. This is to ensure that the number of choices here for S_{r-t+h} and S'_{r-t+h} is *exactly* $\binom{n-2}{\ell-1}\binom{n}{\ell}$; otherwise it would be $\binom{n-1}{\ell-1}\binom{n}{\ell}$. The difference in the two cases is immaterial but it is convenient to have an exact count.

$v_{2h+2} \in T_h$. Indeed, this is because we must simply choose a set of size $\ell - 1$ that does not contain either of v_{2h+1} and v_{2h+2}, and then this determines S_h and T_h.

For $h = 1, \dots, t$, there are exactly $\binom{n-2}{\ell-1}$ choices of (S_{r-t+h}, T_{r-t+h}). Indeed, this is because S_{r-t+h} must contain w_h and T_{r-t+h} must contain w'_h. Note that if $w_h = w'_h$, then there are actually $\binom{n-1}{\ell-1}$ choices! However, using the slightly modified definition of the matrix in the footnote in Definition 9.2, we can again force there to be exactly $\binom{n-2}{\ell-1}$ choices. Finally, there are $\binom{n}{\ell}$ choices for (S'_{r-t+h}, T'_{r-t+h}), as we must have $S'_{r-t+h} = T'_{r-t+h}$.

Combining, we see that $D_t = \binom{n-2}{\ell-1}^{2(r-t+1)} \cdot (\binom{n-2}{\ell-1}\binom{n}{\ell})^t = \binom{n-2}{\ell-1}^{2r+2-t}\binom{n}{\ell}^t$, as required. □

9.3 Step 3: finding a regular submatrix of the Kikuchi matrix

By Lemma 9.3, in order to upper bound $\mathbb{E}_b[\mathrm{val}(\sum_{t=0}^{r} f_M^{(t)})]$, it suffices to bound $\mathbb{E}_b[\|A\|_{\infty\to 1}] \le N\mathbb{E}_b[\|A\|_2]$, where $N = \binom{n}{\ell}^{2r+2}$; here, we use that $\|A\|_{\infty\to 1} \le N\|A\|_2$ always holds.

To bound $\|A\|_2$, we will write $A = \sum_{(i,j)\in M} b_i b_j A_{i,j}$ and apply Fact 3.11. To do this, we need to bound $\|A_{i,j}\|_2$, which we shall do by upper bounding the maximum ℓ_1-norm of any row/column of the matrix. In turns out there are some rows that indeed have a large ℓ_1-norm. To handle this issue, we shall zero out the "bad rows", as follows. To do this, we will need to use the following technical lemma, proven in Section 10, that bounds the expected ℓ_1-norm of a row and the conditional expectation given that the row has a nonzero entry in a specific matrix $A_{i,j}^{(C,C',Q)}$.

Lemma 9.4 (First and conditional moment bounds). *Fix $r \ge 1$, $i, j \in [k]$, and let $\mathcal{H}_i^{(r+1)}$ and $\mathcal{H}_j^{(r+1)}$ denote the $(r+1)$-chain hypergraph with heads in i and j respectively. Let $\cup_{t=0}^{r} \cup_{Q\in P_t} \mathcal{H}_i^{(r+1,Q)}$ be a smooth partition of $\mathcal{H}_i^{(r+1)}$, as defined in Definitions 8.2 and 8.4. Let $A_{i,j}$ be the Kikuchi matrix defined in Definition 9.2, which depends on r, i, j, and the pieces $\cup_{Q\in P_t}\mathcal{H}^{(r+1,Q)}$ of the refinement, and the matching M.*

Let $\vec{S} = (S_0, \dots, S_r, S'_0, \dots, S'_r) \in \binom{[n]}{\ell}^{2r+2}$ be a row of the matrix, and let $\deg_{i,j}(\vec{S})$ denote the ℓ_1-norm of the $\vec{S}$-th row of $A_{i,j}$. Then,

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \le \frac{1}{N \cdot \delta n},$$

where $N = \binom{n}{\ell}^{2r+2}$.

Furthermore, let $t \in \{0, \dots, r\}$, $Q \in P_t$, and $C \in \mathcal{H}_i^{(r+1,Q)}$ and $C' \in \mathcal{H}_j^{(r+1,Q)}$. Let $\mathcal{D}_{C,C',Q}$ denote the uniform distribution over rows of $A_{i,j}^{(C,C',Q)}$ that contain a nonzero entry. Then, if $d^{r+1} \ge n$ and $\ell \ge 2d(r+1)/\delta$, it holds that

$$\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \le \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{4}{N\delta n}.$$

Let us now use Lemma 9.4 to argue the following. For a sufficiently large constant Γ, there exist submatrices $B_{i,j}^{(C,C',Q)}$, i.e., a $\{0,1\}$-matrix where $B_{i,j}^{(C,C',Q)}(\vec{S}, \vec{T}) = 1$ implies $A_{i,j}^{(C,C',Q)}(\vec{S}, \vec{T}) = 1$, such that (1) each $B_{i,j}^{(C,C',Q)}$ contains exactly $D_t/100$ nonzero entries, and (2) the ℓ_1-norm of any row/column of $B_{i,j}$ (defined analogously to $A_{i,j}$) is at most $\frac{\Gamma}{N\cdot\delta n}$.

