Low Degree Local Correction Over the Boolean Cube

Prashanth Amireddy* Amik Raj Behera† Manaswi Paraashar‡
Srikanth Srinivasan § Madhu Sudan¶

November 14, 2024

Abstract

In this work, we show that the class of multivariate degree-d polynomials mapping $\{0,1\}^n$ to any Abelian group G is locally correctable with $\widetilde{\mathcal{O}}_d((\log n)^d)$ queries for up to a fraction of errors approaching half the minimum distance of the underlying code. In particular, this result holds even for polynomials over the reals or the rationals, special cases that were previously not known. Further, we show that they are locally list correctable up to a fraction of errors approaching the minimum distance of the code. These results build on and extend the prior work of the authors [ABP+24] (STOC 2024) who considered the case of linear polynomials ($d = 1$) and gave analogous results.

Low-degree polynomials over the Boolean cube $\{0,1\}^n$ arise naturally in Boolean circuit complexity and learning theory, and our work furthers the study of their coding-theoretic properties. Extending the results of [ABP+24] from linear polynomials to higher-degree polynomials involves several new challenges and handling them gives us further insights into properties of low-degree polynomials over the Boolean cube. For local correction, we construct a set of points in the Boolean cube that lie between two exponentially close parallel hyperplanes and is moreover an interpolating set for degree-d polynomials. To show that the class of degree-d polynomials is list decodable up to the minimum distance, we stitch together results on anti-concentration of low-degree polynomials, the Sunflower lemma, and the Footprint bound for counting common zeroes of polynomials. Analyzing the local list corrector of [ABP+24] for higher degree polynomials involves understanding random restrictions of non-zero degree-d polynomials on a Hamming slice. In particular, we show that a simple random restriction process for reducing the dimension of the Boolean cube is a suitably good sampler for Hamming slices. Thus our

*School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award and NSF Award CCF 2152413 to Madhu Sudan and a Simons Investigator Award to Salil Vadhan. Email: pamireddy@g.harvard.edu

†Department of Computer Science, University of Copenhagen, Denmark. Supported by Srikanth Srinivasan's start-up grant from the University of Copenhagen. The work was begun when the author was a student at the Department of Computer Science, Aarhus University, Denmark and was supported by Srikanth Srinivasan's start-up grant from the Aarhus University. Email: ambe@di.ku.dk

‡Department of Mathematical Sciences, University of Copenhagen, Denmark. Supported by the European Union under the Grant Agreement No 101078107, QInteract. Email: manaswi.isi@gmail.com

§Department of Computer Science, University of Copenhagen, Denmark. Also partially employed by Aarhus University, Denmark. This work was funded by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: srsr@di.ku.dk

¶School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award and NSF Award CCF 2152413. Email: madhu@cs.harvard.edu

exploration unearths several new techniques that are useful in understanding the combinatorial structure of low-degree polynomials over $\{0,1\}^n$.

Contents

1 Introduction

In this paper, we consider the local correction of low-degree polynomial functions over groups evaluated over $\{0,1\}^n$ and give polylogarithmic query local correctors for every constant degree. This extends and generalizes previous work of the authors [ABP+24] who considered and solved the analogous problem for the linear (i.e., $d = 1$) case. We define some of the basic terms and review the previous work before describing the challenges in strengthening to higher degrees and the new tools used to overcome them.

Low degree polynomials over groups. The main objects of interest in this paper are polynomial functions mapping $\{0,1\}^n$ to an Abelian group G. Here a function f is a polynomial of degree at most d if it can be expressed as $\sum_{S\subseteq[n]:|S|\leqslant d} c_S \prod_{i\in S} x_i$, where the product is over the integers and the coefficients c_S come from the Abelian group G. We denote the space of polynomial functions of degree at most d by $\mathcal{P}_d(\{0,1\}^n, G)$ (which we compress to $\mathcal{P}_d$ when G and n are known). The standard proof of the Ore-DeMillo-Lipton-Schwartz-Zippel lemma naturally extends to polynomials over groups. It proves that two different degree d polynomials disagree on at least $\delta_d := 2^{-d}$ fraction of the domain (if $d < n$), and thus form natural classes of error-correcting codes. This paper explores the corresponding correction questions focusing on locality.

A special case that is already of interest is when the group G is the group of real numbers (or rationals) - a setting where relatively few codes are shown to exhibit local correction properties.

Local correction of polynomials. Informally, the local correction problem is that of computing, given oracle access to a function $f : \{0,1\}^n \to G$ and a point $\mathbf{a} \in \{0,1\}^n$, the value $P(\mathbf{a})$ of the nearest degree d polynomial P to the function f at the point $\mathbf{a}$, while making few oracle queries to f. More formally, for functions $f, g : \{0,1\}^n \to G$, let $\delta(f,g)$ denote the fraction of points from the domain where they differ. We say f is ε-close to g if $\delta(f,g) \leqslant \varepsilon$ and ε-far otherwise. For a given G, we say that that $\mathcal{P}_d$ is (δ, q)-locally correctable if for every n there is a probabilistic algorithm that, for every function $f : \{0,1\}^n \to G$ that is $\delta = \delta(n)$-close to some polynomial $P \in \mathcal{P}_d(\{0,1\}^n, G)$ and for every $\mathbf{a} \in \{0,1\}^n$, outputs $P(\mathbf{a})$ with probability at least $3/4$ while making at most $q = q(n)$ queries to f.

One of the main quests of this work is to give non-trivial upper bounds on the query complexity q for which $\mathcal{P}_d$ is $(\Omega_d(1), q)$-locally correctable.

List correction of codes. Note that (δ, q)-locally correctability of $\mathcal{P}_d$ requires that δ is less than half the minimum distance of the space, i.e., $\delta < \delta_d/2$. To go beyond one usually resorts to the notion of list-decoding; and in the local setting, to notions like "local list-decoding" and "local list correction". Roughly list-decoding allows the decoder to output a small list of words with the guarantee that all codewords within a given distance are included in the output. Formally we say $\mathcal{P}_d$ is (combinatorially) (δ, L)-list correctable if for every $f : \{0,1\}^n \to G$ there are at most L degree-d polynomials P satisfying $\delta(f,P) \leqslant \delta$.[1] Unlike the unique decoding problem where the maximum δ such that a code is uniquely correctable up to δ errors is well understood, the list-decoding radius for higher values of L is not well-understood. A natural question that we study here (for the first

[1] The algorithmic version would require the list of nearby polynomials to be algorithmically recoverable from f. We don't consider this notion in this work but move ahead to the harder "local list correction" problem.

time in this generality) is: *What is the largest δ such that $\mathcal{P}_d$ is $(\delta, \mathcal{O}_d(1))$-list correctable?* We refer to this largest value of δ as the list-decoding radius of $\mathcal{P}_d$.

Local list correction of codes. Local list correction is the notion of list decoding combined with the notion of local correction. Formalizing this definition is a bit more subtle and was first done in [STV01]. The notion allows the decoder to work in two phases — a preprocessing phase with $q_1 = q_1(n)$ queries to the function f, that outputs up to L algorithms $\phi_1, \ldots, \phi_L$ and a query phase, where given $\mathbf{a} \in \{0,1\}^n$ each algorithm ϕ_i makes $q_2 = q_2(n)$ queries to f and outputs $\phi_i(\mathbf{a})$. We say that $\mathcal{P}_d$ is (δ, q_1, q_2, L)-local list correctable if for every function f and polynomial $P \in \mathcal{P}_d$ that are δ-close, there is a decoder as above such that one of its outputs includes P with high probability (say 3/4). See Definition 2.2.3 for a formal definition. The final goal of this paper is to locally list-correct $\mathcal{P}_d$ using non-trivially small number of queries (in both the preprocessing and query phases) where the fraction of errors approaches the list-decoding radius.

1.1 Motivation and previous work

Local decoding of polynomials over finite fields has played a central role in computational complexity and in particular in breakthrough results like IP=PSPACE and the PCP theorem. While most of these results consider functions over the entire multivariate domain (i.e., $\mathbb{F}^n$), low-degree polynomials over $\{0,1\}^n$ do arise quite naturally in complexity theory, notably in circuit complexity capturing classes like AC^0 [Raz87, Smo87] and ACC [BHLR19], and in learning theory. Many of these results exploit basic distance properties of multivariate polynomials as given by the Ore-DeMillo-Lipton-Schwartz-Zippel lemma (see Theorem 2.2.1). This lemma roughly says that the space of degree-d polynomial functions mapping S^n to a field $\mathbb{F}$ where $S \subseteq \mathbb{F}$ is finite form an error-correcting code of relative distance $d/|S|$ when $d < |S|$, and $|S|^{-d/(|S|-1)}$ when $d \geqslant |S|$.

The special case of $S = \mathbb{F}$ is extensively studied and heavily used, e,g., in PCPs and cryptography. In this setting, the lemma can also be made algorithmic, with the first such instance handling the special case of $\mathbb{F}_2$ dating back to the works of Reed and Muller [Ree54, Mul54]. More recently, local list correction algorithms were discovered in the works of Goldreich and Levin [GL89] for linear polynomials, Sudan, Trevisan and Vadhan [STV01] for higher-degree polynomials over large fields, Gopalan, Klivans and Zuckerman [GKZ08] for higher-degree polynomials over $\mathbb{F}_2$, and Bhowmick and Lovett [BL18] for polynomials over any small finite field.

The case of general S however has not received much attention and is mostly unexplored. This was first highlighted in a relatively recent work of Kim and Kopparty [KK17] who gave polynomial time (but not local) unique-decoding algorithms correcting errors up to half the minimum distance. Their work exposed the fact that many other algorithmic and even some coding-theoretic questions were not well understood when $S \neq \mathbb{F}$, and our work aims to fill some gaps in knowledge here.

Another motivation for our work is the design of locally correctable codes over the reals. A series of works [BDYW11, DSW14, DSW17] has exposed that there are no known $(\Omega(1), \mathcal{O}(1))$-locally correctable codes over the reals of arbitrarily large dimensions. The underlying challenge here leads to novel questions in incidence geometry. Roughly the goal here is to design a finite set of points $T \subseteq \mathbb{R}^n$ such many pairs of points in T are contained in constant-dimensional "degenerate" subspaces, where a q-dimensional subspace is said to be degenerate if it contains $q+1$ points from T. Till recently no sets that possessed this property with $q = o(n)$ were known, and the recent results of [ABP$^+$24] may be viewed as showing that the set $T = \{0,1\}^n \subseteq \mathbb{R}^n$ has $\widetilde{O}(\log n)$ dimensional subspaces covering most pairs of points of T. Local correctability of degree d polynomials would translate to showing that moment vectors[2] of $\{0,1\}^n$ (viewed as vectors in $\mathbb{R}^N$ for $N = \mathcal{O}(n^d)$) also exhibit a similar property, thus adding to the body of sets in $\mathbb{R}^N$ that show non-trivial degeneracies.

Turning to previous work on local correction of polynomials over grids, the local correction question when $S = \{0,1\}$ was first explored by Bafna, Srinivasan and Sudan [BSS20], who mainly presented a lower bound of $\widetilde{\Omega}(\log n)$ on the number of queries to recover even when $d = 1$ and from some $o(1)$ fraction of errors and $\mathbb{F} = \mathbb{R}$. On the positive side, for fields of characteristic p, they gave an $\mathcal{O}_{d,p}(1)$ query algorithm to recover from $\Omega_{d,p}(1)$ fraction of errors. This left the case for large and zero characteristic fields open.

The recent work of the authors [ABP$^+$24] investigated the case of general fields, and more generally, polynomials over Abelian groups (i.e., $\{0,1\}^n \to G$), for the special case of $d = 1$. For this setting, they consider all three questions posed in the previous section, namely the (unique) local correction limit, the list-decoding radius, and the local list correction problem. In this setting where distinct degree 1 polynomials disagree with each other on at least half the domain, they show that up to $1/4$ fraction of errors can be uniquely locally corrected with $\widetilde{\mathcal{O}}(\log n)$ queries. They further show that the list-decoding radius approaches $1/2$ and that there are $\widetilde{\mathcal{O}}(\log n)$ query algorithms to locally list correct $\mathcal{P}_1$ for any fraction of errors bounded away from $1/2$. Their work exposes a number of technical challenges in going beyond the $d = 1$ case and we address those in this paper.

1.2 Technical challenges in extending beyond the linear case

We start by reviewing the main ideas in [ABP$^+$24] and outlining the challenges in the higher-degree extension. Their unique local corrector correcting up to half the minimum distance works in three steps: Given an oracle for a function $f(\mathbf{x})$ at a distance less than $1/4$ from a linear polynomial $P(\mathbf{x})$,

- It first provides oracle access to a function $f_1(\mathbf{x})$ at any tiny but constant distance from $P(\mathbf{x})$, using $\mathcal{O}(1)$ queries to $f(\mathbf{x})$.
- Then the algorithm provides oracle access to a function $f_2(\mathbf{x})$ at distance $1/\operatorname{poly}(\log n)$ from $P(\mathbf{x})$ while making $\operatorname{poly}(\log\log n)$ queries to $f_1(\mathbf{x})$.
- Finally, the algorithm provides oracle access to the linear polynomial $P(\mathbf{x})$ making $\mathcal{O}(\log n)$ queries to f_2. Composing the three steps gives the desired unique local corrector.

The first two steps in their result are general enough to work for all degrees. The third step in the unique local corrector of [ABP$^+$24] is the most significant one and does not extend immediately to the higher degree setting. [ABP$^+$24] reduce this step to show that any point in $\{0,1\}^n$ can be expressed as a linear combination of $\widetilde{\mathcal{O}}(\log n)$ roughly balanced vectors[3]. To extend their approach to higher degrees, we would need an analogous result for the degree $\leqslant d$-moment vectors of vectors in $\{0,1\}^n$ but we are unable to find such an extension directly. Instead, as we elaborate further

[2] d-moment vector of $\mathbf{v} \in \{0,1\}^n$ is a vector in $\{0,1\}^{\binom{n}{d}}$ which is an evaluation vector of $\mathbf{v}$ on all multilinear monomials of degree $\leqslant d$.

[3] Hamming weight is very close to $n/2$

below, we manage to find a new path for this step, which results in a somewhat different proof even for the linear ($d = 1$) case.

Next, we turn to the combinatorial analysis of the list-decoding radius. For simplicity assume that the Abelian group G is a finite field $\mathbb{F}_p$. The analysis in the linear case [ABP$^+$24] splits into two cases: the low[4] characteristic case ($p \leqslant 3$) and the high characteristic case ($p > 5$). The former case is handled via a suitable version of the Johnson bound. In the latter setting, a key insight used in [ABP$^+$24] for the linear case is that non-sparse linear polynomials tend to be non-zero with very high probability, i.e. *anti-concentration* of non-sparse linear polynomials. In the higher degree setting, both analyses become much more involved. In the low characteristic case, the Johnson bound no longer yields the right answer. For the high characteristic case, the primary obstacle is understanding the anti-concentration statement for non-sparse low-degree polynomials.

Generalizing the result above to all Abelian groups involves multiple steps in [ABP$^+$24] - they extend the latter approach above to groups where every element has a sufficiently high order (specifically order at least 5). Then they consider groups where every element has order a power of 2 or 3 separately and analyze these special cases; and finally use some "special intersection properties" of the agreement sets[5] of different polynomials with any given function to apply a counting tool from the work of Dinur, Grigorescu, Kopparty, and Sudan [DGKS08a] to combine these different steps. While many of the steps extend to the higher degree setting (sometimes needing heavier machinery) the final step involving "special intersection properties" simply does not work in our setting. (Roughly the difference emanates from the fact that the probability that two linearly independent degree-1 functions vanish at a random point is at most $1/4$, which is the square of the probability for a single degree-1 function. This fails for degree 2.) Overcoming this barrier leads to further new challenges in the higher-degree case.

The local list-correction algorithm for degree-1 polynomials of [ABP$^+$24] is inspired by the local list-correction algorithm of Reed-Muller codes from [STV01]. The high-level idea is to start with a function $f(\mathbf{x})$ that is $(1/2 - \varepsilon)$-close to a set List of linear polynomials and produce a small list of oracles such that each polynomial in List is within a small constant distance to an oracle from the list, at which point the unique local corrector becomes applicable. This 'error-reduction' step involves choosing a random subcube C of $\{0,1\}^n$ (as defined in Section 2 below) of sufficiently large but constant dimension k and doing a brute-force list decoding on C to find a list List': it is not hard to argue that the restricted version P' of each polynomial $P \in \mathsf{List}$ appears in the list List' with high probability (this needs the combinatorial bound mentioned above). To complete the error-reduction, they need to decode P at a random point $\mathbf{b} \in \{0,1\}^n$. This is done by repeating the above brute-force algorithm with the subcube $\mathsf{C}^{\mathbf{b}}$ 'spanned' by C and $\mathbf{b}$: informally, this is the smallest subcube spanned by C and $\mathbf{b}$ and has dimension $2k$ (see Section 5.2.1 for details.) They now obtain a new list of polynomials List'' and need to isolate the polynomial P'' corresponding to P in this list to get $P(\mathbf{b})$. Here, [ABP$^+$24] uses the fact that we know the restriction of P to the subcube C inside $\mathsf{C}^{\mathbf{b}}$. The bad event in this case is that there are two polynomials in $\mathcal{L}''$ that disagree on $\mathbf{b}$ but agree on C. Bounding the probability of this event is the key step in the analysis of [ABP$^+$24] and is done by giving a complete understanding of which kinds of distinct polynomials over $\mathsf{C}^{\mathbf{b}}$ can collapse to the same polynomial over C. This kind of understanding seems difficult to obtain for higher degrees, and we need to develop a new analysis for bounding the probability of

[4] The precise constants here are not important, as the analysis works as long as the $p \leqslant \mathcal{O}(1)$.

[5] Agreement set of a polynomial P and a function f is defined as the subset of $\{0,1\}^n$ on which P and f agree.

the bad event in this setting.

We now turn to our results before elaborating on the techniques used to overcome the challenges.

1.3 Our main results

Briefly, our results provide poly-logarithmic query algorithms for unique and list-decoding to the maximal fraction of errors that are allowed. Specifically, the unique decoding algorithm works up to half the distance. We also establish that the list-decoding radius approaches the distance (as the list size tends to infinity) and give matching local algorithms. We give our specific theorems below.

Theorem 1.3.1 (Local correction algorithms for $\mathcal{P}_d$ up to the unique decoding radius)**.** *For every Abelian group G and for every constant $\varepsilon > 0$, the space $\mathcal{P}_d$ has a (δ, q)-local correction algorithm where $\delta = \frac{1}{2^{d+1}} - \varepsilon$ and $q = \widetilde{\mathcal{O}}_\varepsilon(\log n)^d$.*

We show that if all its elements of the group have a constant order, then the query complexity of the local correction algorithm can be brought down from $\widetilde{\mathcal{O}}_d((\log n)^d)$ to a constant (i.e., independent of n). Specifically, we say that an Abelian group G is a *torsion group* if all its elements have finite order, and the *exponent* of a torsion group is the least common multiple of the orders of all the elements. While [BSS20] shows this only for groups underlying fields of constant characteristic and for *some* constant error, we extend their proof to all groups of constant exponent and error up to the unique-decoding radius.

Theorem 1.3.2. *If G is an Abelian torsion group of exponent M, then for every $\varepsilon > 0$, $\mathcal{P}_d$ has a (δ, q)-local correction algorithm where $\delta = \frac{1}{2^{d+1}} - \varepsilon$ and $q = \mathcal{O}_{M,\varepsilon}(1)$.*

As noted earlier, some dependence on n is needed even when the degree is 1 and G is a field of large characteristic (or characteristic 0), as an $\Omega(\log n / \log\log n)$ lower bound was shown in this setting by earlier work of Bafna, Srinivasan, and Sudan [BSS20] (and shown to be tight up to $\mathrm{poly}(\log\log n)$ factors in [ABP$^+$24]). Our upper bound above is thus optimal to within polynomial factors (for constant d). However, we do not know if the query complexity can be improved to, say, $\widetilde{\mathcal{O}}_d(\log n)$ for degree d.

We also extend the above algorithm from Theorem 1.3.1 to the list decoding regime. For this, we first establish a bound on the list-decoding radius. As far as we know, the following result was not known before for G being any group other than the field $\mathbb{F}_2$.

Theorem 1.3.3 (Combinatorial list decoding bound for $\mathcal{P}_d$)**.** *For every Abelian group G and for every constant $\varepsilon > 0$, the space $\mathcal{P}_d$ over any Abelian group G is $(1/2^d - \varepsilon, \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$-list correctable.*

The above is tight in the sense that the number of codewords at distance $1/2^d$ can depend on both n and the size of the group G (and is infinite when G is infinite). We do not know if the dependence on ε is tight. Note that for the setting where $d = 1$, [ABP$^+$24] gives a polynomial bound in terms of $1/\varepsilon$. Our bound as stated above is exponential and while we can see a path to improving this to a quasi-polynomial, we don't see a polynomial upper bound using the proofs of this paper even when $d = 1$.

Finally, we state our local list correction result.

Theorem 1.3.4 (Local list correction for $\mathcal{P}_d$)**.** *For every Abelian group G and for every $\varepsilon > 0$, the space $\mathcal{P}_d$ is $(1/2^d - \varepsilon, \mathcal{O}_\varepsilon(1), \widetilde{\mathcal{O}}_\varepsilon(\log n)^d, \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)}))$-locally list correctable.*

Specifically, there is a randomized algorithm $\mathcal{A}$ that, when given oracle access to a polynomial f and a parameter $\varepsilon > 0$, outputs with probability at least $3/4$ a list of randomized algorithms $\phi_1, \ldots, \phi_L$ ($L \leqslant \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$) such that the following holds. For each $P \in \mathcal{P}_d$ that is $(1/2^d - \varepsilon)$-close to f, there is at least one algorithm ϕ_i that, when given oracle access to f, computes P correctly on every input with probability at least $3/4$.

The algorithm $\mathcal{A}$ makes $\mathcal{O}_\varepsilon(1)$ queries to f, while each ϕ_i makes $\widetilde{\mathcal{O}}_\varepsilon((\log n)^d)$ queries to f.

Remark 1.3.5. *If G is an Abelian torsion group of exponent M, $\mathcal{P}_d$ is $(1/2^d - \varepsilon, \mathcal{O}_\varepsilon(1)$, $\mathcal{O}_{M,\varepsilon}(1)$, $\exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)}))$-locally list-correctable. This follows in a similar manner as Theorem 1.3.4, except we replace the generic local corrector in the unique-decoding regime with that given by Theorem 1.3.2.*

1.4 Technical tools

In the process of proving our main results, we prove several lemmas that we believe are independently interesting.

In the proof of Theorem 1.3.1, the main step is to construct, for any $\mathbf{a} \in \{0,1\}^n$, a distribution $\mathcal{D}_\mathbf{a}$ over $(\{0,1\}^n)^q$ such that the marginal distribution of each point is close to the uniform distribution over $\{0,1\}^n$, and for any degree-d polynomial P, $P(\mathbf{a})$ can be computed via the evaluations of P on a sample from $\mathcal{D}_\mathbf{a}$. Constructing such a distribution $\mathcal{D}$ reduces to the following problem of constructing a geometric set with some nice algebraic properties. We discuss in Section 3 how such a set leads to the distribution $\mathcal{D}_\mathbf{a}$.

Question 1. *Find two parallel hyperplanes in k dimensions that are ε-close in Euclidean distance such that every non-zero degree-d multilinear polynomial is non-zero on the points of the Boolean cube $\{0,1\}^k$ lying between the two hyperplanes.*

The 'closeness' parameter ε plays a crucial role in the efficiency of the local correction algorithm. It is easy to see (and folklore) that we can get $\varepsilon = 1/k^{\Omega(1)}$. However, we show that we can obtain a construction where ε is *exponentially small* in k. This is done in Lemma 3.2.1.

The proof of the combinatorial list decoding bound is broken down into two cases depending on the order of elements in the group. The first case is when all elements have order larger than a prime $p_0(d)$ (a constant dependent on d) and the second case is when the group is a product of p-groups for $p < p_0$. For the first case, the key step is an understanding of the anti-concentration properties of low-degree non-sparse polynomials. More precisely, we have the following question.

Question 2. *Let $P(x_1, \ldots, x_n)$ be a polynomial of degree d with at least s non-zero monomials. Then how large can $\Pr_{\mathbf{a} \sim \{0,1\}^n}[P(\mathbf{a}) = 0]$ be?*

If we take $P = x_1 x_2 \cdots x_{d-1} \cdot L(x_d, \ldots, x_n)$ where L is a linear polynomial with s monomials, we see that P vanishes with probability approximately $1 - 2^{-(d-1)} - \mathcal{O}(1)/\sqrt{s}$ over (say) the reals. In Lemma 4.1.1, we build on known anti-concentration results [Erd45, MNV16] show that this is essentially the worst possible in groups with no elements of small order.