We do this as follows. First, we observe that $A^{(C,C',Q)}_{i,j}(\vec{S},\vec{T}) = A^{(C',C,Q)}_{j,i}(\vec{R},\vec{W})$, where $\vec{R} = (S'_0,\dots,S'_r,S_0,\dots,S_r)$ and $\vec{W} = (T'_0,\dots,T'_r,T_0,\dots,T_r)$. In particular, this symmetry implies that the bounds on the moments for rows in Lemma 9.4 hold for columns as well.

Let $\mathcal{B}_1 = \{\vec{S} : \deg_{i,j}(\vec{S}) \geq \frac{\Gamma}{N\cdot\delta n}\}$ denote the set of bad rows with ℓ_1-norm at least $\frac{\Gamma}{N\cdot\delta n}$, and similarly let $\mathcal{B}_2$ be the same but for the columns. Applying Markov's inequality and the conditional degree bound, we see that $\mathcal{B}_1$ contains at most $O(1/\Gamma)$-fraction of the rows where $A^{(C,C',Q)}_{i,j}$ is nonzero, and similarly $\mathcal{B}_2$ contains at most $O(1/\Gamma)$-fraction of the columns where $A^{(C,C',Q)}_{i,j}$ is nonzero. Thus, after removing these rows, we still have at least $(1-O(1/\Gamma))D_t$ nonzero entries in $A^{(C,C',Q)}_{i,j}$. When Γ is a sufficiently large constant, this is at least $1/2$, and so we can choose an arbitrary subset of *exactly* $D_t/2$ nonzero entries. We let $B^{(C,C',Q)}_{i,j}$ be the matrix with those nonzero entries.

The first property is clearly satisfied by construction. The second property is satisfied because the ℓ_1-norm of any row/column of $B_{i,j}$ is clearly at most $\frac{\Gamma}{N\cdot\delta n}$, again by construction.

9.4 Step 4: finishing the proof

Let $B^{(C,C',Q)}_{i,j}$ be the matrix produced in Section 9.3.

We let $B^{(t)}_{i,j} = \sum_{Q\in P_t}\frac{1}{\mathrm{wt}(Q)}\sum_{C\in\mathcal{H}^{(r+1,Q)}_i, C'\in\mathcal{H}^{(r+1,Q)}_j}\mathrm{wt}_{\mathcal{H}^{(r+1)}_i}(C)\mathrm{wt}_{\mathcal{H}^{(r+1)}_j}(C')\cdot B^{(C,C',Q)}_{i,j}$ and $B_{i,j} = \sum_{t=0}^{r}\frac{1}{D_t}B^{(t)}_{i,j}$. For any matching M on $[k]$, let $B_M = \sum_{(i,j)\in M} b_i b_j B_{i,j}$. We will abuse notation and let $B := B_M$.

By Lemma 9.3 and the fact that $B^{(C,C',Q)}_{i,j}$ has exactly $D_t/2$ nonzero entries of $A^{(C,C',Q)}_{i,j}$ in it, we see that for every $x \in \{-1,1\}^n$, there exists $x' \in \{-1,1\}^N$ such that $x'^{\top}Bx' = \frac{1}{2}f_M(x)$. We also have that $\|B_{i,j}\|_2 \leq \frac{\Gamma}{N\cdot\delta n}$, by construction in Section 9.3.

By Fact 3.11, it therefore follows that

$$\mathbb{E}_b[\mathrm{val}(f_M(x))] \leq 2\mathbb{E}_b[N\|B\|_2] \leq N\cdot\frac{\Gamma}{N\cdot\delta n}\cdot O(\sqrt{k\log N}) = O(\sqrt{k\ell r\log n})\cdot\frac{1}{\delta n}$$

Hence,

$$\begin{aligned}\mathbb{E}_b[\mathrm{val}(\Psi(x,y))]^2 &\leq \mathbb{E}_b[\mathrm{val}(\Psi(x,y)^2)] \leq n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2k\mathbb{E}_M[f_M]\right)\\ &\leq n(r+1)\left(\frac{k(r+1)}{\delta^2 n} + 2kO(\sqrt{k\ell r\log n})\cdot\frac{1}{\delta n}\right) = \frac{k(r+1)}{\delta}\left(\frac{r+1}{\delta} + 2O(\sqrt{k\ell r\log n})\right)\\ &\leq \frac{k(r+1)}{\delta}O(\sqrt{k\ell r\log n}),\end{aligned}$$

as $\ell \geq O(r/\delta)$ and we can assume that $k \geq 1/\delta$ (as otherwise we are already done).

10 Row Pruning: Proof of Lemma 9.4

In this section, we prove Lemma 9.4, restated below.

Lemma 10.1 (First and conditional moment bounds)**.** *Fix $r \geq 1$, $i, j \in [k]$, and let $\mathcal{H}_i^{(r+1)}$ and $\mathcal{H}_j^{(r+1)}$ denote the $(r+1)$-chain hypergraph with heads in i and j respectively. Let $\cup_{t=0}^r \cup_{Q \in P_t} \mathcal{H}_i^{(r+1,Q)}$ be a smooth partition of $\mathcal{H}_i^{(r+1)}$, as defined in Definitions 8.2 and 8.4. Let $A_{i,j}$ be the Kikuchi matrix defined in Definition 9.2, which depends on r, i, j, and the pieces $\cup_{Q \in P_t} \mathcal{H}^{(r+1,Q)}$ of the refinement, and the matching M.*

Let $\vec{S} = (S_0, \dots, S_r, S'_0, \dots, S'_r) \in \binom{[n]}{\ell}^{2r+2}$ be a row of the matrix, and let $\deg_{i,j}(\vec{S})$ denote the ℓ_1-norm of the $\vec{S}$-th row of $A_{i,j}$. Then,

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{N \cdot \delta n} ,$$

where $N = \binom{n}{\ell}^{2r+2}$.

Furthermore, let $t \in \{0, \dots, r\}$, $Q \in P_t$, and $C \in \mathcal{H}_i^{(r+1,Q)}$ and $C' \in \mathcal{H}_j^{(r+1,Q)}$. Let $\mathcal{D}_{C,C',Q}$ denote the uniform distribution over rows of $A_{i,j}^{(C,C',Q)}$ that contain a nonzero entry. Then, if $d^{r+1} \geq n$ and $\ell \geq 2d(r+1)/\delta$, it holds that

$$\mathbb{E}_{\vec{S} \sim \mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \leq \left(1 + \frac{O(\ell r)}{n}\right) \cdot \frac{4}{N \delta n} .$$

Proof. We begin by estimating the first moment, i.e., $\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})]$. By definition, we have that

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] = \frac{1}{N} \sum_{t=0}^{r} \frac{1}{D_t} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \cdot D_t$$

$$= \frac{1}{N} \sum_{t=0}^{r} \sum_{Q \in P_t} \frac{1}{\mathrm{wt}(Q)} \sum_{C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') .$$

We note that the latter quantity is simply equal to $\frac{1}{N} \sum_{C \in \mathcal{H}_i^{(r+1)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \sum_{C' \in \mathcal{H}_j^{(r+1,Q)} : C \in \mathcal{H}_i^{(r+1,Q)}} \frac{1}{\mathrm{wt}(Q)} \cdot \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C')$, where the second sum is over $C' \in \mathcal{H}_j^{(r+1,Q)}$ where Q is determined by the choice of C. We note that for any Q, $\sum_{C' \in \mathcal{H}_j^{(r+1,Q)}} \mathrm{wt}_{\mathcal{H}_j^{(r+1)}}(C') \leq \frac{\mathrm{wt}(Q)}{\delta n}$, and hence we conclude that

$$\mathbb{E}_{\vec{S}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{N} \sum_{C \in \mathcal{H}_i^{(r+1)}} \mathrm{wt}_{\mathcal{H}_i^{(r+1)}}(C) \frac{1}{\delta n} \leq \frac{1}{N \cdot \delta n} .$$

Next, we estimate the conditional first moment. Fix a $Q \in P_t$ for some $0 \leq t \leq r$, and let $C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}$. We now bound $\mathbb{E}_{\vec{S} \sim \mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})]$, where $\mathcal{D}_{C,C',Q}$ is the uniform distribution over all rows $\vec{S}$ such that $A_{i,j}^{(C,C',Q)}$ has a nonzero entry. We note that there are exactly D_t such rows.

We shall proceed in two steps. First, we consider a fixed (D, D', Q') with $D \in \mathcal{H}_i^{(r+1,Q')}, D' \in \mathcal{H}_j^{(r+1,Q')}$. Let $|Q'| = t' + 1$. We will upper bound the number of rows $\vec{S}$ where $A_{i,j}^{(C,C',Q)}$ and $A_{i,j}^{(D,D',Q')}$, normalized by the factor of $1/D_{t'}$. This will depend on the number of shared vertices z between these two pairs of chains, for an appropriate definition of shared vertices. Then, we will, for each choice of z, bound the total weight of the number of chains (D, D', Q') have "intersection z" with (C, C', Q), which will conclude the argument.

Step 1: bounding the normalized number of entries for a fixed (D, D', Q'). To begin, we will define the number of "shared vertices" between two pairs of chains (C, C', Q) and (D, D', Q').

Definition 10.2 (Left vertices)**.** Let (C, C', Q) be such that $Q \in P_t$ and $C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}$. Let $C = (i, v_1, v_2, u_1, \dots, u_{r+1})$ and $C' = (j, v'_1, v'_2, u'_1, \dots, u'_{r+1})$. The tuple of *left vertices* of (C, C', Q) is the sequence $(v_1, v_3, v_5, \dots, v_{2(r-t)+1}, w_1, \dots, w_t, v'_1, v'_3, \dots, v'_{2(r-t)+1})$, where $C_h = \{v_{2h+1}, v_{2h+2}\} = \{w_h, Q_h\}$. We denote this sequence by $L(C, C', Q)$.