In the second case, an important step in our proof is a tail inequality for events defined by the vanishing of low-degree polynomials over a field. Let $Q_1(\mathbf{x}), \ldots, Q_t(\mathbf{x})$ be t degree-d polynomials on the same variable set. We know that each is non-zero on a random input with probability at least 2^{-d} and hence that the expected number of polynomials vanishing at a uniformly random point $\mathbf{a} \sim \{0,1\}^n$ is at most $(1 - 2^{-d}) \cdot t$. This leads to the following question.

Question 3. *Suppose we have a collection of t degree-d polynomials. Can we bound the probability that more than $(1 - 2^{-d} + \varepsilon) \cdot t$ many of these polynomials vanish at a random point $\mathbf{a}$ in $\{0,1\}^n$?*

Clearly, we cannot get a strong tail bound unless we impose some 'independence' constraints on the polynomials (for example, we cannot hope for a strong bound if all the polynomials are in the linear span of a small number of polynomials). We show that we can get a Chernoff-style tail bound under the constraint that the 'leading monomials' (under a suitable ordering) of these polynomials are pairwise disjoint. This is done in Lemma 4.2.1 using the 'Footprint bound' [GH00] (essentially a tool from commutative algebra) and an idea due to Panconesi and Srinivasan [PS97].

We build on this result to prove an optimal bound on the list-decoding radius for degree-d polynomials over small finite fields $\mathbb{F}_p$ (and more generally over groups that are products of p-groups, where $p < p_0$). In the setting when we are working with polynomials mapping $\mathbb{F}_p^n$ to $\mathbb{F}_p$, this was done in the works of [GKZ08, BL18] via an involved mixture of algebraic and analytic techniques. Unfortunately, these do not seem to be applicable here: one significant reason that appears again and again in our work is that we cannot restrict a given function to an arbitrary subspace in our ambient space since the domain $\{0,1\}^n$ does not have this algebraic structure. Instead we use other combinatorial techniques such as the Sunflower lemma in conjunction with the above tail bound to obtain the stated result.

For local list correction, we follow the algorithm of [ABP+24] modulo changes in parameters to handle higher degree polynomials. The main innovation involves analyzing the behavior of degree-d polynomials on a Hamming slice (points with a fixed Hamming weight) after a random process of reducing the dimension. In particular, assume that we have a fixed degree-d polynomial $R(x_1, \ldots, x_{2k})$ in $2k$ dimensions such that R is non-zero on the Hamming slice of weight k. We now choose a random subcube C by pairing the $2k$ variables at random into k pairs and identifying the variables in each pair. We would like to upper bound the probability that R is zero on the cube C by a function that goes to 0 with k.[6] We do this by addressing the following two questions.

The first question is on how the density of any fixed set on a slice changes under the aforementioned random process. In this work, we are particularly interested in the middle slice, i.e. points of Hamming weight k in $\{0,1\}^{2k}$.

Question 4. *For any fixed subset S of the middle slice, how does the density of the set $S \cap \mathsf{C}$ (as a subset of the middle slice of C) compare with the density of S?*

[6] One might hope to prove such a statement under the weaker assumption that R is simply a non-zero multilinear polynomial of degree d. Unfortunately, the simple example $R = x_1 + \cdots + x_{2k}$ over the group $G = \mathbb{F}_2$ shows that such a statement is not possible even in the degree-1 case.

In Lemma 5.1.1 we show that the density is almost preserved under the random process. In other words, this random process is a good sampler for subsets of the middle slice. To prove Lemma 5.1.1, we show that certain kinds of *Johnson graphs* are good spectral expanders.

The second question is on a quantitative estimate of the number of non-zero points of a degree-d polynomial on a Hamming slice. This is a natural question, but has not been addressed before as far as we know.

Question 5. *For a degree-d polynomial R which is non-zero on a Hamming slice, on how many points of the Hamming slice is it non-zero?*

We give a simple lower bound on the number of non-zero points for a degree-d polynomial on the Hamming slice by modifying the proof of the Ore-DeMillo-Lipton-Schwartz-Zippel lemma. We show it in Lemma 5.1.6.

2 Preliminaries

Most of our notation and definitions are identical to [ABP+24].

2.1 Notation

Let $(G, +)$ denote an Abelian group G with addition as the binary operation. For any $g \in G$, let $-g$ denote the inverse of $g \in G$. For any $g \in G$ and integer $a \geqslant 0$, $a \cdot g$ (or simply ag) is the shorthand notation of $\underbrace{g + \ldots + g}_{a \text{ times}}$ and $-ag$ denotes $a \cdot (-g)$. We say that a group is a *p-group* if the order of each element is a power of p. We say that a group is a *torsion group* if all its elements have finite order. The *exponent* of a torsion group is the least common multiple of the orders of all its elements.

For a natural number n, we consider functions $f : \{0,1\}^n \to G$. We denote the set of functions that can be expressed as a multilinear polynomial of degree d, with the coefficients being in G by $\mathcal{P}_d(n, G)$. We will simply write $\mathcal{P}_d$ when n and G are clear from the context. For a polynomial P, we refer to the number of monomials with a non-zero coefficient as the *sparsity* of P and denote it by $\text{spars}(P)$. Similarly we use $\deg(P)$ to denote the degree of P.

For every alphabet set Σ and $\mathbf{x}, \mathbf{y} \in \Sigma^n$, let $\delta(\mathbf{x}, \mathbf{y})$ denote the relative Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, i.e. $\delta(\mathbf{x}, \mathbf{y}) = |\{i \in [n] \mid x_i \neq y_i\}|/n$. For $0 \leqslant m \leqslant n$, let $\{0,1\}^n_m$ denote the set of points in $\{0,1\}^n$ of Hamming weight exactly m.

For any $\mathbf{x} \in \{0,1\}^n$, $|\mathbf{x}|$ denotes the Hamming weight of $\mathbf{x}$. $\tilde{\mathcal{O}}(\cdot)$ notation hides factors that are poly-logarithmic in its argument. For a polynomial $P(\mathbf{x})$, let $\text{vars}(P)$ denote the variables on which P depends, i.e. the variables that appear in a monomial with non-zero coefficient in P.

For any natural number n, U_n denotes the uniform distribution on $\{0,1\}^n$.

2.2 Basic definitions and tools

Probabilistic notions. For any distribution X on $\{0,1\}^n$, let $\text{supp}(X)$ denote the subset of $\{0,1\}^n$ on which X takes non-zero probability. For two distributions X and Y on $\{0,1\}^n$, the statistical distance between X and Y, denoted by $\mathrm{SD}(X,Y)$ is defined as

$$\mathrm{SD}(X,Y) = \max_{T\subseteq\{0,1\}^n} |\Pr[X \in T] - \Pr[Y \in T]|$$

We say X and Y are ε-close if the statistical distance between X and Y is at most ε.

Coding theory notions. Fix an Abelian group G. We use $\mathcal{P}_d$ to denote the space of multilinear polynomials from $\{0,1\}^n$ to G of degree at most d. More precisely, any element $P \in \mathcal{P}_d$ can be described as

$$P(x_1,\ldots,x_n) = \sum_{I\subseteq[n]\,:\,|I|\leqslant d} \alpha_I \prod_{i\in I} x_i$$

where $\alpha_I \in G$ for each I. On an input $\mathbf{a} \in \{0,1\}^n$, each monomial evaluates to a group element in G and the polynomial evaluates to the sum of these group elements.

The following is a standard fact about multilinear polynomials which also holds in the setting when the range is an arbitrary Abelian group G. The proof is standard and omitted.

Theorem 2.2.1 (Basic facts about multilinear polynomials)**.** *Let G be any Abelian group.*

1. *Any function $f : \{0,1\}^n \to G$ has a unique representation as a multilinear polynomial over G. In particular two distinct multilinear polynomials cannot agree on all points in $\{0,1\}^n$.*
2. *(DeMillo-Lipton-Schwartz-Zippel (DLSZ) lemma [DL78, Zip79, Sch80]) More generally, any two distinct multilinear polynomials $P, Q \in \mathcal{P}_d$ differ with probability at least 2^{-d} at a uniformly random input from $\{0,1\}^n$. Equivalently, $\delta(\mathcal{P}_d) \geqslant 2^{-d}$.*

We now turn to the kinds of algorithms we will consider. Below, let $\mathcal{F}$ be any space of functions mapping $\{0,1\}^n$ to G.

Definition 2.2.2 (Local Correction Algorithm)**.** *We say that $\mathcal{F}$ has a (δ, q)-local correction algorithm if there is a probabilistic algorithm that, when given oracle access to a function f that is δ-close to some $P \in \mathcal{F}$, and given as input some $\mathbf{a} \in \{0,1\}^n$, returns $P(\mathbf{a})$ with probability at least $3/4$. Moreover, the algorithm makes at most q queries to its oracle.*

Definition 2.2.3 (Local List-Correction Algorithm)**.** *We say that $\mathcal{F}$ has a (δ, q_1, q_2, L)-local list correction algorithm if there is a randomized algorithm $\mathcal{A}$ that, when given oracle access to a function f, produces a list of randomized algorithms $\phi_1,\ldots,\phi_L$, where each ϕ_i has oracle access to f and have the following property: with probability at least $3/4$, for each codeword P that is δ-close to f, there exists some $i \in [L]$ such that the algorithm ϕ_i computes P with error at most $1/4$, i.e. on any input $\mathbf{a}$, the algorithm ϕ_i outputs $P(\mathbf{a})$ with probability at least $3/4$.*
Moreover, the algorithm $\mathcal{A}$ makes at most q_1 queries to f, while the algorithms $\phi_1,\ldots,\phi_L$ each make at most q_2 queries to f.

Remark 2.2.4. *Our algorithms can all be implemented as standard Boolean circuits with the added ability to manipulate elements of the underlying group G. Specifically, we assume that we can store* *group elements, perform group operations (addition, inverse) and compare two group elements to check if they are equal.*

Definition 2.2.5 (Combinatorial List Decodability)**.** *We say that $\mathcal{F}$ is (δ, L)-list decodable if for any function f, the number of elements of $\mathcal{F}$ that are δ-close to f is at most L.*

Subcubes of $\{0,1\}^n$. It will be instrumental in our algorithms to be able to restrict the given function to a small-dimensional subcube and analyze this restriction. We construct such subcubes by first negating a subset of the variables and then identifying them into a smaller set of variables. More precisely, we have the following definition from [ABP+24].

Definition 2.2.6 (Embedding a smaller cube into $\{0,1\}^n$)**.** *Fix any $k \in \mathbb{N}$ and $k \leqslant n$. Fix a point $\mathbf{a} \in \{0,1\}^n$ and a function $h : [n] \to [k]$. For every $\mathbf{y} \in \{0,1\}^k$, $x(\mathbf{y})$ is defined with respect to $\mathbf{a}$ and h as follows:*

$$x(\mathbf{y})_i = y_{h(i)} \oplus a_i = \begin{cases} a_i, & \text{if } y_{h(i)} = 0 \\ 1 \oplus a_i, & \text{if } y_{h(i)} = 1 \end{cases}$$

$C_{\mathbf{a},h}$ is the subset in $\{0,1\}^n$ consisting of $x(\mathbf{y})$ for every $\mathbf{y} \in \{0,1\}^k$, i.e. $C_{\mathbf{a},h} := \left\{x(\mathbf{y}) \,\middle|\, \mathbf{y} \in \{0,1\}^k\right\}$. In particular, note that this subcube contains the point $\mathbf{a}$, since $x(0^k) = \mathbf{a}$.

Given any polynomial $P(x_1, \ldots, x_n)$ and any subcube $C_{\mathbf{a},h}$ as above, P restricts naturally to a degree-d polynomial $Q(y_1, \ldots, y_k)$ on $C_{\mathbf{a},h}$ obtained by replacing each x_i by $y_{h(i)} \oplus a_i$. We use $P|_{C_{\mathbf{a},h}}$ to denote the polynomial Q.

Random subcubes. Now assume that we choose a subcube $C_{\mathbf{a},h}$ by sampling $\mathbf{a} \sim \{0,1\}^n$ and sampling a random hash function $h : [n] \to [k]$. For every $\mathbf{y} \in \{0,1\}^k$, $x(\mathbf{y})$ is the image of $\mathbf{y}$ in $\{0,1\}^n$ under $\mathbf{a}$ and h and $C_{\mathbf{a},h}$ is the subcube consisting of all 2^k such images. From Definition 2.2.6, we have the following simple observation: For every $\mathbf{y} \in \{0,1\}^k$, distribution of $x(\mathbf{y})$ is the uniform distribution over $\{0,1\}^n$. This is because $\mathbf{a}$ is uniformly distributed over $\{0,1\}^n$.

We use the following sampling lemma for subcubes from [ABP+24] that will be useful at multiple points in the paper.

Lemma 2.2.7 (Sampling lemma for random subcubes)**.** *([ABP+24, Lemma 2.4]) Sample $\mathbf{a}$ and h uniformly at random, and let $\mathsf{C} = C_{\mathbf{a},h}$ be the subcube of dimension k as described in Definition 2.2.6. Fix any $T \subseteq \{0,1\}^n$ and let $\mu := |T|/2^n$. Then, for any $\varepsilon, \eta > 0$*

$$\Pr_{\mathbf{a},h}\left[\left|\frac{|T \cap \mathsf{C}|}{2^k} - \mu\right| \geqslant \varepsilon\right] < \eta$$

as long as $k \geqslant \frac{A}{\varepsilon^4\eta^2} \cdot \log\left(\frac{1}{\varepsilon\eta}\right)$ for a large enough absolute constant $A > 0$.

3 Local correction in the unique decoding regime

In this section, we prove Theorem 1.3.1, i.e. we give a local correction algorithm for degree-d polynomials when the error is less than the unique decoding radius (half the minimum distance). The proof of Theorem 1.3.1 will proceed in two phases:

- We give error reduction algorithms that reduce the error from half the minimum distance to sub-constant.
- We give a local correction algorithm for degree-d polynomials when the error is sub-constant, say less than $\mathcal{O}_d(1/(\log n)^d)$.

The first phase follows from the error reduction algorithm of [ABP+24] and we describe it in Section 3.3. In Section 3.1, we describe the second phase, which is the new result in this work. We start with a proof overview of the second phase.

Proof overview. We describe the proof idea behind our local corrector for degree-d polynomials in the sub-constant error regime. Assume that we have oracle access to $f : \{0,1\}^n \to G$ that is δ-close to an (unknown) polynomial $P \in \mathcal{P}_d(\{0,1\}^n, G)$. For simplicity, let us assume that we want to output the value of P at $\mathbf{a} = 0^n$. The proof for an arbitrary $\mathbf{a}$ is more or less the same, except for a minor change.

The idea (as in other local correction algorithms) is to query f at a set of uniformly distributed (but not independent) points $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(q)}$. Since these points are uniformly distributed, we are likely to obtain $P(\mathbf{u}^{(1)}), \ldots, P(\mathbf{u}^{(q)})$ in this way. If we could use this information to determine $P(0^n)$ for any polynomial P, then we would be done. Indeed, this strategy works when $G = \mathbb{F}_2$ [BLR93, AKK+05]. If we restrict to a random $\mathbb{F}_2$-linear subspace V of dimension $d+1$, then the non-zero points of V are (marginally) uniformly distributed and determine the value of P (which is still degree d restricted to V) at 0^n. Unfortunately, this idea does not make sense for polynomials mapping the Boolean cube to groups other than (vector spaces over) $\mathbb{F}_2$.

An analogous strategy we can employ over any group is to restrict to a random *subcube*. More precisely, we choose a random function $h : [n] \to [k]$ (for $d < k \ll n$) according to some distribution and consider the random subcube $C = C_{0^n,h}$ as defined in Section 2 above (informally, we use the function h to identify co-ordinates in k blocks) and query the function f at points in C. To use the strategy above, we would like to find points $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(q)} \in C$ such that

(a) the points $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(q)} \in C$ are uniformly distributed, and

(b) the values of P at these points determine $P(0^n)$ for an unknown degree-d polynomial P.

These two properties are in tension with each other. To see this, assume that the function h is chosen uniformly at random. In this case, it can be checked that unless k is large (approximately $\sqrt{n}$), the only points in $\{0,1\}^k$ that correspond[7] to uniformly random points in $\{0,1\}^n$ are the *perfectly balanced* points (i.e. the points with an equal number of 0s and 1s). All other points of $\{0,1\}^k$ correspond to points with expected Hamming weight outside the range $[n/2 - n/k, n/2 + n/k]$ and hence do not 'look uniform'. Unfortunately, querying P at the set of perfectly balanced points does not determine $P(0^n)$. Informally, this is because the set of perfectly balanced points lie on a hyperplane not containing 0^n, and hence even a degree-1 polynomial can 'distinguish' between these points and 0^n.[8]

[7] the correspondence maps $\mathbf{y} \in \{0,1\}^k$ to $x(\mathbf{y}) \in C$ as defined in Section 2

[8] This reasoning fails over finite fields of fixed positive characteristic and this failure was used in [BSS20] to devise a local correction algorithm over fixed characteristic via this principle. However, this is true over fields of large characteristic and other groups.

We fix this by choosing h in a non-uniform manner and relaxing criterion (a) to finding *nearly*[9] uniformly distributed points in C. More specifically, assume that we have a probability distribution $\mu = (\mu_1, \ldots, \mu_k)$ over $[k]$, and we sample each $h(i)$ independently according to μ. Again the points in the cube C that are uniformly distributed correspond to points on the hyperplane $\sum_{j\in[k]} \mu_j y_j = 1/2$ in $\{0,1\}^k$. However, we consider the points corresponding to $\mathbf{y}$ satisfying $|\sum_{j\in[k]} \mu_j y_j - 1/2| \leqslant \varepsilon$, i.e. points in between two close-by hyperplanes, i.e. these are the 'nearly-balanced points' under a weighted version of Hamming weight on $\{0,1\}^k$. Standard probabilistic arguments imply that if $\varepsilon \ll 1/\sqrt{n}$, then these correspond to nearly uniformly distributed points in $\{0,1\}^n$ thus satisfying the modified version of criterion (a). Intuitively, getting ε to be so small and meaningful at the same time seems easier when some of the weights $\mu_1, \ldots, \mu_k$ are also similarly small (though not all of them can be since they sum to 1). Standard results about Boolean threshold functions [MTT61, Mur71] show that a hyperplane in k dimensions of this form does not require weights smaller than (approximately) $1/k^k$. This forces $k = \tilde{\Omega}(\log n)$ for this strategy. (More generally, [BSS20] showed a lower bound of $\tilde{\Omega}(\log n)$ queries even for decoding linear polynomials over the Boolean cube and fields of large characteristic.) Indeed, we take $k = \Theta_d(\log n)$, so that this strategy becomes feasible. For this value of k, we will show that we can take $\varepsilon = 1/2^{\Omega_d(k)}$.

The problem of designing μ can now be stated (in a more general form) in geometric language: find two parallel hyperplanes H_1, H_2 in k variables that are at distance $1/2^{\Omega_d(k)}$ (this corresponds to making ε small) such that evaluating a degree-d polynomial P at the points of $\{0,1\}^k$ between H_1 and H_2 allow us to deduce the value of P at all other points of the Boolean cube. A set with the latter property is sometimes called a (degree-d) *interpolating set* in the literature. A standard interpolating set is the set of points of Hamming weight in the range $\{a, \ldots, a+d\}$ for any non-negative integer $a \leqslant k-d$. Unfortunately, the two hyperplanes in this case are at a distance of $\Omega(1/\sqrt{k})$ from each other, which is not good enough in our setting. Indeed, the main technical innovation of this section (Lemma 3.2.1 below) is finding a pair of hyperplanes with this specific property that is exponentially close. We believe that this result is independently interesting.

To do this, we use a pair of carefully chosen Boolean threshold functions that require weights of exponentially different magnitudes to describe. We do this in a way that allows us to prove the interpolating set property via a modified version of the DeMillo-Lipton-Schwartz-Zippel lemma (Lemma 3.2.1 below). To reduce the query complexity of the algorithm, we also need to choose a *small* subset of the nearly-balanced points as defined above that form a small interpolating set. Over a field, an interpolating set of size $\mathcal{O}(k^d)$ follows immediately from a linear algebraic argument. We can get a set $\mathcal{S}$ of similar size $q = \mathcal{O}_d(k^d)$ that works for any Abelian group. We then sample the function h as described above to obtain the corresponding (nearly uniformly distributed) points $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(q)} \in \{0,1\}^n$.

Comparison with [ABP+24]. At a high level, the final construction seems to use similar ideas to an analogous step in the low-error local correction algorithm of [ABP+24], but the technical details are quite different. If we consider the random $n \times q$ matrix A that contains $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(q)}$ as its columns, then in the result of [ABP+24] it is shown how to use a single hyperplane[10] with large coefficients to define the *rows* of the matrix A. Here, each point in between the hyperplanes allows us to sample a distinct *column* of the matrix A.

[9] i.e. statistically close to

[10] in fact the Boolean points on the hyperplane

3.1 Regime of sub-constant error

In this subsection, we give a local correction algorithm for degree d polynomials in the setting of sub-constant error. Formally, we prove the following statement in this subsection.

Theorem 3.1.1 (Local correction for sub-constant error)**.** *Fix a degree parameter $d \in \mathbb{Z}_{>0}$ and an Abelian group G. The space $\mathcal{P}_d(\{0,1\}^n, G)$ has a (δ, q)-local correction algorithm where $q = \mathcal{O}_d((\log n)^d)$ and $\delta = 1/100q$.*

To prove Theorem 3.1.1, we use Theorem 3.1.2 (see below). It roughly says that there exists a set of points such that an arbitrary evaluation of any degree-d polynomial can be computed using evaluations on this set and this set consists of points whose relative "weighted Hamming weights" are very close to 1/2.

Theorem 3.1.2 (Weight balanced interpolating set)**.** *Fix a degree parameter $d \geqslant 0$ and a dimension parameter $k \in \mathbb{Z}_{>0}$ that is divisible by $10(d+1)$. There exists a set $\mathcal{S} \subseteq \{0,1\}^k$ such that for every Abelian group G, $\mathcal{S}$ satisfies the following properties:*

1. *[Interpolating set]. For each point $\mathbf{b} \in \{0,1\}^k$, there exists integral coefficients $c_1, \ldots, c_{|S|}$ such that for every degree-d polynomial $Q(y_1, \ldots, y_k) \in \mathcal{P}_d(\{0,1\}^k, G)$, we have,*

$$Q(\mathbf{b}) \;=\; \sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} Q(\mathbf{u})$$

2. *[Weighted balanced]. There exists positive integers $w_1, \ldots, w_k$ such that*

$$\mathcal{S} \;\subseteq\; \left\{ \mathbf{y} \in \{0,1\}^k \;\middle|\; \left| \sum_{j=1}^{k} w_j y_j - \frac{W}{2} \right| \leqslant \frac{W}{2^{\Omega(k/(d+1))}} \right\},$$

where $W := \sum_{j=1}^{k} w_j$.

Furthermore, $|\mathcal{S}|$ is at most $\mathcal{O}_d(k^d)$.

Before we prove Theorem 3.1.2, let us first see how Theorem 3.1.2 is useful in designing a local correction algorithm and to prove Theorem 3.1.1. Below, we assume Theorem 3.1.2 and prove Theorem 3.1.1.

Proof of Theorem 3.1.1. Fix an input point $\mathbf{a} \in \{0,1\}^n$ to the local correction algorithm. Let $f(x_1, \ldots, x_n) : \{0,1\}^n \to G$ be the input function with $\delta(f, \mathcal{P}_d) \leqslant \delta$. Let $P(\mathbf{x})$ be the unique degree d polynomial such that $\delta(f, P) = \delta$. Our goal is to compute $P(\mathbf{a})$ with probability at least 3/4 using oracle queries to $f(\mathbf{x})$.

For a fixed function $h : [n] \to [k]$, let $C_{\mathbf{a},h}$ denote the subcube as defined in Definition 2.2.6. For a probability distribution $\mu = (\mu_1, \ldots, \mu_k)$ on $[k]$, sampling a random function $h : [n] \to [k]$ according to μ means the following: For each $i \in [n]$ sample independently $h(i) \sim \mu$, i.e. $h(i)$ is equal to j with probability μ_j. We will define it shortly using $h \sim \mu$.

Let $\mathcal{S}$ be the set and $w_1, \ldots, w_k$ be the positive integers as described in Theorem 3.1.2. Let $\mu := \left(\frac{w_1}{W}, \ldots, \frac{w_k}{W}\right)$. We now describe the local correction algorithm.

```
Algorithm 1: Local correction algorithm for sub-constant error
  Input: $f(x_1, \ldots, x_n)$, $\mathbf{a} \in \{0,1\}^n$, $\delta$
1 $k \leftarrow A \cdot (d+1) \cdot (\log n)$          // $A$ an absolute constant chosen below
2 Sample a random function $h : [n] \to [k]$ according to the distribution $\mu$   // The only
    source of randomness
3 $g(y_1, \ldots, y_k) \leftarrow f(x_1, \ldots, x_n)|_{C_{\mathbf{a},h}}$
4 Let $\mathbf{b} = 0^k$ and $c_1, \ldots, c_{|\mathcal{S}|}$ be the integral coefficients for $0^k$ from Theorem 3.1.2.
5 return $\sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}}\, g(\mathbf{u})$
```

Queries: The number of queries is equal to $|\mathcal{S}| \leqslant \mathcal{O}_d(k^d) = \mathcal{O}_d(\log n)^d$ by Theorem 3.1.2 and the value of k in Algorithm 1.