Remark 10.3. The reason for the above definition is the following. If $\vec{S}$ is a row where the matrix $A_{i,j}^{(C,C',Q)}$ has a nonzero entry, then the entries of $L(C, C', Q)$ (in order) are contained in the sets $(S_0, \dots, S_{r-t}, S_{r-t+1}, \dots, S_r, S'_0, \dots, S'_{r-t})$, e.g., $v_1 \in S_0$, $v_3 \in S_1$, $w_1 \in S_{r-t+1}$, etc.

Definition 10.4 (Intersection patterns)**.** Let (C, C', Q) and (D, D', Q') be such that $C \in \mathcal{H}_i^{(r+1,Q)}, C' \in \mathcal{H}_j^{(r+1,Q)}$ and $D \in \mathcal{H}_i^{(r+1,Q')}, D' \in \mathcal{H}_j^{(r+1,Q')}$.

The *intersection pattern* of (C, C', Q) and (D, D', Q'), given by $Z \in \{0, 1\}^{2r+2-t}$, is defined as $Z_h = 1$ if $L(C, C', Q)_h = L(D, D', Q')_h$, and it is 0 otherwise. Note that the sequences $L(C, C', Q)$ and $L(D, D', Q')$ may not have the same length; if h is "out of bounds" for $L(D, D', Q')$, then we set $Z_h = 0$.

We now fix (D, D', Q') and count the number of rows as a function of the intersection pattern Z. Let $t' = |Q'| - 1$. We have two cases. In the first case, $t \geq t'$, which implies that $|L(C, C', Q)| \leq |L(D, D', Q)|$. We observe that in order for a row $\vec{S}$ to have a nonzero entry for both pairs of chains, the following must hold:

1. for $h = 1, \dots, r + 2$ (the first $r + 1$ sets), we have $\{L(C, C', Q)_h, L(D, D', Q)_h\} \subseteq S_h$,
2. for $h = r + 2, \dots, 2r + 3 - t$ (the next $r + 1 - t$ sets), we have $\{L(C, C', Q)_h, L(D, D', Q)_h\} \subseteq S'_{h-(r+2)}$,
3. for $h = 2r + 3 - t, \dots, 2r + 2 - t'$ (the next $t - t'$ sets), we have $L(D, D', Q)_h \in S'_{h-(r+2)}$,
4. for $h = 2r + 2 - t' + 1, \dots, 2r + 2$ (the final t' sets), we have $S'_{h-(r+2)}$ is arbitrary.

We observe that for each intersection point, i.e., an h such that $L(C, C', Q)_h = L(D, D', Q)_h$, there are $\binom{n}{\ell-1}$ choices for the corresponding set, as it needs to only contain one vertex. For each nonintersection point, i.e., an $h \in \{1, \dots, 2r + 2 - t\}$ where $L(C, C', Q)_h \neq L(D, D', Q)_h$, we have $\binom{n}{\ell-2}$ choices, because the set needs to contain both vertices. Finally, we have $\binom{n}{\ell-1}$ choices for each of the $t - t'$ sets in the third case, and $\binom{n}{\ell}$ choices for the last t sets in the final case. In total, we have $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{2r+2-t-z} \binom{n}{\ell-1}^{t-t'} \binom{n}{\ell}^{t'}$.

In the second case, $t \leq t'$. We observe that by swapping the roles of t and t' above, we get a bound of $\binom{n}{\ell-1}^z \binom{n}{\ell-2}^{2r+2-t'-z} \binom{n}{\ell-1}^{t'-t} \binom{n}{\ell}^{t}$.

Now, although the above counts are different, we observe that they are within constant factors of each other. Indeed, we have

$$\frac{\binom{n}{\ell-2}^{-t'}\binom{n}{\ell-1}^{t'-t}\binom{n}{\ell}^{t}}{\binom{n}{\ell-2}^{-t}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}} = \left(\binom{n}{\ell-2}^{-1}\binom{n}{\ell-1}^{2}\binom{n}{\ell}^{-1}\right)^{t'-t}$$
$$= \left(\frac{\ell(n-\ell+2)}{(\ell-1)(n-\ell+1)}\right)^{t'-t} = \left(1+\frac{n-1}{(\ell-1)(n-\ell+1)}\right)^{t'-t} ,$$

and this ratio is between $\frac{1}{2}$ and 2 since $|t'-t| \leq r$ and $\frac{n-1}{(\ell-1)(n-\ell+1)} \geq \frac{2}{\ell} \geq \frac{1}{\Gamma r}$ for a sufficiently large constant Γ.

Next, we observe that while we have an upper bound of $2 \cdot \binom{n}{\ell-1}^{z}\binom{n}{\ell-2}^{2r+2-t}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}$ on the number of rows, which depends on t', each entry has a scaling factor of $\frac{1}{D_{t'}}$. We now give an upper bound on the *normalized* number of entries that does not depend on t'. We have

$$2\frac{\binom{n}{\ell-1}^{z}\binom{n}{\ell-2}^{2r+2-t-z}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}}{D_{t'}} = 2\frac{\binom{n}{\ell-1}^{z}\binom{n}{\ell-2}^{2r+2-t-z}\binom{n}{\ell-1}^{t-t'}\binom{n}{\ell}^{t'}}{\binom{n-2}{\ell-1}^{2r+2-t'}\cdot\binom{n}{\ell}^{t'}} = 2\left(\frac{\binom{n}{\ell-2}}{\binom{n}{\ell-1}}\right)^{2r+2-t-z}\cdot\left(\frac{\binom{n}{\ell-1}}{\binom{n-2}{\ell-1}}\right)^{2r+2-t'}$$
$$= 2\left(\frac{\ell-1}{n-\ell+2}\right)^{2r+2-t-z}\cdot\left(\frac{n(n-1)}{(n-\ell+1)(n-\ell)}\right)^{2r+2-t'}$$
$$\leq 2\left(\frac{\ell}{n}\right)^{2r+2-t-z}\cdot\left(1+\frac{O(\ell r)}{n}\right) .$$

Step 2: bounding the weight of (D, D', Q') with a fixed intersection pattern Z. Let us fix the intersection pattern Z and then determine the total weight of all (D, D', Q') with $D \in \mathcal{H}_i^{(r+1,Q')}$, $D' \in \mathcal{H}_j^{(r+1,Q')}$ with these intersection points. To do this, we will apply Lemma 8.7.

First, we observe that fixing an intersection pattern induces a $Z^{(1)} \in \{[n] \cup \{\star\}\}^{r+1} \times \{\star\}$, simply by filling in $Z^{(1)}$'s non-$\star$ entries with the appropriate vertices of $L(C, C', Q)$. We note that such a $Z^{(1)}$ never has the tail filled in, as the tail is not a potential intersection point. By Lemma 8.7, this implies that the total weight of D that contain $Z^{(1)}$ is at most $(\delta n)^{-|Z^{(1)}|}$.

Next, we bound the total weight of all D' that are valid for a fixed D. We observe that $D \in \mathcal{H}_i^{(r+1,Q')}$ for some i, and hence D' must be in $\mathcal{H}_j^{(r+1,Q')}$. We note that Z induces an intersection pattern $Z^{(2)}$ on D', and moreover $Z^{(2)}$ does not intersect with the "Q'-part" of the chain D', namely the links that contain vertices from Q'. So, it follows that D' contains $(Z^{(2)}, Q')$.

By Lemma 8.7, we have that the total weight of all D' is at most $\mathrm{wt}(Q)d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$. As each entry in $A_{i,j}^{(D,D',Q')}$ is scaled down by a factor of $\mathrm{wt}(Q')$, the normalized weight is therefore at most $d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$.

In total, we get a bound of $(\delta n)^{-|Z^{(1)}|} \cdot d^{|Z^{(2)}|}(\delta n)^{-|Z^{(2)}|-1}$, which is at most $d^{|Z|}(\delta n)^{-|Z|-1}$. Here, we use that $|Z| = |Z^{(1)}| + |Z^{(2)}|$.

Putting it all together. By combining steps (1) and (2) (and paying an additional $\binom{2r+2-t}{z}$ factor to choose the nonzero entries of Z), we thus obtain the final bound of

$$\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \leq \frac{1}{D_t}\sum_{z=0}^{2r+2-t}\binom{2r+2-t}{z}\cdot\left(1+\frac{O(\ell r)}{n}\right)\cdot 2\left(\frac{\ell}{n}\right)^{2r+2-t-z}\cdot d^{z}(\delta n)^{-z-1}$$

$$
\begin{aligned}
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t}\left(\frac{\ell}{n}\right)^{2r+2-t}\cdot\sum_{z=0}^{2r+2-t}(2r+2-t)^z\cdot\left(\frac{\ell}{n}\right)^{-z}\cdot d^z(\delta n)^{-z-1}\\
&= \left(1+\frac{O(\ell r)}{n}\right)\frac{2}{D_t\cdot\delta n}\left(\frac{\ell}{n}\right)^{2r+2-t}\cdot\sum_{z=0}^{2r+2-t}\left(\frac{(2r+2-t)\cdot d}{\delta\ell}\right)^z\\
&\le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{D_t\cdot\delta n}\left(\frac{\ell}{n}\right)^{2r+2-t},
\end{aligned}
$$

where we use that $\ell \ge 2d(2r+2)/\delta$.

To finish the proof, we need to compute $\frac{D_t}{N}$. We have that

$$
\begin{aligned}
\frac{D_t}{N} &= \frac{\binom{n-2}{\ell-1}^{2r+2-t}\cdot\binom{n}{\ell}^t}{\binom{n}{\ell}^{2r+2}} = \left(\frac{\binom{n-2}{\ell-1}}{\binom{n}{\ell}}\right)^{2r+2-t} = \left(\frac{\ell(n-\ell)}{n(n-1)}\right)^{2r+2-t}\\
&\ge \left(\frac{\ell}{n}\right)^{2r+2-t}\cdot\left(1-\frac{\ell-1}{n-1}\right)^{2r+2-t} \ge \left(\frac{\ell}{n}\right)^{2r+2-t}\left(1-\frac{(\ell-1)(2r+2)}{n-1}\right) = \left(\frac{\ell}{n}\right)^{2r+2-t}\left(1-\frac{O(\ell r)}{n}\right),
\end{aligned}
$$

Thus,

$$
\mathbb{E}_{\vec{S}\sim\mathcal{D}_{C,C',Q}}[\deg_{i,j}(\vec{S})] \le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{D_t\cdot\delta n}\left(\frac{\ell}{n}\right)^{2r+2-t} \le \left(1+\frac{O(\ell r)}{n}\right)\frac{4}{N\cdot\delta n},
$$

which finishes the proof. □

References

[AG24] Omar Alrabiah and Venkatesan Guruswami. Near-tight bounds for 3-query locally correctable binary linear codes via rainbow cycles. *Electron. Colloquium Comput. Complex.*, TR24-062, 2024.