Correctness: We now argue that Algorithm 1 returns $P(\mathbf{a})$ with probability $\geqslant 3/4$. Let $E \subset \{0,1\}^n$ denote the set of points where f and P disagree, i.e.

$$E \;=\; \{\mathbf{x} \in \{0,1\}^n \mid f(\mathbf{x}) \neq P(\mathbf{x})\}$$

We have $|E|/2^n \leqslant \delta$ because $\delta(f,P) \leqslant \delta$. Recall that for each $\mathbf{y} \in \{0,1\}^k$, for all $i \in [n]$, $x(\mathbf{y})_i = y_{h(i)} \oplus a_i$, where h is function sampled in Line 1 of Algorithm 1.

We first argue that if f and P agree at $x(\mathbf{y})$ for every $\mathbf{y} \in \mathcal{S}$, then Algorithm 1 returns $P(\mathbf{a})$. If for every $\mathbf{y} \in \mathcal{S}$, $x(\mathbf{y})$ is not in E, then $g = P|_{\mathcal{S}}$. Since $x(0^k) = \mathbf{a}$, the first property of Theorem 3.1.2 implies $g(0^k) = P(\mathbf{a})$.

Next we show that with probability at least 3/4, for every $\mathbf{y} \in \mathcal{S}$, $x(\mathbf{y}) \notin E$. Equivalently, we show the following:

$$\Pr_h[\exists\ \mathbf{y} \in \mathcal{S} \ \text{ s.t. } \ x(\mathbf{y}) \in E] \;<\; \frac{1}{4} \tag{1}$$

First, we understand the distribution of $x(\mathbf{y})$ for a fixed $\mathbf{y} \in \mathcal{S}$ under a random function $h : [n] \to [k]$ sampled according to μ. Fix any $\mathbf{y} \in \mathcal{S}$ and a coordinate $i \in [n]$. Since $\mathbf{a}$ is fixed, we have,

$$\Pr_{h \sim \mu}[x(\mathbf{y})_i = 1] \;=\; \mathbb{E}_{h\sim\mu}[x(\mathbf{y})_i] \;=\; \sum_{j=1}^{k} \frac{w_j}{W} \cdot y_j$$

From the second property in Theorem 3.1.2, we have,

$$\left| \Pr_{h\sim\mu}[x(\mathbf{y})_i = 1] - \frac{1}{2} \right| \leqslant \frac{1}{2^{\Omega(k/(d+1))}}$$

For each $\mathbf{y} \in \{0,1\}^k$, the coordinates $\{x(\mathbf{y})_i \mid i \in [n]\}$ are mutually independent (this is because $h[i]$ is sampled independently for each $i \in [n]$) and $1/2^{\Omega(k/(d+1))}$-close to the uniform distribution over $\{0,1\}$. From Fact 3.1.3, we know that for every $\mathbf{y} \in \mathcal{S}$, $x(\mathbf{y})$ is $\sqrt{n}/2^{\Omega(k/(d+1))}$-close to the uniform distribution (in statistical distance).

Fact 3.1.3 (Closeness to the uniform distribution)**.** *(See [Man11, Theorem 5.5, Claim 5.6]). Let $\eta > 0$. Let $\mathcal{D}'$ be a distribution on $\{0,1\}^n$ such that for any $\mathbf{y} \sim \mathcal{D}'$, the co-ordinates of $\mathbf{y}$ are independent and for all $i \in [n]$,*

$$1/2 - \eta \leqslant \Pr[y_i = 1] \leqslant 1/2 + \eta.$$

Then $\mathcal{D}'$ is $\mathcal{O}(\eta\sqrt{n})$-close to the uniform distribution over $\{0,1\}^n$.

By the definition of statistical distance, we have that for every $\mathbf{y} \in \mathcal{S}$,

$$\Pr_h[x(\mathbf{y}) \in E] \;\leqslant\; \sqrt{n}/2^{\Omega(k/(d+1))} + \delta$$

Taking a union bound over all $\mathbf{y} \in \mathcal{S}$, we have,

$$\Pr_h[\exists\, \mathbf{y} \in \mathcal{S} \text{ s.t. } x(\mathbf{y}) \in E] \;\leqslant\; |\mathcal{S}| \left(\sqrt{n}/2^{\Omega(k/(d+1))} + \delta\right)$$

Recall that the number of queries is $q = |\mathcal{S}|$ and by assumption $\delta < 1/100q$. Thus, we get

$$\Pr_h[\exists\, \mathbf{y} \in \mathcal{S} \text{ s.t. } x(\mathbf{y}) \in E] \leqslant \frac{\tilde{\mathcal{O}}(\sqrt{n})}{2^{\Omega(k/(d+1))}} + \frac{1}{100} \leqslant \frac{1}{4}$$

as long as the constant A in Algorithm 1 is chosen to be large enough. This shows Equation (1) as claimed and thus we have described a (δ, q) local correction algorithm for $\mathcal{P}_d$. ■

3.2 Weight balanced interpolating set

In this subsection, we prove Theorem 3.1.2. We start by proving Lemma 3.2.1, which is our main technical lemma of this subsection. The difference between Lemma 3.2.1 and Theorem 3.1.2 is in the first condition. In Lemma 3.2.1 we require that every non-zero degree-d polynomial is non-zero on $\mathcal{S}$. Later, we will see that this is sufficient to allow us to compute Q at any point.[11]

Lemma 3.2.1 (Main lemma for local correction)**.** *Fix a degree parameter $d \geqslant 0$ and a dimension parameter $k \in \mathbb{Z}_{>0}$ such that k is divisible by $10 \cdot (d+1)$. There exists a set $\mathcal{S} \subseteq \{0,1\}^k$ such that for every Abelian group G, the set $\mathcal{S}$ satisfies the following properties:*

- *For every non-zero degree-d polynomial $Q(y_1, \ldots, y_k) \in \mathcal{P}_d(\{0,1\}^k, G)$, there exists a point $\mathbf{z} \in \mathcal{S}$ such that $Q(\mathbf{z}) \neq 0$.*
- *There exists positive integers $w_1, \ldots, w_k$ such that*

$$\mathcal{S} \;\subseteq\; \left\{ \mathbf{y} \in \{0,1\}^k \;\middle|\; \left| \sum_{j=1}^{k} w_j y_j - \frac{W}{2} \right| \;\leqslant\; \frac{W}{2^{\Omega(k/(d+1))}} \right\},$$

where $W := \sum_{j=1}^{k} w_j$.

Furthermore, $|\mathcal{S}|$ is at most $\mathcal{O}_d(k^d)$.

[11] For polynomials over fields, this follows simply from linear algebra. For Abelian groups, the proof is similar, but we need a result from the theory of Diophantine linear equations.

Proof of Lemma 3.2.1. Let $r = 10 \cdot (d+1)$ and let $\Sigma := \{0,1\}^r$. Let $m = k/r$. In this proof, it will be convenient to make the following identification:

$$[mr] \cong [m] \times [r] \quad \text{and} \quad \{0,1\}^k = \Sigma^m$$

We interpret $\mathbf{y} \in \{0,1\}^{mr}$ as $\mathbf{y} = (\mathbf{y}[1], \ldots, \mathbf{y}[m])$ where for every $i \in [m]$, $\mathbf{y}[i] = (y[i,1], \ldots, y[i,r]) \in \Sigma = \{0,1\}^r$.

We now describe the weights $w_1, \ldots, w_k$. For every for $(i,j) \in [m] \times [r] \cong [mr]$, let $w_{(i,j)} := 2^{i-1}$. Define the set $\mathcal{B}_m$ as follows.

$$\mathcal{B}_m \ := \ \left\{ \mathbf{y} \in \Sigma^m \ \middle| \ \left| \sum_{i=1}^{m} \sum_{j=1}^{r} w_{(i,j)} y[i,j] - \frac{W_m}{2} \right| \leqslant t \right\} \ = \ \left\{ \mathbf{y} \in \Sigma^m \ \middle| \ \left| \sum_{i=1}^{m} 2^{i-1} \sum_{j=1}^{r} y[i,j] - \frac{W_m}{2} \right| \leqslant t \right\}$$

where $W_m := \sum_{i=1}^m 2^{i-1} \sum_{j=1}^r 1 = r(1 + 2 \cdots + 2^{m-1})$ and $t = \lceil \frac{d}{2} \rceil$. We will choose the set $\mathcal{S}$ to be a subset of $\mathcal{B}_m$. Note that $t = W_m/2^m$, which shows that the weights are as required by the second item of the statement of the lemma.

More generally, we can define a subset $\mathcal{B}_\ell \subseteq \{0,1\}^{\ell \cdot r}$ for each $\ell \in [m]$ in a similar way. We let W_ℓ denote the sum of the weights in this case. We will need the following claim about extending points in $\mathcal{B}_\ell$ to points in $\mathcal{B}_{\ell+1}$ in many different ways.

Claim 3.2.2. *Fix $\ell \in [m-1]$. Let $\mathbf{b} = (\mathbf{b}[1], \ldots, \mathbf{b}[\ell])$ be any point in $\mathcal{B}_\ell$. There exists an interval (i.e. set of consecutive integers) $I_{\mathbf{b}} \subseteq \{0, \ldots, r\}$ of size at least $d+1$ such that for every point $\mathbf{z} \in \{0,1\}^r$ such that $|\mathbf{z}| \in I_{\mathbf{b}}$, the point $\mathbf{b}' = (\mathbf{z}, \mathbf{b}[1], \ldots, \mathbf{b}[\ell]) \in \mathcal{B}_{\ell+1}$.*

Proof of Claim 3.2.2. Note that $W_{\ell+1} = 2W_\ell + r$. Since $\mathbf{b} \in \mathcal{B}_\ell$, we have

$$\sum_{i=1}^{\ell} |\mathbf{b}[i]| \cdot 2^{i-1} = \frac{W_\ell}{2} + \tau \tag{2}$$

for some integer τ such that $|\tau| \leqslant t$ (here $|\mathbf{b}[i]|$ is the Hamming weight of $\mathbf{b}[i]$). For $\mathbf{b}' = (\mathbf{z}, \mathbf{b}[1], \ldots, \mathbf{b}[\ell])$ to lie in $\mathcal{B}_{\ell+1}$, we need

$$|\mathbf{z}| + \sum_{i=1}^{\ell} |\mathbf{b}[i]| \cdot 2^i \in \left[\frac{W_{\ell+1}}{2} - t, \frac{W_{\ell+1}}{2} + t \right].$$

By Equation (2) and the relationship between W_ℓ and $W_{\ell+1}$, the latter is equivalent to $|\mathbf{z}| \in I_{\mathbf{b}} := [\frac{r}{2} - 2\tau - t, \frac{r}{2} - 2\tau + t]$. Since $r/2 \geqslant 5(d+1) \geqslant 2|\tau| + t$, the interval $I_{\mathbf{b}} \subseteq \{0, \ldots, r\}$ and further, $|I_{\mathbf{b}}| = 2t + 1 \geqslant d + 1$. ■

We also need a basic fact about degree-d polynomials over the Boolean cube. This is a combination of a few folklore facts, but we prove it here for completeness.

Claim 3.2.3. *Fix any $n \geqslant 1$ and degree parameter $d \leqslant n$. For every interval $I \subseteq \{0, \ldots, n\}$ (i.e. set of consecutive integers) of size $(d+1)$, there exists a set $\mathcal{H}_{I,d} \subseteq \{0,1\}^n$ of size at most $(2(n+1))^d$ such that*

- *$\mathcal{H}_{I,d}$ consists only of points $\mathbf{z}$ such that $|\mathbf{z}| \in I$, and*
- *For any non-zero $P \in \mathcal{P}_d(\{0,1\}^n, G)$, there is a point $\mathbf{z} \in \mathcal{H}_{I,d}$ such that $P(\mathbf{z}) \neq 0$.*

Proof of Claim 3.2.3. Assume that $I = \{a, a+1, \ldots, a+d\}$ for some $a \in [0, n-d]$. For every subset $A \subseteq [n]$ of size $\leqslant d$, we define the set $\mathcal{H}_{I,d}^A$ as follows:

$$\mathcal{H}_{I,d}^A \; := \; \left\{ \mathbf{x}1^a 0^{n-|A|-a} \;\middle|\; \mathbf{x} \in \{0,1\}^{|A|} \right\},$$

where $\mathbf{x}1^a0^{n-|A|-a}$ is a shorthand notation for the point where the variables indexed by A are set to $\mathbf{x}$, the first a variables of the remaining variables are set to 1, and the last $(n-|A|-a)$ variables of the remaining variables are set to 0. In other words, $\mathcal{H}_{I,d}^A$ consists of points where the variables indexed by $[n]\backslash A$ are set to $1^a0^{n-|A|-a}$ (this has Hamming weight a) and the variables indexed by A can be any point of Hamming weight $\leqslant |A| \leqslant d$. We define the set $\mathcal{H}_{I,d}$ as follows:

$$\mathcal{H}_{I,d} \; = \; \bigcup_{\substack{A \subseteq [n] \\ |A| \leqslant d}} \mathcal{H}_{I,d}^A$$

For every subset A, the set $\mathcal{H}_{I,d}^A$ consists only of points $\mathbf{z}$ such that $|\mathbf{z}| \in I$, and so thus every point in $\mathcal{H}_{I,d}$. Note that for each subset $A \subseteq \binom{[n]}{\leqslant d}$, the set $|\mathcal{H}_{I,d}^A| \leqslant 2^d$. The size of $\mathcal{H}_{I,d}$ is at most $2^d \cdot (\sum_{i=0}^d \binom{n}{i}) \leqslant (2(n+1))^d$.

Next we show that for any non-zero $P \in \mathcal{P}_d(\{0,1\}^n, G)$, there is a point $\mathbf{z} \in \mathcal{H}_{I,d}$ such that $P(\mathbf{z}) \neq 0$. Fix any non-zero polynomial $P \in \mathcal{P}_d(\{0,1\}^n, G)$. P can be uniquely expressed as the following multilinear polynomial:

$$P(x_1, \ldots, x_n) \; = \; \sum_{\substack{A \subseteq [n] \\ |A| \leqslant d}} c_A \mathbf{x}^A,$$

where $\mathbf{x}^A$ is the product of variables indexed by the set A. Since P is a non-zero polynomial, at least one of the coefficients c_A's is non-zero. Let A_0 be the maximal (with respect to the inclusion partial order) subset with a non-zero coefficient. Set the variables outside A_0 to $1^a0^{n-|A_0|-a}$. The resulting polynomial is a non-zero degree d polynomial in variables indexed by A_0, i.e. in $\leqslant d$ variables. We know (Theorem 2.2.1) that every non-zero degree-d polynomial is non-zero on the Boolean cube. Thus the resulting polynomial is non-zero on a point in $\{0,1\}^{|A_0|}$. This implies that P is non-zero on a point in $\mathcal{H}_{I,d}^{A_0} \subseteq \mathcal{H}_{I,d}$. ■

We now show how to construct the set $\mathcal{S}$ as required in the statement of the lemma. In fact, we will show a stronger property: for each $\ell \in [m]$ and $j \in \{0, \ldots, d\}$, we show that there is a set $\mathcal{S}_{\ell,j} \subseteq \mathcal{B}_\ell$ that satisfies the first item of the lemma w.r.t. the space of polynomials $\mathcal{P}_j(\{0,1\}^{\ell \cdot r}, G)$. Furthermore, for each ℓ, j, we will have $|\mathcal{S}_{\ell,j}| \leqslant (2(r+1))^j \cdot \ell^j$.

We prove the above by induction on $\ell + j$. The base case corresponds to $\ell = 1$ and $j = 0$, where we can take $\mathcal{S}_{\ell,0}$ to be any fixed point $\mathbf{z} \in \mathcal{B}_\ell$.

Now consider the inductive case for some $j > 0$. Given a non-zero polynomial $P(\mathbf{y}[1], \ldots, \mathbf{y}[\ell]) \in \mathcal{P}_j(\{0,1\}^{\ell \cdot r}, G)$, we can decompose it as a polynomial in the variables in $\mathbf{y}[1]$ with coefficients coming from the space of polynomials in the remaining variable sets $\mathbf{y}[2], \ldots, \mathbf{y}[\ell]$. This gives the following equality.

$$P(\mathbf{y}[1], \ldots, \mathbf{y}[\ell]) = \sum_{A \subseteq [r] : |A| \leqslant j} Q_A(\mathbf{y}[2], \ldots, \mathbf{y}[\ell]) \cdot \mathbf{y}[1]^A \tag{3}$$

where $\mathbf{y}[1]^A$ denotes the product of the variables in $\mathbf{y}[1]$ indexed by A and Q_A denotes the sum of all monomials in $\mathbf{y}[2], \ldots, \mathbf{y}[\ell]$ multiplying this monomial. Note that Q_A has degree at most $j - |A|$.

Fix a set A_0 such that Q_{A_0} is non-zero and $|A_0|$ is as large as possible. Assume that $|A_0| = j' \in \{0, \ldots, j\}$. We know by induction that there is a point $\mathbf{b} \in \mathcal{S}_{\ell-1, j-j'}$ such that $Q_{A_0}(\mathbf{b}) \neq 0$. Let $P_{\mathbf{b}}(\mathbf{y}[1])$ denote the restriction of the polynomial P when the variable sets $\mathbf{y}[2], \ldots, \mathbf{y}[\ell]$ are set according to $\mathbf{b}$. The polynomial $P_{\mathbf{b}}$ is a non-zero polynomial of degree j'.

We want to extend $\mathbf{b}$ to an assignment also setting the variables $\mathbf{y}[1]$ that keeps the polynomial P non-zero. By Claim 3.2.2, there is an interval $I_{\mathbf{b}}$ of size at least $(d+1)$ such that for any $\mathbf{z}$ such that $|\mathbf{z}| \in I_{\mathbf{b}}$, the point $(\mathbf{z}, \mathbf{b}[1], \ldots, \mathbf{b}[\ell-1]) \in \mathcal{B}_\ell$. Fix any subinterval $I_{\mathbf{b},j'} \subseteq I_{\mathbf{b}}$ of size $j'+1$. By Claim 3.2.3, there is a set $\mathcal{H}_{I_{\mathbf{b},j'},j'} \subseteq \{0,1\}^r$ of size at most $(2(r+1))^{j'}$ such that each point $\mathbf{z}$ has Hamming weight in I and further $P_{\mathbf{b}}(\mathbf{z}) \neq 0$.

We have thus shown that P must be non-zero at one of the points in the following set.

$$\mathcal{S}_{\ell,j,j'} = \{(\mathbf{z}, \mathbf{b}[1], \ldots, \mathbf{b}[\ell-1]) \mid \mathbf{b} = (\mathbf{b}[1], \ldots, \mathbf{b}[\ell-1]) \in \mathcal{S}_{\ell-1,j-j'},\ \mathbf{z} \in \mathcal{H}_{I_{\mathbf{b},j'},j'}\}.$$

However, the above assumes that we know the parameter j' of P. To define the set $\mathcal{S}_{\ell,j}$, we take a union of all the sets $\mathcal{S}_{\ell,j,j'}$ for $j' \in \{0, \ldots, j\}$. This satisfies the required inductive property.

It remains to bound $|\mathcal{S}_{\ell,j}|$. We have

$$\begin{aligned}|\mathcal{S}_{\ell,j}| &\leqslant \sum_{j'=0}^{j} |\mathcal{S}_{\ell,j,j'}| \leqslant \sum_{j'=0}^{j} |\mathcal{S}_{\ell-1,j-j'}| \cdot |\mathcal{H}_{I,j'}| \leqslant \sum_{j'=0}^{j} (2(r+1))^{(j-j')} \cdot (\ell-1)^{j-j'} \cdot (2(r+1))^{j'} \\ &= (2(r+1)^j \cdot ((\ell-1)^j + (\ell-1)^{j-1} + \cdots + (\ell-1)^0) \leqslant (2(r+1))^j \cdot \ell^j\end{aligned}$$

proving the required bound on $|\mathcal{S}_{\ell,j}|$. This proves the inductive claim.

To conclude the proof of Lemma 3.2.1, if we take $\mathcal{S} = \mathcal{S}_{m,d}$, then we have a set with the required properties and size. ■

We now show how Lemma 3.2.1 implies Theorem 3.1.2. We recall Theorem 3.1.2 here.

Theorem 3.1.2 (Weight balanced interpolating set)**.** *Fix a degree parameter $d \geqslant 0$ and a dimension parameter $k \in \mathbb{Z}_{>0}$ that is divisible by $10(d+1)$. There exists a set $\mathcal{S} \subseteq \{0,1\}^k$ such that for every Abelian group G, $\mathcal{S}$ satisfies the following properties:*

1. *[Interpolating set]. For each point $\mathbf{b} \in \{0,1\}^k$, there exists integral coefficients $c_1, \ldots, c_{|S|}$ such that for every degree-d polynomial $Q(y_1, \ldots, y_k) \in \mathcal{P}_d(\{0,1\}^k,\ G)$, we have,*

$$Q(\mathbf{b}) \;=\; \sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} Q(\mathbf{u})$$

2. *[Weighted balanced]. There exists positive integers $w_1, \ldots, w_k$ such that*

$$\mathcal{S} \subseteq \left\{ \mathbf{y} \in \{0,1\}^k \;\middle|\; \left| \sum_{j=1}^{k} w_j y_j - \frac{W}{2} \right| \leqslant \frac{W}{2^{\Omega(k/(d+1))}} \right\},$$

where $W := \sum_{j=1}^{k} w_j$.

Furthermore, $|\mathcal{S}|$ is at most $\mathcal{O}_d(k^d)$.

Proof of Theorem 3.1.2. Let $\mathcal{S} \subseteq \{0,1\}^k$ be the subset as given by Lemma 3.2.1. Fix any point $\mathbf{b} \in \{0,1\}^k$. Let B_d denote the set of multilinear monomials of degree $\leqslant d$ in $\{x_1, \ldots, x_k\}$. B_d forms a spanning set of $\mathcal{P}_d(\{0,1\}^k, G)$ for every G, i.e. every polynomial $Q \in \mathcal{P}_d(\{0,1\}^k, G)$ can be expressed as a unique linear combination of monomials from B_d (with coefficients from G). Fix some total orders on $\mathcal{S}$ and B_d.

Construct the matrix M of dimensions $|\mathsf{B}_d| \times |\mathcal{S}|$ as follows: The rows are indexed by monomials in B_d and the columns are indexed by points in $\mathcal{S}$. For $1 \leqslant i \leqslant |\mathsf{B}_d|$ and $1 \leqslant j \leqslant |\mathcal{S}|$, $M[i,j]$ is equal to $m(\mathbf{u})$, where m is the i^{th} monomial in B_d and $\mathbf{u}$ is the j^{th} point in $\mathcal{S}$. In other words, the j^{th} column of M denotes the vector whose entries are the evaluation of all the monomials in the spanning set B_d of the j^{th} point in $\mathcal{S}$.

We will first prove Claim 3.2.4 and later show that Claim 3.2.4 is enough to prove Theorem 3.1.2.

Claim 3.2.4. *Let M be the matrix of dimensions $|\mathsf{B}_d| \times |\mathcal{S}|$ as described above. Define $\boldsymbol{\beta} \in \mathbb{Z}^{|\mathsf{B}_d|}$ as follows: For $1 \leqslant i \leqslant |\mathsf{B}_d|$, β_i is equal to $m(\mathbf{b})$, where m is the i^{th} monomial of B_d. There exists an integral vector $\mathbf{c} = (c_1, \ldots, c_{|\mathcal{S}|})$ such that the following equation is satisfied:*

$$M\mathbf{c} = \boldsymbol{\beta} \tag{4}$$

Equivalently, there exists an integral vector $\mathbf{c} = (c_1, \ldots, c_{|\mathcal{S}|})$ such that for every monomial $m \in \mathsf{B}_d$,

$$\sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} m(\mathbf{u}) = m(\mathbf{b}).$$

Proof. To prove the existence of an integral vector $\mathbf{c}$, we need the following lemma.

Lemma 3.2.5. *[Sch86, Corollary 4.1.a] Let A be a rational matrix and let $\mathbf{a}$ be a rational vector. Then the system $A\mathbf{x} = \mathbf{a}$ has an integral solution $\mathbf{x}$ if and only if for every row vector $\mathbf{y}$ for which $\mathbf{y}A$ is integral, $\mathbf{y}\mathbf{a}$ is an integer.*

Lemma 3.2.5 says that to show the existence of an integral solution $\mathbf{c}$ to Equation (4), it is equivalent to show that for every rational row vector $\mathbf{y} \in \mathbb{Q}^{|\mathsf{B}_d|}$ for which $\mathbf{y}M$ is integral, $\mathbf{y}\boldsymbol{\beta}$ is an integer.