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom CSP refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1438–1448. ACM, 2023.

[AK92] E. F. Assmus and J. D. Key. *Designs and their Codes*. Cambridge Tracts in Mathematics. Cambridge University Press, 1992.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.*, 3:307–318, 1993.

[BGT17] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[BIW10] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private pir. *Algorithmica*, 58(4):831–859, 2010.

[BJM23] Nikhil Bansal, Haotian Jiang, and Raghu Meka. Resolving matrix spencer conjecture up to poly-logarithmic rank. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1814–1819. ACM, 2023.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *Journal of the ACM (JACM)*, 42(1):269–291, 1995.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–595, 1993.

[BSS14] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Rev.*, 56(2):315–334, 2014.

[DHV78] Jean Doyen, Xavier Hubaut, and Monique Vandensavel. Ranks of incidence matrices of steiner triple systems. *Mathematische Zeitschrift*, 163:251–259, 1978.

[Dvi10] Zeev Dvir. On matrix rigidity and locally self-correctable codes. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 291–298. IEEE Computer Society, 2010.

[Dvi12] Zeev Dvir. Incidence theorems and their applications. *CoRR*, abs/1208.5073, 2012.

[Dvi16] Zeev Dvir. Lecture notes on linear locally decodable codes. `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC8.pdf`, Fall 2016.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 39–44. ACM, 2009.

[GKM22] Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. Algorithms and certificates for Boolean CSP refutation: smoothed is no harder than random. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 678–689. ACM, 2022.

[GKST06] Oded Goldreich, Howard Karloff, Leonard J Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[Ham73] Noboru Hamada. On the p-rank of the incidence matrix of a balanced or partially balanced incomplete block design and its applications to error correcting codes. *Hiroshima Mathematical Journal*, 3(1):153–226, 1973.

[HKM23] Jun-Ting Hsieh, Pravesh K. Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph Moore bound. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2324–2344. SIAM, 2023.

[HKM$^+$24] Jun-Ting Hsieh, Pravesh K. Kothari, Sidhanth Mohanty, David Munhá Correia, and Benny Sudakov. Small even covers, locally decodable codes and restricted subgraphs of edge-colored kikuchi graphs. *CoRR*, abs/2401.11590, 2024.

[HO75] N Hamada and H Ohmori. On the bib design having the minimum p-rank. *Journal of Combinatorial Theory, Series A*, 18(2):131–140, 1975.

[IK99] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 79–88. ACM, 1999.

[JT09] Dieter Jungnickel and Vladimir D. Tonchev. Polarities, quasi-symmetric designs, and Hamada's conjecture. *Des. Codes Cryptogr.*, 51(2):131–140, 2009.

[Jun84] Dieter Jungnickel. The number of designs with classical parameters grows exponentially. *Geom. Dedicata*, 16(2):167–178, 1984.

[Jun11] Dieter Jungnickel. Recent results on designs with classical parameters. *J. Geom.*, 101(1-2):137–155, 2011.

[Kan94] William M. Kantor. Automorphisms and isomorphisms of symmetric and affine designs. *J. Algebraic Combin.*, 3(3):307–338, 1994.

[KM23] Pravesh K. Kothari and Peter Manohar. An exponential lower bound for linear 3-query locally correctable codes. *CoRR*, abs/2311.00558, 2023.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86, 2000.

[KV00] Jeong Han Kim and Van H Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.

[KW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69(3):395–420, 2004.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 2–10. IEEE Computer Society, 1990.

[LLT00] Clement Lam, Sigmund Lam, and Vladimir D. Tonchev. Bounds on the number of affine, symmetric, and Hadamard designs and matrices. *J. Combin. Theory Ser. A*, 92(2):186–196, 2000.

[LLT01] Clement Lam, Sigmund Lam, and Vladimir D. Tonchev. Bounds on the number of Hadamard designs of even order. *J. Combin. Des.*, 9(5):363–378, 2001.

[LP91] Françoise Lust-Piquard and Gilles Pisier. Noncommutative Khintchine and Paley inequalities. *Ark. Mat.*, 29(2):241–260, 1991.

[LT02] Clement Lam and Vladimir D. Tonchev. A new bound on the number of designs with classical affine parameters. volume 27, pages 111–117. 2002. Special issue in honour of Ronald C. Mullin, Part II.

[MSS15] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families II: Mixed characteristic polynomials and the Kadison-Singer problem. *Ann. of Math. (2)*, 182(1):327–350, 2015.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[Sha90] Adi Shamir. Ip=pspace. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15. IEEE Computer Society, 1990.

[SS12] Warren Schudy and Maxim Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 437–446. SIAM, 2012.