Consider any $\mathbf{y} \in \mathbb{Q}^{|\mathsf{B}_d|}$ for which $\mathbf{y}M$ is integral. Let H denote the quotient group $\mathbb{Q}/\mathbb{Z}$, which can be identified with rational numbers in $[0,1)$ where addition is carried out modulo 1. Define $\mathbf{y}'$ to be the image of $\mathbf{y}$ in H under the natural projection from $\mathbb{Q}$ to H, i.e. for each coordinate $1 \leqslant i \leqslant |\mathsf{B}_d|$, $y'_i := y_i - \lfloor y_i \rfloor$.

In what follows, we are still treating the entries of $\boldsymbol{\beta}$ and M as integers, and thus it makes sense to multiply these entries with the entries of $\mathbf{y}'$ to get elements of H. The hypothesis that $\mathbf{y}M$ is integral is equivalent to saying that over the group H, $\mathbf{y}'M = 0^{|\mathcal{S}|} \in H^{|\mathcal{S}|}$ is the all-zeroes vector. Similarly, showing that $\mathbf{y}\boldsymbol{\beta}$ is an integer is equivalent to showing that $\mathbf{y}'\boldsymbol{\beta}$ is 0.

Assume for the sake of contradiction that $\mathbf{y}'\boldsymbol{\beta}$ is non-zero. Let Q be the polynomial in $\mathcal{P}_d(\{0,1\}^k, H)$ whose coefficient vector is $\mathbf{y}'$, i.e.

$$Q(\mathbf{x}) := \sum_{m \in \mathsf{B}_d} y'_m m$$

The hypothesis $\mathbf{y}'M = 0$ means that the polynomial Q vanishes on $\mathcal{S}$. On the other hand, $\mathbf{y}'\boldsymbol{\beta} \in (0,1)$ means $Q(\mathbf{b})$ is non-zero, i.e. Q is a non-zero polynomial. So we have a non-zero polynomial Q that vanishes on the set $\mathcal{S}$, which contradicts Theorem 3.1.2. Hence $\mathbf{y}'\boldsymbol{\beta} = 0$, implying that $\mathbf{y}\boldsymbol{\beta}$ is an integer.

As $\mathbf{y}$ is an arbitrary row vector for which $\mathbf{y}M$ is integral, using Lemma 3.2.5 we get the existence of an integral solution to Equation (4). This finishes the proof of Claim 3.2.4. ■

Finally, we argue that Claim 3.2.4 is sufficient to finish the proof of Theorem 3.1.2. This is essentially because $\mathcal{P}_d(\{0,1\}^k, G)$ is spanned by B_d. Consider any polynomial $Q \in \mathcal{P}_d(\{0,1\}^k, G)$. There exists coefficients $\alpha_1, \ldots, \alpha_{|\mathsf{B}_d|}$ such that

$$Q(\mathbf{x}) = \sum_{m \in \mathsf{B}_d} \alpha_m m$$

Let $c_1, \ldots, c_{|\mathcal{S}|}$ be the coefficients from the above claim. Then we have,

$$\sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} Q(\mathbf{u}) = \sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} \sum_{m \in \mathsf{B}_d} \alpha_m m(\mathbf{u}) = \sum_{m \in \mathsf{B}_d} \alpha_m \sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} m(\mathbf{u})$$

Since $M\mathbf{c} = \boldsymbol{\beta}$, for every $m \in \mathsf{B}_d$, we have,

$$\sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} m(\mathbf{u}) = m(\mathbf{b})$$

Thus we get,

$$\sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} Q(\mathbf{u}) = \sum_{m \in \mathsf{B}_d} \alpha_m \sum_{\mathbf{u} \in \mathcal{S}} c_{\mathbf{u}} m(\mathbf{u}) = \sum_{m \in \mathsf{B}_d} \alpha_m m(\mathbf{b}) = Q(\mathbf{b})$$

This finishes the proof of Theorem 3.1.2. ■

3.3 Error close to half the minimum distance (Proof of Theorem 1.3.1)

In this subsection, we explain the second step towards proving Theorem 1.3.1. In the previous subsection, we described a local correction algorithm for $\mathcal{P}_d$ when the error is $1/\mathcal{O}_d((\log n)^d)$. We now want to locally correct degree-d polynomials when the error is close to the unique decoding radius, which is $1/2^{d+1}$.

Suppose we have oracle access to a function f that is $(1/2^{d+1} - \varepsilon)$-close to $\mathcal{P}_d(\{0,1\}^k, G)$ for some constant $\varepsilon > 0$ and let P be the unique polynomial in $\mathcal{P}_d(\{0,1\}^k, G)$ such that $\delta(f, P) \leqslant (1/2^{d+1} - \varepsilon)$. The idea is to design a randomized algorithm $\mathcal{A}$ that has oracle access to the function f and returns a probabilistic oracle $\mathcal{A}^f$ such that $\delta(\mathcal{A}^f, P) < 1/\mathcal{O}((\log n)^d)$, with high probability. The algorithm $\mathcal{A}$ will be referred to as *error reduction* algorithm. More specifically, the error reduction algorithm $\mathcal{A}$ will have two subroutines as follows:

1. There is a randomized algorithm $\mathcal{A}_1$ that reduces the error from $(1/2^{d+1} - \varepsilon)$ down to $1/1000$.
2. There is a randomized algorithm $\mathcal{A}_2$ that reduces the error from $1/1000$ down to $1/\mathcal{O}((\log n)^d)$.

[ABP$^+$24] gave the error reduction algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, which we state below.

Lemma 3.3.1 (Error reduction for error close to half the minimum distance)**.** *[ABP$^+$24, Lemma 3.13] Fix any Abelian group G and a positive integer d. For any η_1, δ, where $\eta < \delta$ and $\delta < 1/2^{d+1} - \varepsilon$ for $\varepsilon > 0$, there exists a randomized algorithm $\mathcal{A}_1$ with the following properties:*
Let $f : \{0,1\}^n \to G$ be a function and let $P : \{0,1\}^n \to G$ be a degree d polynomial such that $\delta(f, P) \leqslant \delta$, and let $\mathcal{A}_1^f$ denotes that $\mathcal{A}$ has oracle access to f, then

$$\Pr[\delta(\mathcal{A}_1^f, P) > \eta_1] < 1/20,$$

where the above probability is over the internal randomness of $\mathcal{A}_1$, and for every $\mathbf{x} \in \{0,1\}^n$, $\mathcal{A}_1^f$ makes 2^k queries to f, where $k = \operatorname{poly}(\frac{1}{\varepsilon}, \frac{1}{\eta_1})$.

Lemma 3.3.2 (Error reduction for constant error)**.** *[ABP$^+$24, Lemma 3.8] Fix any Abelian group G and a positive integer d. The following holds for $\eta_1 < 1/2^{\mathcal{O}(d)}$ and $K = 2^{\mathcal{O}(d)}$ where the $\mathcal{O}(\cdot)$ hides a large enough absolute constant.*

For any $\eta_2 < \eta_1$, there exists a randomized algorithm $\mathcal{A}_2$ with the following properties: Let $f : \{0,1\}^n \to G$ be a function and let $P : \{0,1\}^n \to G$ be a degree-d polynomial such that $\delta(f, P) \leqslant \delta$, and let $\mathcal{A}_2^f$ denotes that $\mathcal{A}_2$ has oracle access to f, then

$$\Pr[\delta(\mathcal{A}_2^f, P) > \eta_2] < 1/20,$$

where the above probability is over the internal randomness of $\mathcal{A}_2^f$. Further, for every $\mathbf{x} \in \{0,1\}^n$, $\mathcal{A}_2^f$ makes K^T queries to f and $T = \mathcal{O}\left(\log\left(\frac{\log(1/\eta_2)}{\log(1/\delta)}\right)\right)$.

Using Lemma 3.3.1 and Lemma 3.3.2 along with Theorem 3.1.1, we get Theorem 1.3.1. We restate Theorem 1.3.1 and finish the proof.

Theorem 1.3.1 (Local correction algorithms for $\mathcal{P}_d$ up to the unique decoding radius)**.** *For every Abelian group G and for every constant $\varepsilon > 0$, the space $\mathcal{P}_d$ has a (δ, q)-local correction algorithm where $\delta = \frac{1}{2^{d+1}} - \varepsilon$ and $q = \widetilde{\mathcal{O}}_\varepsilon(\log n)^d$.*

Proof of Theorem 1.3.1. Let f be a function with oracle access that is δ-close to a degree-d polynomial $P \in \mathcal{P}_d(\{0,1\}^k, G)$.

1. Lemma 3.3.1 (with $\eta_1 = 1/2^{\mathcal{O}(d)}$ chosen to satisfy the hypothesis of Lemma 3.3.2) yields an oracle $\mathcal{A}_1^f$ that makes $\mathcal{O}_\varepsilon(1)$ queries to f and is η_1-close to P with probability at least $19/20$. We fix the randomness of $\mathcal{A}_1^f$ so that this holds.

2. With probability at least 19/20, we have oracle access to a function $\mathcal{A}_1^f$ that is $1/2^{\mathcal{O}(d)}$-close to P. Let $g := \mathcal{A}_1^f$. Lemma 3.3.2 (with $\eta_2 = 1/\mathcal{O}((\log n)^d)$) yields a probabilistic oracle $\mathcal{A}_2^g$ that makes $\text{poly}(\log\log n, d)$ queries to g and is η_2-close to P with probability at least 19/20. We again fix the randomness of $\mathcal{A}_2^g$ so that this holds.

In other words, we have oracle access to an oracle $\mathcal{A}_2^g$ that makes $\text{poly}(\log\log n, d)\cdot\mathcal{O}_\varepsilon(1)$ queries to f and is η_2-close to P with probability at least 9/10. We now apply the local correction algorithm from Theorem 3.1.1 with oracle access to $\mathcal{A}_2^g$ to get a local correction algorithm for $\mathcal{P}_d$ with $\delta = 1/2^{d+1} - \varepsilon$ for any constant $\varepsilon > 0$ and $q = \tilde{\mathcal{O}}_\varepsilon((\log n)^d)$. ■

Remark 3.3.3. *It should be noted that Theorem 1.3.1 also follows from the theorem on local list-correction Theorem 1.3.4 along with known results on* locally testing low-degree polyomials *[BSS20]. However, we state this theorem separately for multiple reasons. Firstly, a weaker form of this theorem is required for the results on local list-correction. Secondly, the above result is natural and this gives a simpler proof of this than the one outlined above. And finally, this proof yields a better dependence on the degree parameter d.*

Having finished the proof of Theorem 1.3.1, in the following subsection, we now prove Theorem 1.3.2 by giving a local correction algorithm with improved query complexity for groups of small exponent (see Section 2 for a definition of exponent of a group).

3.4 Local correction for groups of constant exponent

In this subsection, we show that we can bring down the query complexity of local correction from $\tilde{\mathcal{O}}_d((\log n)^d)$ to a constant (i.e., independent of n) when G is an Abelian torsion group of constant exponent. More specifically, we prove Theorem 1.3.2 from the introduction.

Theorem 1.3.2. *If G is an Abelian torsion group of exponent M, then for every $\varepsilon > 0$, $\mathcal{P}_d$ has a (δ, q)-local correction algorithm where $\delta = \frac{1}{2^{d+1}} - \varepsilon$ and $q = \mathcal{O}_{M,\varepsilon}(1)$.*

We note that to prove Theorem 1.3.2 for every $\delta = \frac{1}{2^{d+1}} - \varepsilon$, it suffices to show the following lemma for *some* constant $\delta = \Omega_{M,d}(1)$, since the error reduction steps from Section 3.3 can be applied without change.

Lemma 3.4.1. *For every Abelian torsion group G of exponent M, the family $\mathcal{P}_d$ has a (δ, q)-local correction algorithm for some $\delta = \Omega_{M,d}(1)$ and $q = \mathcal{O}_{M,d}(1)$.*

The proof of the above lemma proceeds in an identical manner to the analysis of [BSS20] where the authors show this when G is the underlying group of a *field* of constant characteristic. They do this by using Lucas' theorem which gives a criterion for a binomial coefficient to be divisible by a given prime. To handle the more general case of groups of constant exponent, we instead make use of Kummer's theorem which may be thought of as an analog of Lucas' theorem for prime powers. We state Kummer's theorem below, where the notation $S_p(n)$ denotes the sum of the digits of n when written in base p.

Theorem 3.4.2 (Kummer's theorem [Kum52])**.** *Let $p \in \mathbb{N}$ be a prime. Then for any integers $a \geqslant b \geqslant 0$, the largest power of p that divides $\binom{a}{b}$ is equal to $\frac{S_p(b)+S_p(a-b)-S_p(a)}{p-1}$.*

We are now ready to prove Lemma 3.4.1.

Proof of Lemma 3.4.1. Let $M = \prod_{j=1}^{\ell} p_j^{r_j}$ be the prime factorization of M (so $\ell \leqslant \log M$). For each $j \in [\ell]$, let $s_j \in \mathbb{N}$ be the smallest integer such that $p_j^{r_j s_j} > d$. Then, we choose $k = \prod_{j \in [\ell]} p_j^{3 r_j s_j}$. Note that $p_j^{r_j(s_j - 1)} \leqslant d$ and hence $k \leqslant \prod_{j \in [\ell]} (d p_j^{r_j})^3 \leqslant d^{3\ell} M^3 = \mathcal{O}_{M,d}(1)$. We set $\delta = \frac{1}{4\binom{2k}{k}} = \Omega_{M,d}(1)$.

We claim that the algorithm below (Algorithm 2) is the desired local corrector. For a given point $\mathbf{a} \in \{0,1\}^n$, it queries f and outputs $P(\mathbf{a})$ with probability at least $3/4$, where $P \in \mathcal{P}_d$ is the unique degree-d polynomial such that $\delta(f, P) \leqslant \delta$. It is similar to Algorithm 1 in that it samples a random subcube $C_{\mathbf{a},h}$ passing through $\mathbf{a}$ and queries it, but the crucial difference now is that we use a different interpolating set in the last step (as opposed to the "weight balanced interpolating set" of Theorem 3.1.2). In particular, we will prove following claim.

Claim 3.4.3. *There exist integers $c_{\mathbf{b}}$ for $\mathbf{b} \in \binom{[2k]}{k}$ such that for every degree-d polynomial $Q(\mathbf{y}) \in \mathcal{P}_d(\{0,1\}^{2k}, G)$, we have that*

$$Q(0^{2k}) = \sum_{\mathbf{b} \in \binom{[2k]}{k}} c_{\mathbf{b}} \cdot Q(\mathbf{b}). \tag{5}$$

Algorithm 2: Local correction algorithm for sub-constant error

```
Input: $f(x_1, \ldots, x_n)$, $\mathbf{a} \in \{0,1\}^n$, $\delta = \frac{1}{4\binom{2k}{k}}$
Sample a uniformly random function $h : [n] \to [2k]$
$g(y_1, \ldots, y_{2k}) \leftarrow f(x_1, \ldots, x_n)|_{C_{\mathbf{a},h}}$
Let $(c_{\mathbf{b}})_{\mathbf{b} \in \binom{[2k]}{k}}$ be the integral coefficients given by Claim 3.4.3.
return $\sum_{\mathbf{b} \in \binom{[2k]}{k}} c_{\mathbf{b}} \cdot g(\mathbf{b})$
```

Assuming the correctness of Claim 3.4.3, we shall now finish the proof of Lemma 3.4.1.

Queries: The local corrector makes $\binom{2k}{k} = \mathcal{O}_{M,d}(1)$ queries since to get the value of $g(\mathbf{b})$ for some $\mathbf{b} \in \binom{[2k]}{k}$, we only need to know $f(x(\mathbf{b}))$ under the mapping h.

Correctness: For every $\mathbf{b} \in \binom{[2k]}{k}$, we note that the corresponding query point $x(\mathbf{b}) \in \{0,1\}^n$ is uniformly distributed since the map h used in Algorithm 2 is uniformly random and $\mathbf{b}$ has equal number of zeroes and ones. Hence, with probability at least $1 - \delta \cdot \binom{2k}{k} \geqslant 3/4$, $g(\mathbf{b}) = Q(\mathbf{b})$ for all $\mathbf{b} \in \binom{[2k]}{k}$, where $Q \in \mathcal{P}_d(\{0,1\}^{2k}, G)$ is the restriction of P on the subcube $C_{\mathbf{a},h}$. Hence, by Claim 3.4.3, the outputted value equals $Q(0^{2k}) = P(\mathbf{a})$ with probability at least $3/4$. ■

It remains to prove Claim 3.4.3. For every $\mathbf{b} \in \binom{[2k]}{k}$, we set $c_{\mathbf{b}} = 0$ if $\mathbf{b}$ contains a 1 in any of the last $k - d$ coordinates and we set $c_{\mathbf{b}} = A$ otherwise, where $A \in \mathbb{Z}$ will be decided later. Recall that $M = \prod_{j \in [\ell]} p_j^{r_j}$ and $k = \prod_{j \in [\ell]} p_j^{3 r_j s_j}$, and we have that $p_j^{r_j s_j} > d \geqslant p_j^{r_j(s_j - 1)}$ for all $j \in [\ell]$.

By linearity, it suffices to show (5) for $Q(\mathbf{y})$ of the form $g \cdot \prod_{j \in I} y_j$ for all $I \in \binom{[2k]}{\leqslant d}$ and $g \in G$. According to our assignment of $c_{\mathbf{b}}$, it is clear that (5) holds true (LHS = RHS = 0) if I contains any of the last $k-d$ coordinates. Otherwise, we have that $I \subseteq \binom{[k+d]}{\leqslant d}$. If $I = \varnothing$, we have $Q(0^{2k}) = g$ and $\sum_{\mathbf{b} \in \binom{[2k]}{k}} c_{\mathbf{b}} \cdot Q(\mathbf{b}) = \binom{k+d}{k} A \cdot g$. On the other hand, if $|I| = i \geqslant 1$, we have $Q(0^{2k}) = 0$ and $\sum_{\mathbf{b} \in \binom{[2k]}{k}} c_{\mathbf{b}} \cdot Q(\mathbf{b}) = \binom{k+d-i}{k-i} A \cdot g$ since every non-zero term must have $b_j = 1$ for all $j \in I$. Hence, it suffices to find an integer A satisfying the following two conditions:

$$g = \binom{k+d}{k} A \cdot g, \text{ for all } g \in G, \text{ and}$$
$$0 = \binom{k+d-i}{k-i} A \cdot g, \text{ for all } g \in G \text{ and } i \in [d].$$

Since the order of every element g divides the exponent M of the group, for the above two conditions to hold, it suffices if for all $j \in [\ell]$ and $i \in [d]$, p_j does not divide $\binom{k+d}{k}$ and that $p_j^{r_j}$ divides $\binom{k+d-i}{k-i}$. Then we can take A to be any integer such that $A\binom{k+d}{k} + A'M = 1$ for some integer A' (such A and A' are guaranteed to exist as M and $\binom{k+d}{k}$ are coprime). The rest of the proof is dedicated to verifying these divisibility constraints hold.

- **p_j does not divide $\binom{k+d}{k}$:** We will represent all the numbers k, d, i etc. in base p_j. We note that the last $r_j s_j$ digits of k are zeroes since $p_j^{r_j}$ divides k. Furthermore, since $d < p_j^{r_j s_j}$, all the digits of d except the last $r_j s_j$ many are zeroes. Hence, the sum of digits of $k + d$ is equal to the sum of the digits of k and d combined. That is, $S_{p_j}(k) + S_{p_j}(d) - S_{p_j}(k+d) = 0$. Applying Kummer's theorem (Theorem 3.4.2) now finishes the proof.

- **$p_j^{r_j}$ divides $\binom{k+d-i}{k-i}$:** By Kummer's theorem (Theorem 3.4.2), it suffices to show that

$$\frac{S_{p_j}(d) + S_{p_j}(k-i) - S_{p_j}(k+d-i)}{p_j - 1} \geqslant r_j.$$

  We note that $S_{p_j}(k+d-i) = S_{p_j}(k) + S_{p_j}(d-i)$ by the same argument as the above paragraph. In addition, we have the trivial bounds $S_{p_j}(d) \geqslant 1$ and $S_{p_j}(d-i) \leqslant (p_j - 1) r_j s_j$. Finally, we give a lower bound for $S_{p_j}(k-i)$. Since k has at least $3 r_j s_j$ trailing zeroes, we get that $S_{p_j}(k-1) \geqslant S_{p_j}(k) + 3 r_j s_j (p_j - 1) - 1$. But observe that $S_{p_j}(k-i) = S_{p_j}((k-1) - (i-1)) = S_{p_j}(k-1) - S_{p_j}(i-1)$ since the number of trailing $(p_j - 1)$'s of $k-1$ exceeds the total number of (non-zero) digits of $(i-1)$. Therefore, we get

$$\begin{aligned} S_{p_j}(d) + S_{p_j}(k-i) - S_{p_j}(k+d-i) &\geqslant 1 + S_{p_j}(k-1) - S_{p_j}(i-1) - S_{p_j}(k) - S_{p_j}(d-i) \\ &\geqslant 1 + (3 r_j s_j (p_j - 1) - 1) - (p_j - 1) r_j s_j - (p_j - 1) r_j s_j \\ &\geqslant r_j s_j (p_j - 1) \\ &\geqslant r_j (p_j - 1). \end{aligned}$$

This finishes the proof of Claim 3.4.3 and hence Lemma 3.4.1 and Theorem 1.3.2.

4 Combinatorial list-decoding bound

In this section, we are going to prove the following theorem.

Theorem 1.3.3 (Combinatorial list decoding bound for $\mathcal{P}_d$)**.** *For every Abelian group G and for every constant $\varepsilon > 0$, the space $\mathcal{P}_d$ over any Abelian group G is $(1/2^d - \varepsilon, \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$-list correctable.*

In other words, we will show that for any function $f : \{0,1\}^n \to G$, the number of degree-d polynomials that are $(1/2^d - \varepsilon)$-close to f is $\exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$.

We use the following result of [ABP+24] which gives a naive double-exponential upper bound on the list size. While [ABP+24] prove it for linear polynomials, the same proof extends to higher degree without much change.

Claim 4.0.1 ([ABP+24], Claim 4.1)**.** *For any function $f : \{0,1\}^n \to G$, the number of degree-d polynomials that are $(1/2^d - \varepsilon)$-close to f is at most 2^{2^n}.*

We will subsequently improve the above bound to something independent of n, but to do that, we will need this naive bound. Furthermore, using a result from previous work [ABP+24, Claim 4.2],[12] we know that proving Theorem 1.3.3 for finite Abelian G implies the same theorem for all Abelian G. Hence, we will assume that G is a finite Abelian group. By using the structure theorem of finite Abelian groups, we can decompose G as

$$G \cong G_1 \times G_2,$$

where G_1 is the product of finitely many p-groups where each p is a prime number that is at least p_0 (for some appropriate choice of $p_0 = p_0(d)$ to be fixed later) and G_2 is the product of finitely many p-groups where each p is a prime less than p_0. We provide upper bounds for list-decoding over G_1 and G_2 separately and combine the two bounds to get a final bound on the list size over G. We state the upper bounds formally below, where we use the notation $\mathsf{List}_\varepsilon^f$ to denote the set of degree-d polynomials that are $(1/2^d - \varepsilon)$-close to f.

Theorem 4.0.2 (Combinatorial bound for a product of p-groups where each $p \geqslant p_0$)**.** *Let $d \geqslant 1$ and G be a product of finitely many p-groups, where each $p \geqslant p_0 = 2^{2^{\alpha d^3}}$ for a sufficiently large constant α. Then for every function $f : \{0,1\}^n \to G$, we have $|\mathsf{List}_\varepsilon^f| \leqslant (1/\varepsilon)^{2^{2^{\mathcal{O}(d^3)}}}$.*

In particular, the list size is polynomial in $1/\varepsilon$ for a constant d.

We now state the combinatorial bound for the second case.

Theorem 4.0.3 (Combinatorial bound for a product of p-groups where each $p < p_0$)**.** *Let $d \geqslant 1$ and G be a product of finitely many p-groups, where each $p \leqslant 2^{2^{\mathcal{O}(d^2)}}$. Then for every function $f : \{0,1\}^n \to G$ we have, $|\mathsf{List}_\varepsilon^f| \leqslant \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$.*

Assuming Theorem 4.0.2 and Theorem 4.0.3, we immediately get Theorem 1.3.3 because for any function f with co-domain $G = G_1 \times G_2$ can be written as $f = (f_1, f_2)$ where f_1 has co-domain G_1 and f_2 has co-domain G_2. Furthermore, if $P = (P_1, P_2) \in \mathsf{List}_\varepsilon^f$, then for each $i \in [2]$, P_i must be in $\mathsf{List}_\varepsilon^{f_i}$.

We move on to proving the above two theorems in the next two subsections respectively.

[12] Though this result is only stated for degree 1 in [ABP+24], it works without any change for all degrees.

4.1 Combinatorial bound for a product of p-groups ($p \geqslant p_0$)

Our proof of Theorem 4.0.2 builds upon some of the ideas of the combinatorial bound of [ABP$^+$24] (Theorem 4.4) which handles degree $d = 1$. However, there are various places where higher degree polynomials are not as well-behaved and need more complicated analysis. Indeed the anti-concentration bound (see Lemma 4.1.1 below) we need is more involved and can be of independent interest. Before the full proof, we now give a rough outline of the proof of Theorem 4.0.2. It can be divided into the following two parts.

1. **Anti-concentration of non-sparse polynomials**: Suppose there are two degree-d polynomials P_1 and P_2, both $(1/2^d - \varepsilon)$-close to a function f. Then they must agree with each other on a sufficiently large fraction of the domain. Indeed, $\delta(P_1, P_2) \leqslant \delta(f, P_1) + \delta(f, P_2) < 1/2^d + 1/2^d = 1/2^{d-1}$. Hence, the density of the zeroes of $P = P_1 - P_2$ in $\{0,1\}^n$ is greater than $1 - 1/2^{d-1}$. Suppose that it is at least $1 - 1/2^{d-1} + c$ for some constant c.[13] Our main idea here is that this cannot happen for polynomials P with many monomials. In particular, we show that if P has sufficiently large sparsity (defined as the number of non-zero monomials), then this fraction is less than $1 - 1/2^{d-1} + c$. This allows us to reduce the combinatorial bound to the case of just counting polynomials of "small" sparsity (by a small blow-up in the list size). We expand more on this in the second step. Going back to showing the anti-concentration bound itself, we will prove the following.

   Lemma 4.1.1 (Anti-concentration bound for non-sparse polynomials)**.** *For all positive integers d, s and for every Abelian group G in which all the non-zero elements have order greater than $(s+1)!$, the following holds: For every degree-d polynomial $Q(\mathbf{x})$ over G of sparsity at least s, we have*

   $$\Pr_{\mathbf{x} \sim \{0,1\}^n} [Q(\mathbf{x}) \neq 0] \geqslant 1/2^{d-1} - 2^{\mathcal{O}(d^3)}/\sqrt{s}.$$

   When $d = 1$, [ABP$^+$24] show an anti-concentration lemma along the lines of Littlewood and Offord [LO38] which bounds the density of the zeroes of non-sparse linear polynomials by an arbitrarily small constant (as long as the sparsity is large enough). However, even for $d = 2$, this problem is somewhat subtle. For example $P(\mathbf{x}) = x_1 \cdot P'(x_2, \ldots, x_n)$ (where P' is of degree 1 and large sparsity), is always 0 when $x_1 = 0$. In other words, we cannot hope to bound the density of the zeroes of non-sparse degree-2 polynomials by an arbitrarily small constant. Nevertheless, we can still argue that it cannot be much larger than 1/2 (i.e., $1 - 1/2^{d-1}$ when $d = 2$). For this particular example of $P = x_1 \cdot P'$, we note that when $x_1 = 1$, we can defer to the $d = 1$ case to bound the fraction of roots by a small constant (say c) and when $x_1 = 0$, $P(\mathbf{x})$ is always zero; thus the fraction of roots of P over $\{0,1\}^n$ is less than $1/2 + c$, which is what we wanted to prove. We formalize this for general polynomials (of large sparsity) and general degree d. In particular, we rely on an anti-concentration bound of Meka, Nguyen and Vu [MNV16] when P has many disjoint (non-zero) monomials of degree d – in other words, a "d-matching". Otherwise, there has to be a small vertex cover among the monomials and we use this to reduce to the case of smaller degree (and perform an induction on the degree). There is the further complication of the fact that [MNV16] state their results

[13] We will show the existence of such a c in the formal proof.

only for polynomials over the reals whereas our goal is to also prove it over groups without elements of small order. However, we show that we can use a linear-algebraic argument to deduce the same bound for our setting by making use of the anti-concentration statement over the reals.

2. **Counting sparse polynomials**: We want to show that the number of degree-d polynomials $P_1, P_2, \ldots, P_t$ of *constant sparsity* that are $(1/2^d - \varepsilon)$-close to a function f is $\mathrm{poly}(1/\varepsilon)$. The case of $d = 1$ was handled by [ABP+24] by reducing (at least implicitly) to the case of $P_1, \ldots, P_t$ depending on *disjoint* sets of variables and uses the "independence" of such polynomials to get a bound on t. This part of the reduction is more involved for higher degrees. The reduction in [ABP+24] occurs by setting certain subsets of variables to constants. For linear polynomials, [ABP+24] has the advantage that setting one variable cannot make a polynomial zero (assuming it depends on at least two variables). However, even for $d = 2$, we cannot afford to set variables to arbitrary constants. For example, $P(\mathbf{x}) = x_1 \cdot (x_2 + 3x_3 - x_4)$ vanishes if we set $x_1 = 0$. We get around this by analyzing the structure of the polynomials and setting variables in two stages: in each stage we prove that the list size does not change too much. We defer the remaining details about these two stages (and the full argument) to Section 4.1.1.

We now proceed with the proof of Theorem 4.0.2 with all the details. We will start by assuming the anti-concentration lemma (Lemma 4.1.1) and deducing Theorem 4.0.2 in Section 4.1.1. We then show how to prove Lemma 4.1.1 in Section 4.1.2.

4.1.1 Pruning the list

Roughly speaking, we first show that it suffices to bound the number of "sparse polynomials" in the list to get an upper bound on the total list size. From now on, we will use $\mathrm{spars}(P)$ to denote the number of monomials with non-zero coefficient in the polynomial P.

Reducing to counting sparse polynomials. Let $P_1, P_2, \ldots, P_t$ be all the distinct degree-d polynomials that are $(1/2^d - \varepsilon)$-close to f. We consider the following graph G with vertex set $[t]$ and an edge between i and j if and only if $\mathrm{spars}(P_i - P_j) \leqslant s_0$ where $s_0 = 2^{\mathcal{O}(d^3)}$ is large enough so that the probability on the RHS of Lemma 4.1.1 is at least $1/2^{d-1} - 0.5/2^{2d}$. Here we are using the fact for a sufficiently large constant α, the order of all the non-zero elements of G are greater than $p_0 = 2^{2^{\alpha d^3}} \geqslant (s_0 + 1)!$.

We now show that G cannot contain an independent set of size $\ell = 4^d$. Here, we are assuming $t > \ell$ as otherwise we are done. That is, without loss of generality, assume that the polynomials $P_1, P_2, \ldots, P_\ell$ are such that $\mathrm{spars}(P_i - P_j) \geqslant s_0$ for all $i \neq j \in [\ell]$. We will use the following fact.

Lemma 4.1.2 (e.g. [Juk11], Lemma 2.1)**.** *Suppose $A_1, \ldots, A_\ell \subseteq U$ are subsets each of size r such that the pairwise intersections $A_i \cap A_j$ are of size at most r' for all $i \neq j \in [\ell]$. Then the size of the union of the sets $\cup_{i=1}^{\ell} A_i$ is at least $r^2\ell/(r + (\ell - 1)r')$.*

We take U to be $\{0,1\}^n$ and for $i \in [\ell]$, A_i to be any subset of $\{\mathbf{x} \mid f(\mathbf{x}) = P_i(\mathbf{x})\}$ of size $(1 - 1/2^d)2^n$ and apply the above lemma. We note that for any $i \neq j \in [\ell]$, $|A_i \cap A_j| \leqslant r'$ for $r' = 2^n((1 - 1/2^{d-1}) + 0.5/2^{2d})$ by applying Lemma 4.1.1 for $Q = P_i - P_j$ since we have assumed that $\mathrm{spars}(P_i - P_j) \geqslant s_0$. Since $|\cup_{i=1}^{\ell} A_i| \leqslant 2^n$, we obtain

$$2^n \geqslant \frac{((1-1/2^d)2^n)^2 \ell}{(1-1/2^d)2^n + (\ell-1)(1-1/2^{d-1}+0.5/2^{2d})2^n}.$$

Simplifying the above, we get that $\ell < 4^d$. Thus, there is no independent set of size 4^d in G, which in turn implies by Turán's theorem that there is at least one vertex $\nu \in [t]$ of G with degree at least

$$t' \geqslant t/4^d - 1. \tag{6}$$

Let $\nu_1, \nu_2, \ldots, \nu_{t'}$ be distinct neighbors of ν. Then consider the polynomials $Q_i = P_{\nu_i} - P_\nu$ for $i \in [t']$. We note that $\mathrm{spars}(Q_i) \leqslant s_0$ and $\delta(Q_i, f') \leqslant 1/2^d - \varepsilon$ for $f' = f - P_\nu$.

We now bucket the polynomials $Q_1, Q_2, \ldots, Q_{t'}$ based on which subset of variables they depend on[14]. Since the sparsity of each Q_i is at most s_0, it must depend on at most $s_0 d$ variables.

In the next paragraph, we bound the size of each bucket.

Counting sparse polynomials depending on the same set of variables. We will show that the number of polynomials Q that depend (only) on the variables x_i for $i \in I$ for some fixed $I \in \binom{[n]}{\leqslant s_0 d}$ such that Q is $(1/2^d - \varepsilon)$-close to f' is at most $(2/\varepsilon)2^{2^{s_0 d}}$. Suppose that $Q_1, Q_2, \ldots, Q_{t''}$ are such polynomials over the variables indexed by I and we want to show that $t'' \leqslant (2/\varepsilon)2^{2^{s_0 d}}$. Note that if f' also depends only on the variables indexed by I, then we get the bound $2^{2^{s_0 d}}$ by applying Claim 4.0.1. However, in general, f' can depend on variables outside I and this results in an additional $2/\varepsilon$ factor.

To make this precise, we define a function $f'' : \{0,1\}^I \to G$ over just the variables in I as $f''(\mathbf{y}) = f'(\mathbf{z})$ where $\mathbf{z}|_I = \mathbf{y}$ and $\mathbf{z}|_{I^c}$ is a uniformly random Boolean assignment. Let $\mathcal{X}_i$ be the indicator random variable for the event that $\delta(f'', Q_i) \leqslant 1/2^d - \varepsilon/2$ for $i \in [t'']$. Since $\delta(f', Q_i) \leqslant 1/2^d - \varepsilon$, we conclude that with probability at least $\varepsilon/2$, it holds that $\delta(f'', Q_i) \leqslant 1/2^d - \varepsilon/2$ i.e., $\Pr[\mathcal{X}_i = 1] \geqslant \varepsilon/2$. By linearity of expectation, there must be a setting of $\mathbf{z}|_{I^c}$ such that for the corresponding f'', at least $(\varepsilon/2)t''$ indices $i \in [t'']$ exist such that $\delta(Q_i, f'') \leqslant 1/2^d - \varepsilon/2$. But by Claim 4.0.1 this is at most $2^{2^{s_0 d}}$. Hence, we get $t'' \leqslant (2/\varepsilon)2^{2^{s_0 d}}$.

Therefore, there must be at least $t'/((2/\varepsilon)2^{2^{s_0 d}})$ non-empty buckets. Recall that we label each bucket by a subset of variables that the polynomials in that bucket depend on i.e., a subset of $[n]$ of size at most $k = s_0 d$. Thus there must be at least

$$t''' \geqslant t'/((2/\varepsilon)2^{2^{s_0 d}}) \tag{7}$$

non-empty buckets that are labeled with a subset of $[n]$ of size at most k.

[14] We say that a polynomial $Q(\mathbf{x})$ *depends* on x_i if x_i appears in at least one monomial that has non-zero coefficient in Q.

Reducing to the case where the variable sets form a sunflower. We now invoke the sunflower lemma with the sets S_i's below being the labels of those non-empty buckets.

Lemma 4.1.3 ([ER60], Theorem 3)**.** *Suppose $S_1, S_2, \ldots, S_{t'''}$ are distinct subsets of $[n]$ of size at most k with $t''' \geqslant k!(r-1)^k$ for some integer $r \geqslant 3$. Then there exists at least r sets among $S_1, S_2, \ldots, S_{t'''}$ that form a sunflower. That is, there exists distinct indices $i_1, i_2, \ldots, i_r \in [t''']$ and $C \subseteq [n]$ (called the* core *of the sunflower) such that $C = S_{i_{j_1}} \cap S_{i_{j_2}}$ for all $j_1 \neq j_2 \in [r]$, and $S_{i_j} \backslash C$ are non-empty for all $j \in [r]$.*

Using the bound $k \leqslant s_0 d$, we can take

$$r = (t'''/(s_0 d)!)^{1/s_0 d} \tag{8}$$

in the above lemma. Now, since the corresponding buckets are non-empty, we can choose one (arbitrary) polynomial from each bucket that forms the sunflower – suppose, without loss of generality, that $Q_1, Q_2, \ldots, Q_r$ are polynomials depending on variables indexed by subsets $S_1, S_2, \ldots, S_r \subseteq [n]$ respectively such that the S_i's form a sunflower, say with core $C \subseteq [n]$. Furthermore, recall that we have $\delta(f', Q_i) \leqslant 1/2^d - \varepsilon$ for all $i \in [r]$.

Reducing to the case where the variable sets are pairwise disjoint. The main idea is to set the variables in the core of the sunflower at random. However, we will need to do this in two steps.

- For the sake of analysis, we shall relabel the variables indexed by C by $\mathbf{z} = \{z_1, z_2, \ldots, z_{n_0}\}$ arbitrarily (note that an empty core C corresponds to $n_0 = 0$). We also relabel the variables indexed by $S_i \backslash C$ by $\mathbf{y}^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_{n_i}^{(i)}\}$ for $i \in [r]$ and some integers $n_i \geqslant 1$. Then, being a degree-d polynomial, we can express each polynomial $Q_i(\mathbf{x}) = Q_i(\mathbf{z}, \mathbf{y}^{(i)})$ as a polynomial in the $\mathbf{y}^{(i)}$ variables with coefficients being some polynomials over $\mathbf{z}$ variables. That is,

$$Q_i(\mathbf{z}, \mathbf{y}^{(i)}) = \sum_{I \in \binom{[n_i]}{\leqslant d}} (\mathbf{y}^{(i)})^I Q_{i,I}(\mathbf{z}), \tag{9}$$

  where $(\mathbf{y}^{(i)})^I$ denotes the monomial corresponding to taking the product of variables indexed by I and $Q_{i,I}$ is a polynomial of degree at most $d - |I|$. Since Q_i depends on the variables $\mathbf{y}^{(i)}$, there must exist a *non-empty* subset I for which the polynomial $Q_{i,I}(\mathbf{z})$ is non-zero. Thus, we can define the $\mathbf{y}$-degree of each Q_i as the maximum size of an $I \in \binom{[n_i]}{\leqslant d}$ such that $Q_{i,I}(\mathbf{z}) \neq 0$. By pigeonhole principle, there must be at least r/d indices $i \in [r]$ such that the corresponding $\mathbf{y}$-degrees of Q_i's are identical – we will denote this by $d' \in [d]$. Without loss of generality, we will assume that $Q_1, Q_2, \ldots, Q_{r'}$ have $\mathbf{y}$-degree equal to d' for some

$$r' \geqslant r/d. \tag{10}$$

  Now for each $i \in [r']$, let $m(i) \subseteq C$ denote an arbitrary non-zero monomial of $Q_{i,I}(\mathbf{z})$ for an arbitrary $I \in \binom{[n_i]}{d'}$ such that $Q_{i,I} \neq 0$. Note that $m(i)$ can only take at most $2^{|C|} \leqslant 2^{s_0 d}$ values. Thus, by pigeonhole principle, there exist at least

$$r'' \geqslant r'/2^{s_0 d} \tag{11}$$

indices for all which $m(i) = C'$ for some $C' \subseteq C$. Again, without loss of generality, we assume that $Q_1, Q_2, \ldots, Q_{r''}$ are such polynomials.

We now show that there exists an assignment to variables in $C \backslash C'$ such that

$$r''' \geqslant r''\varepsilon/2 \tag{12}$$

many of the respective restricted polynomials $Q'_1, Q'_2, \ldots, Q'_{r'''}$ (here we are again assuming that the first r''' polynomials satisfy this property) depend on at least one variable outside C i.e., on some variable(s) in their respective $\mathbf{y}^{(i)}$. Furthermore, the distance from the corresponding restriction of f'' is small, i.e., $\delta(f''', Q'_i) \leqslant 1/2^d - \varepsilon/2$ for all $i \in [r''']$. For a uniformly random assignment to the variables in $C \backslash C'$, we have that the expected distance $\delta(f''', Q'_i)$ is at most $1/2^d - \varepsilon$. Hence, with probability at least $\varepsilon/2$ over the choice of the random assignment, we have that $\delta(f''', Q'_i) \leqslant 1/2^d - \varepsilon/2$ for each $i \in [r'']$. Now, by linearity of expectation, we conclude that there exists at least one assignment such that $\delta(f''', Q'_i) \leqslant 1/2^d - \varepsilon/2$ holds for at least $r''\varepsilon/2$ many polynomials.

- We note that none of the restricted polynomials Q'_i become zero or become identical to each other since for each i, there exists an $I \in \binom{[n_i]}{d'}$ such that $Q_{i,I}(\mathbf{z})$ remains non-zero. This is because, by construction, it contains the monomial $\mathbf{z}^{C'}$ with a non-zero coefficient as a monomial of maximum degree, and setting the variables outside C' cannot change the coefficient of this monomial. Let $\mathbf{z}' \subseteq \mathbf{z}$ be the variables indexed by C' and let $Q'_{i,I}(\mathbf{z}')$ denote the restriction of $Q_{i,I}(\mathbf{z})$ for each $i \in [r''']$ and $I \in \binom{[n_i]}{\leqslant d}$ corresponding to the above assignment to the variables in $C \backslash C'$. We have

$$Q'_i(\mathbf{z}', \mathbf{y}^{(i)}) = \sum_{I \in \binom{[n_i]}{\leqslant d}} (\mathbf{y}^{(i)})^I Q'_{i,I}(\mathbf{z}').$$

Therefore, for each i, there exists an $I_i \in \binom{[n_i]}{d'}$ for which $Q'_{i,I_i}(\mathbf{z}')$ has a non-zero coefficient for the product of all $\mathbf{z}'$ variables. Thus we conclude that for a random setting of $\mathbf{z}' = \mathbf{a}$, with probability at least $1/2^{\deg(Q'_{i,I_i})}$ it holds that $Q'_{i,I_i}(\mathbf{a}) \neq 0$. But notice that $\deg(Q'_{i,I_i}) \leqslant d - d'$ since $\deg(Q'_i) \leqslant d$ and $|I_i| = d'$. Hence, by linearity of expectation, there exists some assignment $\mathbf{a} \in \{0,1\}^{C'}$ such that at least $r'''/2^{d-d'}$ of the corresponding restricted polynomials (which we will denote by $Q''_i(\mathbf{y}^{(i)}) = Q'_i(\mathbf{z}' = \mathbf{a}, \mathbf{y}^{(i)})$) are non-constant. In particular, the coefficient of $(\mathbf{y}^{(i)})^I$ is non-zero in Q''_i.

We further claim that $\delta(f'''', Q''_i) \leqslant 1/2^{d'} - \varepsilon/2$ where f'''' is the restriction of f''' obtained by setting $\mathbf{z}' = \mathbf{a}$. This follows by recalling that $\delta(f''', Q'_i) \leqslant 1/2^d - \varepsilon/2$ and we are setting $d - d'$ variables to get f'''' and Q''_i. Indeed, $\delta(f'''', Q''_i) \leqslant 2^{d-d'}(1/2^d - \varepsilon/2) \leqslant 1/2^{d'} - \varepsilon/2$ for any choice of $\mathbf{a}$. In turn, this implies that for the setting $\mathbf{z}' = \mathbf{a}$, we have $m \geqslant r'''/2^{d-d'}$ polynomials $Q''_1, Q''_2, \ldots, Q''_m$ of degree at most d' (recall that the $\mathbf{y}$-degree of the polynomials we are considering is d') depending on pairwise disjoint sets of variables $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(m)}$ respectively such that $\delta(f'''', Q''_i) \leqslant 1/2^{d'} - \varepsilon/2$.

Counting polynomials depending on pairwise disjoint variables. For the rest of the analysis, we treat f'''' and the polynomials Q''_i as functions over all the n variables even though we fixed the values of certain variables in the preceding steps. With this setup, note that $\Pr_{\mathbf{x} \in \{0,1\}^n, i \in [m]} \Big[f''''(\mathbf{x}) =$

$Q''_i(\mathbf{x})\Big] \geqslant 1 - 1/2^{d'} + \varepsilon/2$, which implies that with probability at least $\varepsilon/4$ over the choice of $\mathbf{x}$, we have that $\Pr_{i \in [m]}[f''''(\mathbf{x}) = Q''_i(\mathbf{x})] \geqslant 1 - 1/2^{d'} + \varepsilon/4$. Note that for any such $\mathbf{x} \in \{0,1\}^n$, if $Q''_i(\mathbf{x}) = f'''(\mathbf{x})$, then $Q''_i(\mathbf{x}) = Q''_j(\mathbf{x})$ for at least $(1 - 1/2^{d'} + \varepsilon/4)m - 1$ many $j \in [m] \backslash \{i\}$. In particular, this gives

$$\Pr_{\mathbf{x} \in \{0,1\}^n}\left[\exists i \in [m] : \left|\{j \mid Q''_j(\mathbf{x}) = Q''_i(\mathbf{x})\}\right| \geqslant (1 - 1/2^{d'} + \varepsilon/4)m - 1\right] \geqslant \varepsilon/4. \tag{13}$$

We will now prove an upper bound on the same quantity above: fixing an arbitrary $i \in [m]$ and an arbitrary setting to the variables appearing in Q_i, we see that since the polynomials depend on disjoint subsets of variables, the indicator random variables for the events $Q''_j(\mathbf{x}) = Q''_i(\mathbf{x})$ for a uniformly random $\mathbf{x} \in \{0,1\}^n$ are Bernoulli random variables $\mathrm{bern}(p_j)$ for some $p_j \leqslant 1 - 1/2^{d'}$ (here we are applying the Schwartz-Zippel lemma Theorem 2.2.1) and across $j \in [m] \backslash \{i\}$. Thus, by a Chernoff bound, we have

$$\Pr_{\mathbf{x} \in \{0,1\}^n}\left[\left|\{j \mid Q''_j(\mathbf{x}) = Q''_i(\mathbf{x})\}\right| \geqslant (1 - 1/2^{d'} + \varepsilon/4)(m-1)\right] \leqslant \exp(-\Omega(\varepsilon^2 m)). \tag{14}$$

Applying a union bound over $i \in [m]$ for (14) and combining with (13), we get $\varepsilon/4 \leqslant m \exp(-\Omega(\varepsilon^2 m))$, which gives that

$$m \leqslant \mathcal{O}(1/\varepsilon^3). \tag{15}$$

Chaining together the sequence of inequalities in $t, t', t''', r, r', r'', r'''$ and m, and using $s_0 = 2^{\mathcal{O}(d^3)}$ and $\varepsilon \leqslant 1/2^d$, we get the desired bound on the list size:

$$t \leqslant (1/\varepsilon)^{2^{2^{\mathcal{O}(d^3)}}}.$$

This finishes the proof of Theorem 4.0.2 assuming Lemma 4.1.1.

4.1.2 Anti-concentration lemma

We now prove the anti-concentration lemma (Lemma 4.1.1). That is, we will show that there exists an absolute constant $M > 0$ such that for any degree-d polynomial $Q(\mathbf{x})$ of sparsity at least s, over an Abelian group G in which all non-zero elements have order greater than $(s+1)!$, that

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[Q(\mathbf{x} \neq 0)] \geqslant 1/2^{d-1} - M^{d^3}/\sqrt{s}.$$

Note that the above bound is trivial unless $s \geqslant M^{2d^3}$.

The proof proceeds by an induction on d.

The base case. For $d = 1$, we use the known anti-concentration result of Littlewood and Offord [LO38], or rather the subsequent improvement due to Erdős [Erd45]:

Theorem 4.1.4 ([Erd45], Theorem 2 modified)**.** *There exists a constant $B > 0$ such that any degree-1 multilinear polynomial $P(\mathbf{x})$ over the reals with at least r many variables with non-zero coefficients, we have*

$$\Pr_{\mathbf{x}\sim\{0,1\}^n}[P(\mathbf{x}) = 0] \leqslant \frac{B}{\sqrt{r}}.$$

However, we cannot apply the above theorem directly since G need not be the group of real numbers. Nevertheless, using a simple linear algebraic argument, we show in the below claim that the same anticoncentration inequality actually holds as long as the non-zero elements of G have order greater than $s!$.