[Tei80] Luc Teirlinck. On projective and affine hyperplanes. *Journal of Combinatorial Theory, Series A*, 28(3):290–306, 1980.

[Ton99] Vladimir D Tonchev. Linear perfect codes and a characterization of the classical designs. *Designs, Codes and Cryptography*, 17:121–128, 1999.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *arXiv preprint cs/0409044*, 2004.

[Tro15] Joel A. Tropp. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1-2):1–230, 2015.

[WAM19] Alexander S. Wein, Ahmed El Alaoui, and Cristopher Moore. The Kikuchi Hierarchy and Tensor PCA. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1446–1468. IEEE Computer Society, 2019.

[Yan24] Tal Yankovitz. A stronger bound for linear 3-lcc. *Electron. Colloquium Comput. Complex.*, pages TR24–036, 2024.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

A The Case of Imperfect Completeness in Theorem 2

In this appendix, we prove Theorem 2 when the code has completeness $1 - \varepsilon$. The proof is essentially identical to the proof in the perfect completeness case presented in Sections 5 to 10, with only minor changes that we describe here.

First, we observe that the reduction in Section 5.1 holds with the following minor change: the decoding function $f_{v_1,a_1,v_2,a_2}(a_3)$ for $C = (v_1, a_1, v_2, a_2, v_3)$ might not be deterministic. This means that the function $f_{v_1,a_1,v_2,a_2}(a_3)$ is a convex combination of the deterministic functions specified in Section 5.1, and so we may need to add multiple copies of $C = (v_1, a_1, v_2, a_2, v_3)$ with different weights to handle the different deterministic functions in the convex combination. This only introduces some minor issues with notation.

The key change that we need to make lies in Claim 5.7. We no longer have that the chain polynomials correctly decode x_u for every $x \in C$. In fact, we can see that, by the "chain decoder" interpretation of the adaptive chains given in Section 2.3, the chain polynomial computes the advantage of the chain decoder $\mathrm{Dec}^x_{r+1}(u)$ when decoding x_u, namely $\mathbb{E}[x_u \mathrm{Dec}^x_{r+1}(u)]$, where the expectation is over the internal randomness of the chain decoder. In this case, by union bound, the chain decoder is correct with probability at least $1-(r+1)\varepsilon$, and so $\mathbb{E}[x_u \mathrm{Dec}^x_{r+1}(u)] \geq 1 - 2(r+1)\varepsilon$.

Now, when we use Lemma 5.9 to refute the chain polynomial instances, we set parameters as follows. Let $\eta > 0$ to be chosen later, and set r_0 be such that $r_0 + 1 = \lfloor \frac{1}{2\varepsilon} - \eta \rfloor$ and r_1 be such that $r_1 = \gamma\sqrt{\log_2 n}$ for some constant γ to be chosen later. We then let $r = \min(r_0, r_1)$. Note that by choice of r, $\frac{1}{2\varepsilon} - \eta \geq r + 1$, and so $1 - 2(r+1)\varepsilon \geq 2\eta\varepsilon$.

Now, we set d to be such that $d^{r+1} \geq n$, so we have to set $d = n^{1/r+1}$. Finally, we set $\ell = dr/\delta$. Following the calculations, we thus get that either

$$\eta^2\varepsilon^2 k \leq r^2 2^{O(r)} O(\ell r \log n) = \frac{O(1)}{\delta} r^6 2^{O(r)} d \log n\,,$$

or

$$\eta^4 eps^4 k \leq \frac{r^6}{\delta^2} 2^{O(r)} O(\ell r \log n) = \frac{r^8}{\delta^3} 2^{O(r)} d \log n\,.$$

The second equation is always the dominant term. If $\varepsilon \leq \gamma/\sqrt{\log_2 n}$, then we observe that we are simply in the same parameter regime as in the perfect completeness, and we get the same bound. If $\varepsilon \geq 2\gamma/\sqrt{\log_2 n}$, then we have that

$$k \leq \frac{r^8}{\delta^3\eta^4\varepsilon^4} 2^{O(r)} n^{\frac{1}{r+1}} \log n\,.$$

Taking γ large enough and $\eta = O(1/\log n)$ implies that $(\delta^3\varepsilon^4 k) \leq \tilde{O}(n^{\frac{1}{r+1}})$. This finishes the proof, as $r + 1 = \lfloor \frac{1}{2\varepsilon} - \eta \rfloor$. Note that the final $\log(1/\delta)$ loss comes from Fact 3.4.

B Design 3-LCCs over $\mathbb{F}_2$ from Reed–Muller Codes

In this appendix, we give a simple folklore construction of a design 3-LCCs (Definition 3.8) using Reed–Muller codes.

Lemma B.1 (Design 3-LCCs over $\mathbb{F}_2$ from Reed–Muller Codes)**.** *Let t be an integer, and let $k = 1 + t + \binom{t}{2}$. Then, there is a design 3-LCC with blocklength $n = 4^t$ of dimension k. In particular, $n \leq 2^{2\sqrt{2k}}$.*

To prove this lemma, we will need the following fact about polynomials over $\mathbb{F}_4$.