Claim 4.1.5. *Suppose that every degree-d multilinear polynomial $P(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ with at least r disjoint non-zero monomials of degree d has at most $c \cdot 2^n$ roots over $\{0,1\}^n$ for some $c = c(d,r)$. Then the following holds for every Abelian group G in which the order of all the non-zero elements is greater than $\binom{rd}{\leqslant d}!$: Every degree-d polynomial $Q(\mathbf{x})$ over G with at least r disjoint non-zero monomials of degree d has at most $c \cdot 2^n$ roots over $\{0,1\}^n$.*

For a degree-1 polynomial with sparsity s, we can apply Claim 4.1.5 and Theorem 4.1.4 with $r = s - 1$ to get the base case (as long as M is a large enough constant). This finishes the proof of Lemma 4.1.1 for the base case $d = 1$, assuming Claim 4.1.5.

The induction step. Suppose $d \geqslant 2$ and that the claim is true for all degrees until $d-1$. To then prove it for degree d, we split the analysis into three cases, where the parameters s_1 and s_2 are to be fixed later.

- **Case 1:** There exists at least one variable (say x_1) that is contained in at least s_1 monomials of Q. That is, let

  $$Q(x_1, x_2, \ldots, x_n) = x_1 Q_1(x_2, \ldots, x_n) + Q_2(x_2, \ldots, x_n), \tag{16}$$

  where we have $\deg(Q_1) \leqslant d-1$ and $\text{spars}(Q_1) \geqslant s_1$. We analyze the probability that $Q(\mathbf{x})$ by first setting the variables $x_2, x_3, \ldots, x_n$ and then setting the value of x_1. By induction hypothesis, we have that

  $$\Pr_{(x_2,\ldots,x_n)\sim\{0,1\}^{n-1}}[Q_1(x_2, \ldots, x_n) \neq 0] \geqslant 1/2^{d-2} - M^{(d-1)^3}/\sqrt{s_1} \tag{17}$$

  Interpreting (16) as a linear polynomial in x_1 with coefficients being Q_1 and Q_2, we have that $\Pr[Q(\mathbf{x}) \neq 0] \geqslant \Pr[Q_1(x_2, \ldots, x_n) \neq 0] \cdot \Pr[Q(\mathbf{x}) \neq 0 \mid Q_1(x_2, \ldots, x_n) \neq 0]$. By the DLSZ lemma (Theorem 2.2.1) for degree-1 polynomials, we have that the second factor above $\Pr[Q(\mathbf{x}) \neq 0 \mid Q_1(x_2, \ldots, x_n) \neq 0]$ is at least $1/2$. Combined with (17), we thus get that $\Pr[Q(\mathbf{x}) \neq 0] \geqslant 1/2 \cdot \left(1/2^{d-2} - M^{(d-1)^3}/\sqrt{s_1}\right) \geqslant 1/2^{d-1} - M^{d^3}/\sqrt{s}$, by taking $s_1 = s/M^{2d^2}$ and M a sufficiently large constant.

- **Case 2:** There exist $s_2 = M^{d^2}$ many disjoint monomials of degree d of Q (with non-zero coefficients), again assuming M is sufficiently large. We now apply the following anticoncentration result of Meka, Nguyen and Vu [MNV16] which can be thought of as a generalization of [LO38, Erd45] to higher degree when there are many disjoint monomials of maximal degree:

Theorem 4.1.6 ([MNV16], Theorem 1.6 modified)**.** *There exists a constant $B > 0$ such that for any degree-d multilinear polynomial $P(\mathbf{x})$ over the reals with at least r many disjoint degree-d monomials with non-zero coefficients, we have*

$$\Pr_{\mathbf{x}\sim\{0,1\}^n}[P(\mathbf{x}) = 0] \leqslant \frac{Bd^{4/3}\sqrt{\log r}}{r^{1/(4d+1)}}.$$

Applying the above theorem for $r = s_2 = M^{d^2}$ and taking M sufficiently large (as a function of B), we see that

$$\Pr[Q(\mathbf{x}) = 0] \leqslant \frac{Bd^{4/3}\sqrt{\log s_2}}{s_2^{1/(4d+1)}} \leqslant 1/2 \leqslant 1 - 1/2^{d-1}. \tag{18}$$

However, Theorem 4.1.6 as stated only holds over the reals. Regardless, using Claim 4.1.5, we know that the same bound applies if all the non-zero elements of G are of order greater than $\binom{rd}{\leqslant d}!$. Recall that we are already assuming that the non-zero elements of G are of order greater than $(s+1)!$ and that $s \geqslant M^{2d^3}$ from the hypothesis of Lemma 4.1.1. Hence, all that remains to be checked is that $\binom{rd}{\leqslant d}! \leqslant (s+1)!$ for the above choice of $r = s_2 = M^{d^2}$. Indeed this inequality holds for sufficiently large M. This finishes the proof of Lemma 4.1.1 in Case 2.

- **Case 3:** Suppose neither Case 1 nor Case 2 occur. Using a greedy algorithm that repeatedly picks disjoint monomials of degree d for as long as possible, we can find a vertex cover of size at most $s_2 d$ among the non-zero monomials of Q of degree d. That is, there exists at most $s_2 d$ many variables (call them $\mathbf{y}$) such that any non-zero monomial of Q of degree d contains at least one of these variables.

  We will analyze the probability that $Q(\mathbf{x})$ is zero by setting the $\mathbf{y}$ variables arbitrarily. Let Q' denote the polynomial in the variables $\mathbf{x}\backslash\mathbf{y}$ after an arbitrary assignment to those of $\mathbf{y}$. Note that $\deg(Q') \leqslant d-1$, since we set at least one variable in each monomial of degree d. It suffices to show that Q' is non-zero to get that $\Pr_{\mathbf{z}}[Q'(\mathbf{z}) \neq 0] \geqslant 1/2^{d-1}$ by a direct use of the DLSZ lemma. Since Q has at least s monomials and each variable is contained in at most s_1 monomials, the total number of monomials containing at least one variable from y is at most $|\mathbf{y}| \cdot s_1 \leqslant s_1 s_2 d$. Hence, even upon setting the variables in $\mathbf{y}$, at least $s - s_1 s_2 d$ monomials of Q remain unaffected. However, the monomials that do get affected can cancel out these monomials. Nevertheless, there would be at least $s - 2s_1 s_2 d = s - 2M^{d^2} ds/M^{2d^2} = s(1 - 2d/M^{d^2}) > 0$ non-zero monomials in Q'. Thus $\Pr_{\mathbf{x}}[Q(\mathbf{x}) \neq 0] \geqslant 1/2^{d-1}$.

This concludes the proof of Lemma 4.1.1 assuming Claim 4.1.5, which we prove below.

Proof of Claim 4.1.5. We first reduce the number of variables of Q from $n = |\mathbf{x}|$ to rd by setting the variables that are *not* part of the r disjoint disjoint degree-d monomials of Q arbitrarily. It then suffices to show that this restricted polynomial, which we shall denote by $Q'(\mathbf{y})$, has at most $c \cdot 2^{rd}$ roots over $\{0,1\}^{rd}$. Towards a contradiction, let $S \subseteq \{0,1\}^{rd}$ be a subset of size greater than $c \cdot 2^{rd}$ such that Q' evaluates to 0 on all the points in S. Consider the following $\{0,1\}$-valued matrix M of dimensions $\ell \times |S|$ where $\ell = \binom{rd}{\leqslant d}$: the (i,j)-th entry of M denotes the evaluation of the i-th monomial at the j-th point in S. Let $a_1, a_2, \ldots, a_r \in [\ell]$ denote the rows corresponding to the disjoint monomials of Q' and for each $i \in [\ell]$, let $e_i \in \{0,1\}^\ell$ denote the vector that takes value 1 at the i-th index and 0 everywhere else.

We claim that there exists at least one index $k \in [r]$ such that e_{a_k} lies in the column span of M, *when M is treated as a matrix over $\mathbb{R}$.* We prove this by contradiction. That is, suppose that $e_{a_k} \notin V$ for all $k \in [r]$, where $V \subseteq \mathbb{R}^\ell$ denotes the column space of M. Note that given any vector $\mathbf{u} \in \mathbb{R}^\ell$, we can uniquely express it as $\mathbf{u} = \mathbf{u}^\parallel + \mathbf{u}^\perp$ such that $\mathbf{u}^\parallel \in V$ and $\mathbf{u}^\perp \perp V$ (i.e., $\left\langle \mathbf{u}^\perp, \mathbf{v} \right\rangle = 0$ for all $\mathbf{v} \in V$. Denoting the orthogonal subspace of V by $V^\perp$, this is equivalent to $\mathbf{u}^\perp \in V^\perp$). Since $e_{a_k} \notin V$, we have that $e_{a_k}^\perp \neq \mathbf{0}$. Hence, $\{\mathbf{x} \in V^\perp \mid \left\langle \mathbf{x}, e_{a_k}^\perp \right\rangle = 0\}$ is a subspace of $V^\perp$ of co-dimension 1. Since a finite union of subspaces of co-dimension 1 is never equal to the ambient vector space over $\mathbb{R}$ (which is $V^\perp$ in our case), we conclude that there exists a vector $\mathbf{p} \in V^\perp$ such that $\left\langle \mathbf{p}, {e_{a_k}}^\perp \right\rangle \neq 0$ for all $k \in [r]$. Since $\left\langle \mathbf{p}, e_{a_k}^\parallel \right\rangle = 0$ as the two vectors are in orthogonal subspaces, we get that $\langle \mathbf{p}, e_{a_k} \rangle \neq 0$ for all $k \in [r]$. Moreover, since $\mathbf{p}$ is orthogonal to the columnspace of M, we get that $\langle \mathbf{p}, M_j \rangle = 0$ where $M_j \in \{0,1\}^\ell$ denotes the j-th column of M for every $j \in [|S|]$. Let $P(\mathbf{y}) \in \mathbb{R}[\mathbf{y}]$ denote the polynomial with coefficients represented by $\mathbf{p}$. Then, the above inner product relations imply that P is a degree-d polynomial with r disjoint non-zero monomials yet it vanishes on S. This is a contradiction to the assumption in the claim statement since $|S| > c \cdot 2^{rd}$. Hence, indeed there exists $k \in [r]$ such that e_{a_k} is spanned the columns of M.

Suppose

$$e_{a_k} = \sum_{p=1}^{t} \alpha_p M_{j_p}, \tag{19}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_t) \in \mathbb{R}^t$ and $M_{j_1}, M_{j_2}, \ldots, M_{j_t}$ are linearly independent columns of M for some $t \leqslant \ell$. Let us denote the submatrix indexed by the columns $j_1, j_2, \ldots, j_t$ by $M' \in \mathbb{R}^{\ell \times t}$. Since M' is full-column-rank, let $i_1, i_2, \ldots, i_t \in [\ell]$ be the indices of the rows of M' that are linearly independent – we shall denote the corresponding submatrix of M' by $M'' \in \mathbb{R}^{t \times t}$. Let $\mathbf{v}' = e_{a_k}$ and $\mathbf{v}'' \in \{0,1\}^t$ be the restriction of $\mathbf{v}'$ to the indices $i_1, i_2, \ldots, i_t$. From (19), we have $M'\boldsymbol{\alpha} = \mathbf{v}'$, and hence $M''\boldsymbol{\alpha} = \mathbf{v}''$. Since M'' is invertible, by Cramer's rule, we have that $\alpha_p = \frac{\det(M''_p)}{\det(M'')}$ for all $p \in [t]$, where $\det(\cdot)$ denotes the determinant and M''_p is the matrix M'' with its p-th column replaced by the vector $\mathbf{v}''$. By multiplying $\det(M'')$ on both sides of (19), we see that there exist integers $\beta_0 \in [t!]$ and $\beta_1, \ldots, \beta_t \in \{-t!, \ldots, t!\}$ such that $\beta_0 e_{a_k} = \sum_{p=1}^t \beta_p M_{j_p}$, where we are using the fact that the determinant of any $\{0,1\}^{t \times t}$ matrix lies in $\{-t!, \ldots, t!\}$.

We can use this to argue about the polynomial Q' defined above over the group G. Assume that the coefficient of the ith monomial in Q' is $Q'_i \in G$. Recall that Q' evaluates to 0 on all points in S and that Q' has disjoint monomials corresponding to rows $a_1, \ldots, a_r \in [\ell]$, implying that $Q'_{a_j} \neq 0$ for each $j \in [r]$. The previous paragraph implies that $\beta_0 Q'_{a_k} = \sum_{p=1}^t \beta_p Q'(\mathbf{x}_p)$, where $Q'_{a_k} \in G$ denotes the coefficient of the corresponding monomial in Q' and $\mathbf{x}_p$ is the p-th point in S (following the same indexing as the columns of M). Since Q' vanishes on S, we have $\beta_0 Q'_{a_k} = 0$. This is a contradiction since $\beta_0 \leqslant t! \leqslant \binom{rd}{\leqslant d}!$ and we assumed that all non-zero elements of G have order greater than $\binom{rd}{\leqslant d}!$. ■

4.2 Combinatorial bound for a product of p-groups ($p < p_0$)

In this subsection, we prove a combinatorial bound on the list size in case of Abelian groups G that are products of p-groups for small p, i.e. Theorem 4.0.3. We say that G is a $(< p_0)$-*group* if every prime factor of $|G|$ is smaller than p_0 (or equivalently, that G is a finite product of p-groups where $p < p_0$).

The proof of the combinatorial bound in this case is quite different from results that prove similar statements when the domain is $\mathbb{Z}_p^n$ for constant p [GKZ08, BL18]. In particular, we depart from the analytic ideas of [BL18] and use combinatorial and algebraic techniques based on monomial orderings and the well-known technique of 'fingerprinting' (see, e.g. [GH00]).

Monomial ordering and leading monomials. We fix variables $x_1, \ldots, x_n$ and define an ordering among (not necessarily multilinear) monomials in these variables as follows. Given monomials m_1, m_2, we say that $m_1 \preceq m_2$ if $\deg(m_1) < \deg(m_2)$ or $\deg(m_1) = \deg(m_2)$ and for the least i such that the two monomials differ in the exponent of x_i, we have a lower power of x_i dividing m_1 than m_2. This is also called the *graded lexicographic order*.[15] Some examples are as follows.

$$x_2x_3^2 \preceq x_1^4, \quad x_1x_2x_3^2 \preceq x_1x_2^2x_3$$

Now, given a polynomial $P \in \mathcal{P}_d(\{0,1\}^n, G)$ for a group G, we define its *leading monomial* $\mathrm{LM}(P)$ to the largest monomial (w.r.t. $\preceq$) with non-zero coefficient in P. We identify $\mathrm{LM}(P)$ with the set $S \subseteq [n]$ of size at most d indexing the variables that appear in it.

We will use monomial orderings in the setting when $G = \mathbb{Z}_p$ to show that many distinct polynomials cannot agree with the same function at too many points. This uses crucially the following tail bound, which we view as independently interesting. It is proved using the fairly standard technique of using 'footprints' [GH00] and an idea for proving tail bounds by Panconesi and Srinivasan [PS97]. As far as we know, such a bound for low-degree polynomials has not been observed before.

Lemma 4.2.1 (Tail bound for degree-d polynomials). *Fix any field $\mathbb{F}$ and any integer $d \geqslant 1$. Let $S_1, \ldots, S_t \in \mathcal{P}_d(\{0,1\}^n, \mathbb{F})$ be polynomials such that $\mathrm{LM}(S_i) \cap \mathrm{LM}(S_j) = \varnothing$ for every distinct $i, j \in [t]$. Then for every $\eta > 0$, we have*

$$\Pr_{\mathbf{a}}\left[|\{i \in [t] \mid S_i(\mathbf{a}) = 0\}| \geqslant \left(1 - \frac{1}{2^d} + \eta\right) \cdot t\right] \leqslant \exp(-\Omega(\eta^2 t)). \tag{20}$$

The lemma is proved in Section 4.2.1 below.

We now outline the proof of the main theorem (Theorem 4.0.3) of this section, which involves several steps.

1. The first, and more involved, step is to prove the bound in the case that $G = \mathbb{Z}_p$ where $p < p_0$ is prime. The proof in this case splits into three smaller steps.
   (a) **Pigeonhole argument**: Fix an $f : \{0,1\}^n \to \mathbb{Z}_p$. Given a list of L polynomials $P_1, \ldots, P_L \in \mathcal{P}_d(\{0,1\}^n, \mathbb{Z}_p)$ that are $(\frac{1}{2^d} - \varepsilon)$-close to f, we show how to obtain a

[15] Any graded monomial order in the sense of [CLO15, Section 2, Chapter 2] will do just as well.

sub-list of size $\ell = \Omega(\log_p L)$ polynomials $Q_1, \ldots, Q_\ell$ that moreover satisfy the property that their leading monomials are *distinct*.

   (b) **Sunflower lemma**: We then apply the Sunflower lemma (Lemma 4.1.3) to the monomials $\mathrm{LM}(Q_1), \ldots, \mathrm{LM}(Q_\ell)$ to find a subset of $t = \Omega_d(\ell^{1/d})$ polynomials from $\{Q_1, \ldots, Q_\ell\}$ whose leading monomials form a *sunflower*, i.e. they are pairwise disjoint except for their common intersection.[16]

   (c) **Using the tail bound**: We can now apply the tail bound (Lemma 4.2.1) stated above and a simple combinatorial argument to show that $t \leqslant \mathrm{poly}(1/\varepsilon)$. Overall, this leads to a bound of $L \leqslant \exp\left(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)}\right)$, concluding the proof in the prime case.

2. **Modifying [DGKS08b]**: The second step is to 'lift' the list bound L that holds for $\mathbb{Z}_p$ ($p < p_0$) to a list bound that holds over all the finite Abelian ($< p_0$)-groups G. In the linear ($d = 1$) case handled in [ABP+24], this was done using the work of [DGKS08b], which gives a combinatorial characterization for such a lifting. This combinatorial property holds for the space of linear polynomials, implying (using [DGKS08b]) that a list bound of L in the prime case 'lifts' to a list bound of $\mathrm{poly}(L)$ for all ($< p_0$)-groups G.

   Unfortunately, for $d > 1$, the characterization of [DGKS08b] is not applicable as stated. However, we show that a simpler proof allows us to recover a weaker bound of $L^{\mathcal{O}(\log(1/\varepsilon))}$. Using the bound for the prime case, we again get a bound of $\exp\left(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)}\right)$ on the list size for any Abelian ($< p_0$)-group G.

In the rest of the section, we show how to carry out the above strategy. We start with the proof of Lemma 4.2.1 in Section 4.2.1, followed by the prime case in Section 4.2.2 and the case of Abelian ($< p_0$)-groups in Section 4.2.3.

4.2.1 Proof of the tail bound

To derive the tail bounds (Lemma 4.2.1) we will need the following theorem of Panconesi and Srinivasan [PS97], which is an extension of Chernoff-Hoeffding bound.

Theorem 4.2.2. *[PS97, Theorem 3.4] Let $Z_1, \ldots, Z_t$ be Boolean random variables such that for some $\alpha \in [0, 1]$ and for every subset $I \subseteq [t]$, we have $\Pr\left[\wedge_{i \in I} Z_i = 1\right] \leqslant \alpha^{|I|}$. Then, we have the following tail bound.*

$$\Pr\left[\sum_{i=1}^{t} Z_i \geqslant (\alpha + \eta) \cdot t\right] \leqslant \exp(-\Omega(\eta^2 t)).$$

Proof of Lemma 4.2.1. We start by defining the Boolean random variables $Z_1, \ldots, Z_t$ as follows: For $i \in [t]$, $Z_i = 1$ exactly when S_i is equal to 0, i.e. $Z_i(\mathbf{a}) = 1$ iff $S_i(\mathbf{a}) = 0$. To use Theorem 4.2.2, we need to show the following:
For each subset $I \subseteq [t]$,

$$\Pr[\wedge_{i \in I} Z_i = 1] \leqslant \alpha^{|I|},$$

where $\alpha := \left(1 - \frac{1}{2^d}\right)$.

[16] The recently improved sunflower lemma due to Alweiss, Lovett, Wu, and Zhang [ALWZ21] unfortunately does not lead to significantly improved parameters here, and so we stick to the classical version.

For simplicity in notation, assume that $I = \{1, \ldots, r\}$ and define the $\mathrm{Zero}_I \subseteq \{0,1\}^n$ as the set of common zeroes of $S_1, \ldots, S_r$ in $\{0,1\}^n$, i.e.

$$\mathrm{Zero}_I := \{\mathbf{a} \in \{0,1\}^n \mid S_i(\mathbf{a}) = 0, \text{ for all } i \in [t]\}$$

We want to show that $|\mathrm{Zero}_I| \leqslant \alpha^r \cdot 2^n$.

To do this, we use the standard 'footprint bound' (see e.g. [GH00] and [CLO15, §5.3]) which can be seen as a version of the linear algebra method in combinatorics (see, e.g. [BF22]). More precisely, we consider the vector space of functions $g : \mathrm{Zero}_I \to \mathbb{F}$. We denote this vector space by $\mathcal{F}_I$ and we will show that

$$\dim(\mathcal{F}_I) \leqslant \cdot\alpha^r \cdot 2^n. \tag{21}$$

Note that the set of indicator functions for each point in Zero_I is a basis for $\mathcal{F}_I$ and thus $\dim(\mathcal{F}_I)$ is equal to $|\mathrm{Zero}_I|$. Combining it with Equation (21), we will get that $|\mathrm{Zero}_I| \leqslant 2^n \cdot \alpha^r$ and this will finish the proof.

We bound $\dim(\mathcal{F}_I)$ by constructing a spanning set for $\mathcal{F}_I$ of size at most $\alpha^r \cdot 2^n$. The set Zero_I is a subset of the product set $\{0,1\}^n$. We define another set of polynomials and show that their common set of zeroes in $\mathbb{Z}_p^n$ is exactly equal to Zero_I, gaining the advantage of working over a field and use linear algebra. Define the set I' of polynomials as follows:

$$I' := \left\{x_1^2 - x_1, \ldots, x_n^2 - x_n, S_1, \ldots, S_r\right\}$$

It is easy to see that the common set of zeroes of polynomials in I' over $\mathbb{Z}_p^n$ is exactly Zero_I (the polynomial constraint $x_i^2 - x_i$ forces the i^{th} coordinate to be in $\{0,1\}$).

Now, given any function $g : \mathrm{Zero}_I \to \mathbb{F}$, we can express g as a polynomial (e.g. via Theorem 2.2.1). Using a standard division algorithm for multivariate polynomials (see [CLO15, §2.3, Theorem 3]), we can write

$$g = \sum_{i=1}^{n} a_i \cdot (x_i^2 - x_i) + \sum_{j=1}^{r} b_j S_j + R \tag{22}$$

where $a_1, \ldots, a_n, b_1, \ldots, b_r \in \mathbb{Z}_p[x_1 \ldots, x_n]$ are some polynomials and $R \in \mathbb{Z}_p[x_1 \ldots, x_n]$ is such that no monomial with non-zero coefficient in R is divisible by any of the leading monomials of the polynomials in the set I'. In other words, R is a linear combination of *multilinear* monomials that are not divisible by $\mathrm{LM}(S_1), \ldots, \mathrm{LM}(S_r)$.[17]

By Equation (22), the polynomials R and g represent the same function over Zero_I. It follows that the set of multilinear monomials that are not divisible by $\mathrm{LM}(S_1), \ldots, \mathrm{LM}(S_r)$ span the space $\mathcal{F}_I$. In particular, the dimension of the vector space $\mathcal{F}_I$ is upper bounded by the number of such multilinear monomials.

Fnially we argue that the number of multilinear monomials not divisible by $\mathrm{LM}(S_1), \ldots, \mathrm{LM}(S_r)$ is at most $\alpha^r \cdot 2^n$. Picking a uniformly random multilinear monomial corresponds to choosing a uniformly random subset $J \subseteq [n]$. The chance that the random multilinear monomial is not divisible by $\mathrm{LM}(S_j)$ ($j \in [r]$) is equal to the chance that the set J does not contain the set corresponding to $\mathrm{LM}(S_j)$ which is at most $\alpha = 1 - \frac{1}{2^d}$ since $|\mathrm{LM}(S_j)| \leqslant d$. Since the leading monomials of

[17] The last couple of paragraphs is essentially a summary of part of [CLO15, §5.3, Proposition 4] stated here for completeness.

$S_1, \ldots, S_r$ are pairwise disjoint, these events are mutually independent, leading to the conclusion that a uniformly random multilinear monomial is not divisible by $\mathrm{LM}(S_1), \ldots, \mathrm{LM}(S_r)$ is at most α^r. This is equivalent to the desired claim.

This proves Equation (21), which concludes the proof of Lemma 4.2.1 as described above. ■

We will use a corollary of this tail bound, which we now prove.

Corollary 4.2.3. *Fix any field $\mathbb{F}$ and any integer $d \geqslant 1$. Let $Q_1, \ldots, Q_t \in \mathcal{P}_d(\{0,1\}^n, \mathbb{F})$ be polynomials such that $\mathrm{LM}(Q_i) \cap \mathrm{LM}(Q_j) = \varnothing$ for every distinct $i, j \in [t]$. Then for every $\eta > 0$, we have*

$$\Pr_{\mathbf{a}}\left[\exists c \in \mathbb{F} \text{ s.t. } |\{i \in [t] \mid Q_i(\mathbf{a}) = c\}| \geqslant \left(1 - \frac{1}{2^d} + \eta\right) \cdot t\right] \leqslant t \exp(-\Omega(\eta^2 \cdot t)).$$

Proof. Without loss of generality, assume that $\mathrm{LM}(Q_1) \lneq \mathrm{LM}(Q_2) \cdots \lneq \mathrm{LM}(Q_t)$.

Assume that $\mathbf{a} \in \{0,1\}^n$ is chosen uniformly at random as in the statement of the corollary. For $j < t$, let $\mathcal{E}_j$ denote the event that there is a $c \in \mathbb{F}$ such that for at least $\left(1 - \frac{1}{2^d} + \eta\right) \cdot t$ many $i \in [t]$, we have $Q_i(\mathbf{a}) = c$ and furthermore that j is the smallest index such that $Q_j(\mathbf{a}) = c$.

The event whose probability we are trying to bound is contained in $\mathcal{E}_1 \cup \mathcal{E}_2 \cdots \cup \mathcal{E}_r$ where $r = t/2^d$.

Fix any $j \leqslant r$. The probability of $\mathcal{E}_j$ is upper bounded by the probability that at least a $\left(1 - \frac{1}{2^d} + \eta\right)$ fraction of the polynomials

$$Q_{j+1} - Q_j, Q_{j+2} - Q_j, \ldots, Q_t - Q_j$$

all simultaneously vanish at the point $\mathbf{a}$. Note that these polynomials have leading monomials $\mathrm{LM}(Q_{j+1}), \ldots, \mathrm{LM}(Q_t)$ respectively, which are pairwise disjoint. Hence the tail bound Lemma 4.2.1 is applicable and we can bound the probability of $\mathcal{E}_j$ by $\exp(-\Omega(\eta^2 \cdot (t - j))) = \exp(-\Omega(\eta^2 \cdot t))$.

The corollary follows by a union bound over the probability of $\mathcal{E}_1, \ldots, \mathcal{E}_r$. ■

4.2.2 Combinatorial bound for the prime case

In this subsection, we prove a combinatorial bound on the list size for degree d polynomials when the coefficients are from $\mathbb{Z}_{|}$ for a prime p. The list size is a constant dependent on d, p, and ε (independent of n).

Lemma 4.2.4. *Fix any $d \geqslant 1$ and any prime p. Then for every function $f : \{0,1\}^n \to \mathbb{Z}_p$ we have, $|\mathsf{List}_\varepsilon^f| \leqslant \exp(\mathcal{O}_{d,p}(1/\varepsilon)^{\mathcal{O}(d)})$.*

To prove this lemma, we follow the outline from the beginning of the section. For the rest of this section, let f be an arbitrary function as in the statement of the lemma and let $\mathsf{List}_\varepsilon^f = \{P_1, \ldots, P_L\}$ where $L \in [p^\ell, p^{\ell+1})$ for an integer ℓ. Note that $\ell = \Omega(\log_p L)$.

Pigeonhole argument. We start with a pigeonhole argument that allows us to find a sub-list of polynomials from $\mathsf{List}_\varepsilon^f$ that have distinct leading monomials. More formally we prove the following.

Claim 4.2.5. *For each non-negative integer $i \leqslant \ell$, there is a function $f_i : \{0,1\}^n \to \mathbb{Z}_p$ and a set of polynomials $\mathcal{Q}_i = \{Q_1, \ldots, Q_i\} \cup \mathcal{Q}'_i$ such that*

1. $\mathcal{Q}_i \subseteq \mathsf{List}_\varepsilon^{f_i}$,
2. $\mathrm{LM}(Q_1) \gneq \cdots \gneq \mathrm{LM}(Q_i) \gneq \mathrm{LM}(Q)$ *for each $Q \in \mathcal{Q}'_i$, and*
3. $|\mathcal{Q}'_i| \geqslant p^{\ell - i}$.

Proof. The proof is via induction on i. For the base case $i = 0$, we can simply take $f_0 = f$ and $\mathcal{Q}_0 = \mathsf{List}_\varepsilon^f$.

Assuming the statement for $i < \ell$, we prove it for $i + 1$ as follows. We define the 'plurality polynomial' $\mathrm{Pl'}_i$ as follows: for every multilinear monomial m, the coefficient of m in $\mathrm{Pl'}_i$ is the plurality of the coefficient of m among all the polynomials in $\mathcal{Q}'_i$, where we break ties arbitrarily.

Define the function $f_{i+1} := f_i - \mathrm{Pl'}_i$ and define $\mathcal{Q}''_i := \{Q - \mathrm{Pl'}_i \mid Q \in \mathcal{Q}'_i\}$. Using f_{i+1} and $\mathcal{Q}''_i$, we define $\mathcal{Q}_{i+1}$ next:

$$\mathcal{Q}_{i+1} := \{Q_1 - \mathrm{Pl'}_i, \ldots, Q_i - \mathrm{Pl'}_i\} \cup \{Q_{i+1}\} \cup \mathcal{Q}'_{i+1},$$

where

- Q_{i+1} is chosen to be any polynomial in $\mathcal{Q}''_i$ whose leading monomial m is as large as possible among the leading monomials of polynomials in $\mathcal{Q}''_i$,
- $\mathcal{Q}'_{i+1}$ is the set of polynomials in $\mathcal{Q}''_i$ that have a leading monomial *strictly smaller* than m, the leading monomial of Q_{i+1}.

We now show that f_{i+1} and $\mathcal{Q}_{i+1}$ satisfy the required properties.

1. It is clear that $\mathcal{Q}_{i+1} \subseteq \mathsf{List}_\varepsilon^{f_i}$ because each polynomial $\tilde{Q} \in \mathcal{Q}_{i+1}$ can be written as $Q - \mathrm{Pl'}_i$ for some $Q \in \mathcal{Q}_i$. Hence $\delta(f_{i+1}, \tilde{Q}) = \delta(f_i, Q) \leqslant \frac{1}{2^d} - \varepsilon$.
2. We have defined $\mathcal{Q}_{i+1} = \{Q_1 - \mathrm{Pl'}_i, \ldots, Q_i - \mathrm{Pl'}_i, Q_{i+1}\} \cup \mathcal{Q}'_{i+1}$. If m_j denotes the leading monomial of Q_j for $j \leqslant i$, we observe that the coefficient of m_j in Pl'_i is 0 since the plurality used in defining $\mathrm{Pl'}_i$ is only taken over the polynomials in $\mathcal{Q}'_i$ all of which have a leading monomial smaller than m_j by the induction hypothesis. In particular, the leading monomial of $Q_j - \mathrm{Pl'}_i$ is also m_j. We thus have

$$\mathrm{LM}(Q_1 - \mathrm{Pl'}_i) \gneq \cdots \gneq \mathrm{LM}(Q_i - \mathrm{Pl'}_i) \gneq \mathrm{LM}(Q_{i+1})$$

   where for the last inequality we again used the inductive hypothesis. Finally, given any $Q \in \mathcal{Q}'_{i+1}$, we have $\mathrm{LM}(Q) \lneq m = \mathrm{LM}(Q_{i+1})$ by definition of $\mathcal{Q}'_{i+1}$.
3. Finally, we note that for any monomial m, the plurality of the coefficients of m among the polynomials in $\mathcal{Q}''_i$ is zero, since we defined $\mathcal{Q}''_i$ by subtracting from each $Q \in \mathcal{Q}'_i$ the 'plurality polynomial' $\mathrm{Pl'}_i$. In particular, the number of polynomials in $\mathcal{Q}''_i$ that have a leading monomial strictly smaller than m (or equivalently, the number of polynomials in $\mathcal{Q}''_i$ where the coefficient of m is 0) is at least $|\mathcal{Q}'_i|/p \geqslant p^{\ell-i-1}$.

This concludes the induction and hence the proof of the claim. ■

Applying Claim 4.2.5 with $i = \ell$, we see that there is a function f_ℓ such that $|\mathsf{List}_\varepsilon^{f_\ell}| \geqslant \ell$ and $\mathsf{List}_\varepsilon^{f_\ell}$ contains polynomials $Q_1, \ldots, Q_\ell$ with distinct leading monomials. The rest of the argument will bound ℓ.

Sunflower lemma. We now come to the second step of the argument, which is an application of the Sunflower lemma Lemma 4.1.3 to the set of leading monomials of $Q_1, \ldots, Q_\ell$ (seen as subsets of the universe $[n]$ of size at most d). By Lemma 4.1.3, there is a sub-collection of $t = \Omega(\ell^{1/d}/d)$ many polynomials such that their leading monomials form a sunflower. Without loss of generality, we assume that these polynomials are $Q_1, \ldots, Q_t$. Similarly, we assume that $\mathrm{LM}(Q_1) \cap \cdots \cap \mathrm{LM}(Q_t) = \{j_1, \ldots, j_k\}$ for some $k \in \{0, \ldots, d-1\}$. Let J denote this core of the sunflower.

For each $i \in [t]$, we can express the polynomial Q_i as a polynomial in the variables in core J, i.e.

$$Q_i(x_1, \ldots, x_n) = \sum_{A \subseteq J} Q_{i,A}(x_j : j \in [n]\backslash J) \cdot x^A$$

where x^A denotes the product of the variables in A. Note that $\deg(Q_{i,J}) \leqslant d - k$ for each $i \in [t]$. We also note that $\mathrm{LM}(Q_{i,J}) = \mathrm{LM}(Q_i)\backslash J$ by the definition of our monomial order and hence the leading monomials of $\mathrm{LM}(Q_{1,J}), \ldots, \mathrm{LM}(Q_{t,J})$ are *pairwise disjoint.*

Using the tail bound. We are now ready to conclude the bound on ℓ (and hence on $L = |\mathsf{List}_\varepsilon^f|$) using Corollary 4.2.3. More precisely we prove the following claim.

Claim 4.2.6. *Assume $i \in [t]$ and $\mathbf{a} \in \{0,1\}^n$ are chosen uniformly at random from their respective domains. Then*

$$\Pr_{i,\mathbf{a}}\left[f_\ell(\mathbf{a}) \neq Q_i(\mathbf{a})\right] \geqslant \frac{1}{2^d} - \frac{\varepsilon}{2} - t \cdot \exp(-\Omega(\varepsilon^2 \cdot t)).$$

The statement of the claim immediately implies that $t = \mathcal{O}(\log(1/\varepsilon)/\varepsilon^2)$ since we have $\Pr_{\mathbf{a}}\left[f_\ell(\mathbf{a}) \neq Q_i(\mathbf{a})\right] = \delta(f_\ell, Q_i) \leqslant \frac{1}{2^d} - \varepsilon$. It therefore suffices to prove the claim, which we do now.

Proof of Claim 4.2.6. We sample $\mathbf{a}$ in two steps: we first sample its projection to the variables indexed by $[n]\backslash J$, which we denote by $\mathbf{a}'$, followed by its projection to the variables indexed by J, which we denote by $\mathbf{a}''$.

Given a fixing $\mathbf{a}'$ of the variables indexed by $[n]\backslash J$, we denote by $f_{\ell,\mathbf{a}'}$ and $Q_{i,\mathbf{a}'}$ the corresponding restrictions of f_ℓ and Q_i ($i \in [t]$) respectively. Note that each of these is a function of the k variables indexed by J and can hence be expressed uniquely as a multilinear polynomial in these variables (Theorem 2.2.1). Moreover, the coefficient of the monomial x^J in $Q_{i,\mathbf{a}'}$ is exactly $Q_{i,J}(\mathbf{a}')$.

We denote by $\mathcal{B} = \mathcal{B}(\mathbf{a}')$ the event that

$$\exists c \in \mathbb{Z}_p \text{ such that } \left|\{i \in [t] \mid Q_{i,J}(\mathbf{a}') = c\}\right| \geqslant \left(1 - \frac{1}{2^{d-k}} + \frac{\varepsilon}{2}\right) \cdot t.$$

Since the polynomials $Q_{1,J}, \ldots, Q_{t,J}$ have degree at most $d - k$ each and their leading monomials are pairwise disjoint, we can apply Corollary 4.2.3 to bound the probability of $\mathcal{B}$ by $t \cdot \exp(-\Omega(\varepsilon^2 t))$.

Fix any $\mathbf{a}'$ such that $\mathcal{B}$ does not occur. In this case, the *multiset* of polynomials $\{Q_{i,\mathbf{a}'} \mid i \in [t]\}$ has the property that no polynomial appears more than $(1 - \frac{1}{2^{d-k}} + \frac{\varepsilon}{2}) \cdot t$ times in it. In particular, when we choose $i \in [t]$ uniformly at random, we see that the probability that $Q_{i,\mathbf{a}'} \neq f_{\ell,\mathbf{a}'}$ is at least $\frac{1}{2^{d-k}} - \frac{\varepsilon}{2}$. Since any pair of distinct functions on k variables disagree at at least a single point, we have

$$\Pr_{i,\mathbf{a}''}\left[f_{\ell,\mathbf{a}'}(\mathbf{a}'') \neq Q_{i,\mathbf{a}'}(\mathbf{a}'')\right] \geqslant \frac{1}{2^k} \cdot \left(\frac{1}{2^{d-k}} - \frac{\varepsilon}{2}\right) \geqslant \frac{1}{2^d} - \frac{\varepsilon}{2}$$

Overall, we have

$$\begin{aligned}\Pr_{i,\mathbf{a}}\left[f_\ell(\mathbf{a}) \neq Q_i(\mathbf{a})\right] &= \Pr_{\mathbf{a}',i,\mathbf{a}''}\left[f_{\ell,\mathbf{a}'}(\mathbf{a}'') \neq Q_{i,\mathbf{a}'}(\mathbf{a}'')\right] \\ &\geqslant (1 - \Pr_{\mathbf{a}'}[\mathcal{B}]) \cdot \left(\frac{1}{2^d} - \frac{\varepsilon}{2}\right) \geqslant \frac{1}{2^d} - \frac{\varepsilon}{2} - \Pr[\mathcal{B}] \\ &\geqslant \frac{1}{2^d} - \frac{\varepsilon}{2} - t \cdot \exp(-\Omega(\varepsilon^2 \cdot t))\end{aligned}$$

proving the claim. ■

As noted above, Claim 4.2.6 implies that $t = \mathcal{O}_d(\log(1/\varepsilon)/\varepsilon^2)$, implying that $\ell = \mathcal{O}_d(t)^d = \mathcal{O}_d(1/\varepsilon)^d$. Since $\ell = \Omega(\log_p L)$, we get $L = \exp(\mathcal{O}_{d,p}(1/\varepsilon)^d)$, proving Lemma 4.2.4.

4.2.3 Combinatorial bound for the general case

In this subsection, we prove Theorem 4.0.3 which we recall below.

Theorem 4.0.3 (Combinatorial bound for a product of p-groups where each $p < p_0$)**.** *Let $d \geqslant 1$ and G be a product of finitely many p-groups, where each $p \leqslant 2^{2^{\mathcal{O}(d^2)}}$. Then for every function $f : \{0,1\}^n \to G$ we have, $|\mathsf{List}_\varepsilon^f| \leqslant \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$.*

To prove Theorem 4.0.3 for any ($< p_0$)-groups G, we follow the argument from [ABP$^+$24]. The main bottleneck (as mentioned also above) is that the work of [DGKS08b], which is useful in 'lifting' the combinatorial bound from the prime case to the case of general groups for degree-1 polynomials, no longer seems to be applicable here. So the main innovation is to give a proof of such a lifting strategy that also works for higher-degree polynomials (albeit with worse parameters).

More precisely we prove the following lemma, that in conjunction with Lemma 4.2.4 immediately implies Theorem 4.0.3.

Lemma 4.2.7. *Fix any $\varepsilon > 0$ and any positive integer d. Let $\mathcal{S}$ be a set of primes such that for each $p \in \mathcal{S}$, we know that the space $\mathcal{P}_d(\{0,1\}^n, \mathbb{Z}_p)$ is $(\frac{1}{2^d} - \varepsilon, L)$-list decodable for some positive integer L. Then, for every finite Abelian group G that is a product of p-groups for $p \in \mathcal{S}$, it holds that $\mathcal{P}_d(\{0,1\}^n, G)$ is $(\frac{1}{2^d} - \varepsilon, L')$-list decodable, where $L' = L^{\mathcal{O}_d(\log 1/\varepsilon)}$.*

The proof of the lemma is inspired by the proof of the analogous combinatorial bound in [DGKS08b], but needs less structure on the codewords in the underlying code.

Proof. Fix any function $f : \{0,1\}^n \to G$. We want to bound the size of $\mathsf{List}_\varepsilon^f$.

We start by defining a sequence of quotient groups of G as follows. We start with $G_0 = G$. Having defined G_i, we define G_{i+1} by fixing any element $h_i \in G_i$ of prime order $p_i \in \mathcal{S}$ (such an element always exists in an Abelian group by a trivial argument, but one can also use Cauchy's theorem (see e.g. [Con]) which applies to all groups) and defining $G_{i+1} = G_i/H_i$ where H_i is the group generated by h_i. We stop when the group G_i is the trivial group containing just the identity element.

Assume that the sequence of groups thus constructed is $G_0 = G, \ldots, G_h = \{0\}$. We define a sequence of functions $f_i : \{0,1\}^n \to G_i$ ($i \in \{0, \ldots, h\}$) inductively by defining $f_0 = f$ and f_{i+1} by

$$f_{i+1}(\mathbf{a}) = f_i(\mathbf{a}) \pmod{G_i}.$$

for each $\mathbf{a} \in \{0,1\}^n$. We also define $\mathsf{List}_\varepsilon^{f_i}$ to be a subset of $\mathcal{P}_d(\{0,1\}^n, G_i)$ in the natural way.

Finally, we define a tree T of height h as follows. For each $i \in \{0, \ldots, h\}$, we add one vertex at level i in T for each $P_i \in \mathsf{List}_\varepsilon^{f_i}$; note that, in particular, there is exactly one vertex at level h. Further, the children of the polynomial P_{i+1} at level $i+1$ in T are (the vertices corresponding to) those polynomials P_i at level i such that

$$P_i(\mathbf{x}) = P_{i+1}(\mathbf{x}) \pmod{H_i}.$$

It will be useful to note that by Theorem 2.2.1, the above equality holds both in terms of the evaluations of the two polynomials and also in terms of their coefficients.

It is easy to see that each P_i at level i is a child of a unique P_{i+1} at level $i+1$, namely the polynomial P_{i+1} defined by the equality above. We thus have indeed defined a tree T. The size of $\mathsf{List}_\varepsilon^{f}$ is upper bounded by the number of leaves (or paths) of T, which we bound using the following claim.

Notation. Recall that each vertex v of the tree is associated to a polynomial P over group G_i where i denotes the level of v. We define by $\mathrm{agr}(v)$ the fraction of points of agreement between P and f_i. Further, let $\rho(v) = \mathrm{agr}(v) - \left(1 - \frac{1}{2^d}\right)$.

Claim 4.2.8. *The tree T defined above has the following properties.*

1. *Each non-leaf vertex in T has at most L children.*
2. *If v is a child of u, then $\rho(u) \geqslant \rho(v)$.*
3. *If u has two distinct children v and w, then $\rho(u) \geqslant \rho(v) + \rho(w)$.*

Using the above claim, we can finish the proof of the lemma as follows. Let $\ell = \lceil \log L \rceil$. We show that for any node u with children $v_1, \ldots, v_t$, we have

$$\rho(u)^\ell \geqslant \rho(v_1)^\ell + \cdots + \rho(v_t)^\ell. \tag{23}$$

Assuming this, we get by induction from the root r of T that

$$\rho(r)^\ell \geqslant \sum_{v \text{ a leaf}} \rho(v)^\ell \geqslant \varepsilon^\ell \cdot (\#\text{ of leaves of } T) \geqslant \varepsilon^\ell \cdot |\mathsf{List}_\varepsilon^{f}|$$

where for the second inequality, we used the fact that $\rho(v) \geqslant \varepsilon$ for all nodes v in the tree (by definition of ρ and the properties of the polynomials associated to each v). Since $\rho(r) \leqslant 1$, we immediately get $|\mathsf{List}_\varepsilon^{f}| \leqslant (1/\varepsilon)^\ell = L^{\mathcal{O}(\log 1/\varepsilon)}$ as desired.

It remains to prove Equation (23) and Claim 4.2.8, which we do in that order. To see Equation (23), we assume without loss of generality that $t \geqslant 2$ (otherwise the inequality is trivial) and that $\rho(v_1) \geqslant \rho(v_2) \geqslant \cdots \geqslant \rho(v_t)$. Then we have

$$\rho(v_1)^\ell + \cdots + \rho(v_t)^\ell \leqslant (\rho(v_1) + \rho(v_2))^\ell \leqslant \rho(u)^\ell$$

where the second inequality is a consequence of Item 3 of Claim 4.2.8 and the first inequality follows by examining the expansion of $(\rho(v_1) + \rho(v_2))^\ell$ which contains one term that is $\rho(v_1)^\ell$ and $2^\ell - 1 \geqslant L - 1 \geqslant t - 1$ (Claim 4.2.8 Item 1) terms that are at least $\rho(v_2)^\ell$.

Proof of Claim 4.2.8. Assume u is a vertex at some level $i+1$ in the tree. Then u corresponds to a polynomial P taking values in the group G_{i+1}. Assume that u has children $v_1, \ldots, v_t$. Each child v_j is associated to a polynomial Q_j over G_i.

Let $A(u)$ denote the set of points where P agrees with f_{i+1} and similarly, let $A(v_j)$ denote the set of points where Q_j agrees with f_i. By the definition of the tree T, we have $A(v_j)$ is contained in $A(u)$ for each j and this implies Item 2 of the claim.

To prove the other two items, we need some notation. Let $h_i \in G_i$ be the group element used to define H_i and hence G_{i+1} above. Fix any system of coset representatives $c_1, \ldots, c_M$ of H_i in G_i. Given any $g \in G_i$, we can write g uniquely as

$$g = \hat{g} + g' \cdot h_i \tag{24}$$

where $\hat{g} \in \{c_1, \ldots, c_M\}$ and $g' \in \{0, \ldots, p_i - 1\}$ (since h_i has order p_i).

Hence, for each $j \in [t]$, we can write

$$Q_j(\mathbf{x}) = \sum_{|I| \leqslant d} \alpha_I x^I = \underbrace{\sum_{|I| \leqslant d} \hat{\alpha}_I x^I}_{\hat{Q}_j(\mathbf{x})} + \underbrace{\left(\sum_{|I| \leqslant d} \alpha'_I x^I \right)}_{Q'_j(\mathbf{x})} \cdot h_i.$$

Since h_i has order p_i, we can think of Q'_j as taking values in $\mathbb{Z}_{p_i}$.

We note for any two $j, k \in [t]$, the polynomials $\hat{Q}_j$ and $\hat{Q}_k$ are in fact exactly the same. This is because

$$P(\mathbf{x}) = Q_j(\mathbf{x}) = Q_k(\mathbf{x}) \pmod{H_i}$$

by definition of the tree T, implying that the coefficients of Q_j and Q_k are all the same modulo H_i. By the uniqueness of the decomposition in Equation (24), the polynomials $\hat{Q}_j$ and $\hat{Q}_k$ must therefore be the same. We denote this polynomial $\hat{Q}$. Note that this also implies that Q'_j and Q'_k are distinct for any distinct $j, k \in [t]$, since otherwise $Q_j = Q_k$.

Let $F_i(\mathbf{x}) = f_i(\mathbf{x}) - \hat{Q}(\mathbf{x})$. In a similar way to what we did above, we can write for each $\mathbf{x} \in \{0,1\}^n$,

$$F_i(\mathbf{x}) = \hat{F}_i(\mathbf{x}) + F'_i(\mathbf{x}) \cdot h_i$$

where again we think of F'_i as taking values in $\mathbb{Z}_{p_i}$.

Fix any $\mathbf{x} \in \{0,1\}^n$. We have

$$\mathbf{x} \in A(v_j) \iff f_i(\mathbf{x}) = Q_j(\mathbf{x}) \iff F_i(\mathbf{x}) = Q'_j(\mathbf{x}) \cdot h_i \implies F'_i(\mathbf{x}) = Q'_j(\mathbf{x})$$

where the last equality holds as elements of $\mathbb{Z}_{p_i}$. In particular, this implies that since $Q_j \in \mathsf{List}_\varepsilon^{f_i}$, we must also have $Q'_j \in \mathsf{List}_\varepsilon^{F'_i}$. Using the hypothesized bound of L from the prime case (statement of Lemma 4.2.7), we get $t \leqslant L$ implying the first item of the claim.