Fact B.2. *Let $f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$ be a degree-2 polynomial over $\mathbb{F}_4$. Then, $\sum_{\beta\in\mathbb{F}_4} f(\beta) = 0$.*

Proof. Recall that the field $\mathbb{F}_4$ is equivalent to the polynomial ring $\mathbb{F}_2[\beta]$ modulo the equation $\beta^2 + \beta + 1 = 0$. We have

$$\begin{aligned}
&f(0) = \alpha_0\\
&f(1) = \alpha_0 + \alpha_1 + \alpha_2\\
&f(\beta) = \alpha_0 + \alpha_1\beta + \alpha_2\beta^2\\
&f(1+\beta) = \alpha_0 + \alpha_1(1+\beta) + \alpha_2(1+\beta)^2\\
&\implies f(0) + f(1) + f(\beta) + f(1+\beta) = \alpha_0 \cdot 4 + \alpha_1 \cdot 2(1+\beta) + \alpha_2(1 + \beta^2 + (1 + 2\beta + \beta^2))\\
&= 0\,,
\end{aligned}$$

as $2 = 0$ in $\mathbb{F}_4$. □

Proof of Lemma B.1. We will define the code in two stages. First, we will define, via an encoding map, a code over $\mathbb{F}_4$ with the desired dimension argue that it is a design 3-LCC. Then, we will use this code to construct a code over $\mathbb{F}_2$.

Let $\mathcal{V}$ denote the vector space of degree ≤ 2 polynomials over $\mathbb{F}_4$ in t variables $x_1, \dots, x_t$. We note that $\mathcal{V}$ has dimension k.

For each $b \in \mathbb{F}_4^k$, we encode b by (1) letting $f_b(x_1, \dots, x_t)$ be the degree-2 polynomial with coefficients given by b, and (2) evaluating f_b over all $x \in \mathbb{F}_4^t$; this yields an output $Z \in \mathbb{F}_4{}^{4^t} = \mathbb{F}_4^n$, which is the encoding $\mathrm{Enc}(b)$. We note that Enc is clearly an $\mathbb{F}_4$ linear map.

We now argue that this encoding map is a design 3-LCC. Indeed, we need to define a system of constraints such that for every pair $x^{(0)}, x^{(1)} \in \mathbb{F}_4^t$, there is a unique constraint containing $x^{(0)}, x^{(1)}$. Let $x^{(\beta)} = x^{(0)} + \beta(x^{(1)} - x^{(0)})$ and $x^{(1+\beta)} = x^{(0)} + (1+\beta)(x^{(1)} - x^{(0)})$. We note that $x^{(0)}, x^{(1)}, x^{(\beta)}$ and $x^{(1+\beta)}$ is the line $L(t) = x^{(0)} + \lambda(x^{(1)} - x^{(0)})$ containing $x^{(0)}, x^{(1)}$. Fix $b \in \mathbb{F}_4^k$, and let f_b be the corresponding polynomial. We know that $g(\lambda) = f_b(L(\lambda))$ is a degree-2 univariate polynomial in λ. Hence, by Fact B.2, it follows that $f_b(x^{(0)}) + f_b(x^{(1)}) + f_b(x^{(\beta)}) + f_b(x^{(1+\beta)}) = 0$. Hence, for each pair $x^{(0)}, x^{(1)} \in \mathbb{F}_4^t$, there exists a constraint containing this pair, and moreover, because two points determine a line, any constraint containing this pair must be exactly this line. Thus, the code given by Enc is a design 3-LCC.

We now use the above code to construct a binary code. Let $\mathrm{Tr} \colon \mathbb{F}_4 \to \mathbb{F}_2$ be the trace map. We let $\mathcal{V}'$ be the image of $\mathcal{V}$ under Tr (applied element-wise to each vector in $\mathcal{V}$). We note that because $\mathcal{V}$ has dimension k over $\mathbb{F}_4$ is a linear code, it is systematic, meaning that there is a subset $S \subseteq \mathbb{F}_4^t$ such that $\mathcal{V}|_S = \mathbb{F}_4^k$. Therefore, because the trace map is identity on $\mathbb{F}_2$, it follows that $\mathcal{V}'|_S = \mathbb{F}_2^k$, i.e., that $\mathcal{V}'$ has dimension k also.

To finish the proof, we need to argue that $\mathcal{V}'$ is a design 3-LCC. Let $g \in \mathcal{V}'$. We will show that for each line $x^{(0)}, x^{(1)}, x^{(\beta)}, x^{(1+\beta)}$ in $\mathbb{F}_4^t$ as defined earlier, it holds that $g(x^{(0)}) + g(x^{(1)}) + g(x^{(\beta)}) + g(x^{(1+\beta)}) = 0$. Indeed, we have that $g = \mathrm{Tr}(f)$ for some $f \in \mathcal{V}$, and that $f(x^{(0)}) + f(x^{(1)}) + f(x^{(\beta)}) + f(x^{(1+\beta)}) = 0$. Because all the coefficients in the linear constraint are 1, i.e., they are in $\mathbb{F}_2$, the constraint still holds after applying $\mathrm{Tr}(\cdot)$, as this is an $\mathbb{F}_2$-linear map. Thus, the constraint holds, which finishes the proof. □