Finally, we note using the same reasoning that if $\mathbf{x} \in A(v_j) \cap A(v_k)$ for distinct j and k, we must have $Q'_j(\mathbf{x}) = Q'_k(\mathbf{x})$. Since distinct degree-$d$ polynomials can agree on at most a $(1 - \frac{1}{2^d})$ fraction of inputs (Theorem 2.2.1), we see that $|A(v_j) \cap A(v_k)| \leqslant 2^n \cdot \alpha$ where α denotes $(1 - \frac{1}{2^d})$. We have

$$\begin{aligned}(\rho(u) + \alpha) \cdot 2^n = |A(u)| \geqslant |A(v_j) \cup A(v_k)| &= |A(v_j)| + |A(v_k)| - |A(v_j) \cap A(v_k)| \\ &\geqslant (\rho(v_j) + \alpha) \cdot 2^n + (\rho(v_k) + \alpha) \cdot 2^n - \alpha \cdot 2^n = (\rho(v_j) + \rho(v_k) + \alpha) \cdot 2^n\end{aligned}$$

where we used the fact that $A(u)$ contains $A(v_j)$ and $A(v_k)$ (argued above) for the first inequality. This proves Item 3 and concludes the proof. ■

■

5 Local list correction

In this section, we design a local list corrector (see Definition 2.2.3) for the class $\mathcal{P}_d$ and prove Theorem 1.3.4.

Let G be an Abelian group and $f : \{0,1\}^n \to G$ be any function with oracle access to it. Let $\mathsf{List}_\varepsilon^f$ denote the set of degree d polynomials that are $(1/2^d - \varepsilon)$-close to f, and let $L(\varepsilon) = |\mathsf{List}_\varepsilon^f|$. Recall from Theorem 1.3.3 that $L(\varepsilon) = \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$.

Our local list corrector has two phases, inspired by previous work [GL89, STV01, ABP+24].

- First we construct $L' = \widetilde{\mathcal{O}}(L(\varepsilon/2))$ algorithms with oracle access such that with high probability, for each polynomial in the list $\mathsf{List}_\varepsilon^f$, there exists an algorithm that is δ-close to the polynomial where $\delta < 1/(10 \cdot 2^{d+1})$, i.e. δ is in the unique decoding regime.
- Secondly we apply the (unique) local corrector for the class $\mathcal{P}_d$ from Theorem 1.3.1 on each of the algorithms from the first phase.

We state the main theorem of this section below that describes the first phase of our local list corrector. Recall that $\mathcal{A}^f$ denotes that the algorithm $\mathcal{A}$ has oracle access to the function f.

Theorem 5.0.1 (Approximate oracles)**.** *Fix $n \in \mathbb{N}$, $\varepsilon > 0$. Let $f : \{0,1\}^n \to G$ be any function. There exists a randomized algorithm $\mathcal{A}_1^f$ that makes at most $\mathcal{O}_\varepsilon(1)$ oracle queries and outputs deterministic algorithms $\Psi_1, \ldots, \Psi_{L'}$ satisfying the following property: with probability at least $3/4$, for every polynomial $P \in \mathsf{List}_\varepsilon^f$, there exists a $j \in [L']$ such that $\delta(\Psi_j, P) < 1/(10 \cdot 2^{d+1})$, and moreover, for every $\mathbf{x} \in \{0,1\}^n$, Ψ_j computes $P(\mathbf{x})$ by making at most $\mathcal{O}_\varepsilon(1)$ oracle queries to f. Here $L' = \mathcal{O}(L(\varepsilon/2) \log L(\varepsilon/2)) = \mathcal{O}_\varepsilon(1)$.*

The algorithm $\mathcal{A}_1$ is in fact nearly identical to a similar algorithm from [ABP+24] for the case $d = 1$. The main novelty in this paper is in extending the analysis of the algorithm to the case of larger degrees. In particular, we prove the following technical lemmas, which we believe are independently interesting.

- **A sampling lemma for Hamming slices using subcubes**: Let k be an even integer and C be a subcube of $\{0,1\}^{2k}$ be obtained by partitioning the $2k$ co-ordinates into k pairs uniformly at random and identifying the co-ordinates in each pair (equivalently, we choose a random hash function $h : [2k] \to [k]$ that is 2-to-1 and consider the subcube $C_{0^{2k},h}$ as defined in Section 2). We show that this process has good sampling properties in the sense that the density of a set $S \subseteq \{0,1\}^{2k}$ of vectors of Hamming weight k is roughly preserved in the subcube C. More formally, we show

$$\Pr_{\mathsf{C}}\left[\left|\frac{|S \cap \mathsf{C}|}{\binom{k}{k/2}} - \frac{|S|}{\binom{2k}{k}}\right| \geqslant \frac{1}{k^{\Omega(1)}}\right] \leqslant \frac{1}{k^{\Omega(1)}}.$$

  See Section 5.1.1 for a formal statement and the proof.

- **DLSZ lemma for low-degree polynomials over Hamming slices**: We also give a simple proof of the fact that if a polynomial $P \in \mathcal{P}_d(\{0,1\}^{2k}, G)$ does not vanish on the set of points of Hamming weight k, then it does not vanish at an $\Omega_d(1)$ fraction of it. While there have been many works (see [Sri11, Fil16, FI19, Fil23]) addressing the properties of polynomial functions over slices in recent years (especially over the reals), we do not know if this fact has appeared in the literature before in this generality. See Section 5.1.2 for a formal statement and the proof.

Putting the above two lemmas together, one can easily derive the following consequence. Fix a degree-d polynomial $R \in \mathcal{P}_d(\{0,1\}^{2k}, G)$ that does not vanish on the Hamming slice of weight k. Then, the probability that it vanishes on a random subcube C chosen as above is very small, taking a sufficiently large k. This corollary will play a crucial role in the proof of Theorem 5.0.1. The analogous statement for degree 1 was proved in [ABP+24]. However, the proof there is based on a strategy that utilizes an understanding of the structure of the polynomial R in a way that seems difficult to implement for higher degrees.

We start with the proofs of the above lemmas in Section 5.1 and then give the proofs of Theorem 5.0.1 and Theorem 1.3.4 in subsequent sections (which closely follow the analogous proofs in [ABP+24] modulo the above facts).

5.1 The main technical lemmas

We prove now the main new technical lemmas that we use to prove Theorem 5.0.1. Throughout this section, we use $\{0,1\}^n_m$ to denote strings in $\{0,1\}^n$ of Hamming weight exactly m.

5.1.1 Sampling Hamming slices via subcubes

Throughout this section, we fix an even positive integer k and consider a random k-dimensional subcube $\mathsf{C} \subseteq \{0,1\}^{2k}$ obtained as follows. We partition the $2k$ coordinates into k pairs uniformly at random and identify the coordinates in each pair. Equivalently, we choose a uniformly random map $h : [2k] \to [k]$ that is 2-to-1, and set $\mathsf{C} = C_{0^{2k},h}$ (where the latter subcube is as defined in Definition 2.2.6).

Lemma 5.1.1 (Sampling lemma for Hamming slices)**.** *There is an absolute constant $\eta > 0$ such that for every set $S \subseteq \{0,1\}^{2k}_k$, we have*

$$\Pr_{\mathsf{C}}\left[\left|\frac{|S|}{\binom{2k}{k}} - \frac{|S \cap \mathsf{C}|}{\binom{k}{k/2}}\right| \geqslant \frac{1}{k^\eta}\right] \leqslant \mathcal{O}\left(\frac{1}{k^\eta}\right).$$

Remark 5.1.2. *The above lemma is seen to be tight up to the value of the constant η. Consider $S \subseteq \{0,1\}^{2k}_k$ containing exactly those points that differ in the first two co-ordinates. It is easily seen that $|S| = \Omega\left(\binom{2k}{k}\right)$ and further $S \cap \mathsf{C} = \varnothing$ whenever the random partitioning defining C pairs the first two co-ordinates with each other. The latter event occurs with probability $\Omega(1/k)$.*

The proof is via a standard second moment bound, the main step of which is a bound on the spectral gap of a suitable graph. We start with some notations and then state this bound.

An undirected graph G is an (N, D, λ)-*expander* if it is a D-regular graph on N vertices and the magnitude of the second-largest eigenvalue of its adjacency matrix (in absolute value) is at most $\lambda \cdot D$. We refer to the monograph of Hoory, Linial and Wigderson [HLW06] for a more thorough treatment of expander graphs. We will use primarily the following lemma, due to Alon and Chung [AC88].

Lemma 5.1.3 (Expander Mixing Lemma)**.** *Let $G = (V, E)$ be an (N, D, λ)-expander and let $S, T \subseteq V$ be any sets of density σ and τ respectively. Then, for u a uniformly random vertex of V and v a uniformly random neighbour of u, we have*

$$\left|\Pr_{u,v}[u \in S \wedge v \in T] - \sigma \cdot \tau\right| \leqslant \lambda.$$

We apply the above lemma to a combinatorially defined graph called the *Johnson graph.* We define the undirected graph $J(2k, k, d)$ with vertex set $V = \{0,1\}^{2k}_k$, where two points $\mathbf{a}, \mathbf{b} \in V$ are connected exactly when their Hamming distance $\Delta(\mathbf{a}, \mathbf{b}) = 2d$ [18] (note that this number is always even). The main technical lemma is the following.

Lemma 5.1.4 (Eigenvalues of the Johnson graph)**.** *Let $J(2k, k, d)$ be defined as above and assume that d is such that $|d - k/2| \leqslant C\sqrt{k \log k}$ for $C > 0$ being an absolute constant. Then $J(2k, k, d)$ is a $(\binom{2k}{k}, \binom{k}{d}^2, \frac{A}{k^\eta})$-expander for some absolute constants $A > 0$ and $\eta \in (0,1)$ depending only on C.*

We start by proving the sampling lemma Lemma 5.1.1 assuming Lemma 5.1.4.

Proof of Lemma 5.1.1. Throughout, let σ denote the density of S inside $\{0,1\}^{2k}_k$.

[18] Not to be confused with the degree of polynomials d used in the other sections of the paper.

Recall that $\mathsf{C} = C_{0^{2k},h}$ contains a unique point $x(\mathbf{y}) \in \{0,1\}^{2k}$ for each point $\mathbf{y} \in \{0,1\}^k$. Define an indicator random variable $Z_{\mathbf{y}}$ that is 1 exactly when $x(\mathbf{y}) \in S$. We note that for each $\mathbf{y} \in \{0,1\}^k_{k/2}$, the point $x(\mathbf{y})$ is a uniformly random element of $\{0,1\}^{2k}_k$, implying that $\mathbb{E}[Z_{\mathbf{y}}] = \sigma$ for each $\mathbf{y} \in \{0,1\}^k_{k/2}$.

Let $Z := |S \cap \mathsf{C}| = \sum_{\mathbf{y} \in \{0,1\}^k_{k/2}} Z_{\mathbf{y}}$. We thus have $\mathbb{E}[Z] = \binom{k}{k/2} \cdot \sigma$.

We prove the lemma via a second moment estimate. The variance of Z can be bounded as follows.

$$\mathrm{Var}[Z] \;=\; \sum_{\mathbf{u},\mathbf{v} \in \{0,1\}^k_{k/2}} \mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}]$$

We divide the above sum into two parts depending on $\Delta(\mathbf{u},\mathbf{v})$. Let $I = [k/2 - C\sqrt{k \log k}, k/2 + C\sqrt{k \log k}]$ for a suitably large absolute constant (to be chosen below).

$$\begin{aligned}\mathrm{Var}[Z] &= \sum_{\substack{\mathbf{u},\mathbf{v} \in \{0,1\}^k_{k/2} \\ \Delta(\mathbf{u},\mathbf{v}) \in I}} \mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}] \;+\; \sum_{\substack{\mathbf{u},\mathbf{v} \in \{0,1\}^k_{k/2} \\ \Delta(\mathbf{u},\mathbf{v}) \notin I}} \mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}] \\ &\leqslant \sum_{\substack{\mathbf{u},\mathbf{v} \in \{0,1\}^k_{k/2} \\ \Delta(\mathbf{u},\mathbf{v}) \in I}} \mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}] \;+\; \binom{k}{k/2}^2 \cdot \frac{1}{k^{\Omega(C)}},\end{aligned}$$

where the last inequality follows from standard concentration bounds for sampling without replacement [Hoe63]. Now it remains to upper bound $\mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}]$ for a pair $(\mathbf{u},\mathbf{v}) \in \{0,1\}^k_{k/2} \times \{0,1\}^k_{k/2}$ such that $\Delta(\mathbf{u},\mathbf{v}) \in I$. Fix such a pair $(\mathbf{u},\mathbf{v})$ and note that $\Delta(\mathbf{u},\mathbf{v}) = 2d$ for some integer d such that $|d - (k/4)| \leqslant (C/2) \cdot \sqrt{k \log k}$. Then we have,

$$\mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}] \;=\; \mathbb{E}[Z_{\mathbf{u}} Z_{\mathbf{v}}] - \mathbb{E}[Z_{\mathbf{u}}]\mathbb{E}[Z_{\mathbf{v}}] \;=\; \Pr[x(\mathbf{u}) \in S \wedge x(\mathbf{v}) \in S] - \sigma^2.$$

Moreover, the distribution of the random point $x(\mathbf{v})$ given $x(\mathbf{u})$ can be checked to be the uniform distribution over the neighbours of $x(\mathbf{u})$ in the graph $J(2k, k, 2d)$.[19] We can thus apply Lemma 5.1.4 to see that

$$\mathrm{Cov}[Z_{\mathbf{u}}, Z_{\mathbf{v}}] \leqslant \frac{A}{k^\eta}$$

for some absolute constants $A > 0$ and $\eta \in (0,1)$ depending on C. Continuing the variance computation above, we get

$$\mathrm{Var}[Z] \;\leqslant\; \frac{\binom{k}{k/2}^2 \cdot A}{k^\eta} \;+\; \frac{\binom{k}{k/2}^2}{k^{\Omega(C)}} \;\leqslant\; \frac{\mathcal{O}(1)}{k^\eta} \cdot \binom{k}{k/2}^2$$

for a large enough choice of the constant C (and the corresponding η). Now, by Chebyshev's inequality, we have

$$\Pr_{\mathsf{C}}\left[\left|Z - \sigma \cdot \binom{k}{k/2}\right| \geqslant \frac{1}{k^{\eta/4}} \cdot \binom{k}{k/2}\right] \;\leqslant\; \frac{\mathrm{Var}[Z]}{\frac{1}{k^{\eta/2}} \cdot \binom{k}{k/2}^2} \;\leqslant\; \frac{\mathcal{O}(1)}{k^{\eta/2}}$$

which implies the statement of Lemma 5.1.1. ■

[19] More precisely, $x(\mathbf{u})$ and $x(\mathbf{v})$ differ exactly at the co-ordinates given by $h^{-1}(D)$ where $D \subseteq [k]$ is the set of $2d$ co-ordinates where $\mathbf{u}$ and $\mathbf{v}$ differ. Given $x(\mathbf{u})$, this is a uniformly random set $D' \subseteq [2k]$ of size $4d$ subject to the constraint that symmetric difference of D' and the support of $x(\mathbf{u})$ has size exactly $4d$.

It remains to prove Lemma 5.1.4, which we do below.

Proof of Lemma 5.1.4. We use the known exact expressions for the eigenvalues of the Johnson graphs [Del78]. In particular, following [Kar99, Corollary 2], we know that the eigenvalues of the adjacency matrix of $J(2k, k, d)$ are $\beta_0, \ldots, \beta_k$, where for every $0 \leqslant s \leqslant k$,[20]

$$\beta_s = \sum_{r=0}^{s} (-1)^{s-r} \binom{s}{r} \binom{k-r}{k-d-r} \binom{k-s+r}{d-s+r} \tag{25}$$

Furthermore, for $0 \leqslant s \leqslant k$, the eigenvalue β_s has multiplicity $\binom{2k}{s} - \binom{2k}{s-1}$.

Clearly, $\beta_0 = \binom{k}{d}^2$, which is equal to the degree of the graph. It suffices to show that the magnitude of the remaining eigenvalues are all at most $\frac{\mathcal{O}(1)}{k^\eta} \cdot \beta_0$, where η is as in the statement of the Lemma 5.1.4. By assuming that the constant factor in the $\mathcal{O}(1)$ term is large enough, we may assume that k is greater than a large enough absolute constant.

We split the analysis of the remaining eigenvalues into two regimes, the first being when $1 \leqslant s \leqslant C_1$ for a suitably large constant C_1 depending on C (chosen below), and the second when $s > C_1$.

Case 1: $1 \leqslant s \leqslant C_1$. In this case, we analyze β_s using Equation (25). We need the following simple fact about binomial coefficients, which is easily verified from the standard definition using factorials.

Fact 5.1.5. *Fix any non-negative integers $r \leqslant \ell \leqslant k$. Then*

$$\binom{k}{\ell} \cdot \left(\frac{\ell - r}{k - r}\right)^r \leqslant \binom{k-r}{\ell-r} \leqslant \binom{k}{\ell} \cdot \left(\frac{\ell}{k}\right)^r.$$

In particular, we note that for $\ell \in \{d, k-d\}$ and $r \leqslant s \leqslant C_1$, we have

$$\frac{1}{2}\left(1 - \frac{1}{k^{1/4}}\right) \leqslant \frac{\ell - r}{k - r} \leqslant \frac{\ell}{k} \leqslant \frac{1}{2} \cdot \left(1 + \frac{1}{k^{1/4}}\right),$$

and thus for large enough k, we have by Fact 5.1.5

$$\binom{k-r}{k-d-r} = \binom{k}{k-d} \cdot \frac{1}{2^r} \cdot (1 \pm \gamma) \quad \text{and} \quad \binom{k-s+r}{d-s+r} = \binom{k}{d} \cdot \frac{1}{2^{s-r}} \cdot (1 \pm \gamma)$$

for $\gamma = \mathcal{O}(s \cdot k^{-1/4})$. Here the notation $a = b \cdot (1 \pm \gamma)$ denotes that $a \in [b \cdot (1-\gamma), b \cdot (1+\gamma)]$. Plugging this into Equation (25) above, we see that

$$\begin{aligned} |\beta_s| &= \left| \sum_{r=0}^{s} (-1)^{s-r} \binom{s}{r} \cdot \frac{1}{2^r} \cdot \binom{k}{k-d} \cdot (1 \pm \gamma) \cdot \frac{1}{2^{s-r}} \cdot \binom{k}{d} \cdot (1 \pm \gamma) \right| \\ &\leqslant \binom{k}{d}^2 \cdot \frac{1}{2^s} \cdot \left| \sum_{r=0}^{s} (-1)^{s-r} \binom{s}{r} \cdot (1 \pm \mathcal{O}(\gamma)) \right| \end{aligned}$$

[20] The bound in [Kar99] looks slightly different than what is stated here, since the parameter b in that statement is the size of the intersection of the supports of the two points, which implies that $b = k - d$.

$$\leqslant \binom{k}{d}^2 \cdot \frac{1}{2^s} \cdot \left(\left| \sum_{r=0}^{s} (-1)^{s-r} \binom{s}{r} \right| + \mathcal{O}(\gamma) \cdot \sum_{r=0}^{s} \binom{s}{r} \right) \leqslant \binom{k}{d}^2 \cdot \frac{1}{2^s} \cdot \mathcal{O}(\gamma) \cdot 2^s = \binom{k}{d}^2 \cdot \mathcal{O}(\gamma).$$

We have thus shown that $|\beta_s| \leqslant |\beta_0| \cdot \frac{\mathcal{O}(1)}{k^{1/4}}$, proving the required bound in this case.

Case 2: $s > C_1$**.** In this case, we use an argument from [BCIM18], which utilizes just the multiplicity $m_s = \binom{2k}{s} - \binom{2k}{s-1}$ of the eigenvalue β_s. We note that $m_s \geqslant \frac{1}{2} \cdot \binom{2k}{C_1}$ as long as k is large enough.

Let M denote the adjacency matrix of the Johnson graph $J(2k, k, d)$. We know that the squared Frobenius norm of M (i.e. the sum of the squares of the entries of M) is the sum of the squares of its eigenvalues and hence at least $\beta_s^2 m_s$. On the other hand, since M is an adjacency matrix, this quantity is simply the number of edges in the graph $J(2k, k, d)$. In particular, we have

$$\beta_s^2 m_s \leqslant \binom{2k}{k} \cdot \binom{k}{d}^2,$$

yielding the bound

$$|\beta_s| \leqslant \frac{\mathcal{O}(1)}{\binom{k}{C_1}} \sqrt{\binom{2k}{k}} \cdot \binom{k}{d} \leqslant \frac{\mathcal{O}(2^k)}{\binom{k}{C_1}} \cdot \binom{k}{d}.$$

Recall that $|d - k/2| \leqslant C\sqrt{k \log k}$, implying that $\binom{k}{d} \geqslant 2^k \cdot k^{-\mathcal{O}(C^2)}$ by standard binomial estimates.[21] Thus, for C_1 large enough in comparison with C and for k large enough, we see that $|\beta_s| \leqslant \frac{1}{k} \cdot \binom{k}{d}^2 = \frac{\beta_0}{k}$. This finishes the analysis of Case 2 and hence finishes the proof of the lemma. ■

5.1.2 DLSZ lemma over Hamming slices

Lemma 5.1.6 (A DLSZ lemma over Hamming slices)**.** *The following holds for any non-negative integers integers n, k, d where $k \leqslant n$ and $d \leqslant \min\{k, n-k\}$. Let $R \in \mathcal{P}_d(\{0,1\}^n, G)$ be a polynomial such that R does not vanish at some point in $\{0,1\}^n_k$. Then*

$$|\{\mathbf{a} \in \{0,1\}^n_k \mid R(\mathbf{a}) \neq 0\}| \geqslant \binom{n-2d}{k-d}.$$

Proof. As is standard, we proceed by induction on d. The base case $d = 0$ is trivial.

Fix $d \geqslant 1$. Let $R(x_1, \ldots, x_n)$ be a degree d polynomial that does not vanish on all of $\{0,1\}^n_k$.

We can assume that d is *strictly smaller* than $\min\{k, n-k\}$ by the following reasoning. If $d = \min\{k, n-k\}$, the claim reduces to showing that R is non-zero at least one point of $\{0,1\}^n_k$, which trivially follows from the hypothesis. We can also assume that R does not evaluate to the same non-zero value on all of $\{0,1\}^n_k$ since otherwise, the lemma is trivially true.

[21] for example, when $d \leqslant k/2$, we can use [KY15, Lemma 4] with $p = 1/2$ to lower bound $\frac{1}{2^k} \sum_{i \leqslant d} \binom{k}{i}$ which itself is at most $\frac{\mathcal{O}(k)}{2^k} \cdot \binom{k}{d}$

Without loss of generality, we assume that $R(\mathbf{p}) \neq R(\mathbf{q})$ where $\mathbf{p} \in \{0,1\}_k^n$ is the point where the first k co-ordinates are 1 and $\mathbf{q}$ is obtained from $\mathbf{p}$ by flipping the first and last co-ordinates to 0 and 1 respectively (note that this makes sense as $1 \leqslant d \leqslant \min\{k, n-k\}$).

By replacing x_n by the linear polynomial $k - \sum_{i<n} x_i$, we get a degree-d polynomial $R'(x_1, \ldots, x_{n-1})$ not involving the variable x_n but that nevertheless evaluates to the same value as R at every point in $\{0,1\}_k^n$. Moreover, we can assume that R' is multilinear since the inputs are Boolean.

We write

$$R' = x_1 P(x_2, \ldots, x_{n-1}) + Q(x_2, \ldots, x_{n-1}).$$

where P has degree at most $d-1$. Note that $P(\mathbf{p}') \neq 0$ where $\mathbf{p}' \in \{0,1\}_{k-1}^{n-2}$ is the point obtained by restricting $\mathbf{p}$ to its middle $n-2$ co-ordinates, since otherwise $R'(\mathbf{p}) = R'(\mathbf{q})$ in contradiction to our assumptions above.

Thus, we see that P does not vanish on all of $\{0,1\}_{k-1}^{n-2}$. Applying the inductive hypothesis, we see that for

$$S' = \{\mathbf{a}' \in \{0,1\}_{k-1}^{n-2} \mid P(\mathbf{a}') \neq 0\},$$

we have $|S'| \geqslant \binom{n-2-2(d-1)}{k-1-(d-1)} = \binom{n-2d}{k-d}$.

For each $\mathbf{a}' \in S'$, we note that we get at least one distinct point $\mathbf{a} \in \{0,1\}_k^n$ such that $R'(\mathbf{a})$ is non-zero. This is because setting the variables $x_2, \ldots, x_{n-1}$ according to $\mathbf{a}'$ restricts the polynomial R' to a non-zero linear polynomial, which must be non-zero at least one of the extensions of $\mathbf{a}'$ to a point $\mathbf{a} \in \{0,1\}_k^n$.

This shows that R' (and hence R) is non-zero at at least $\binom{n-2d}{k-d}$ many points in $\{0,1\}_k^n$, proving the inductive claim. ■

5.1.3 A useful corollary

We will use Lemma 5.1.1 and Lemma 5.1.6 in the form of the following corollary.

Corollary 5.1.7. *Fix a degree parameter d. Let k be an even positive integer such that $k \geqslant d$, and $R \in \mathcal{P}_d(\{0,1\}^{2k}, G)$ be such that R computes a non-zero function on the slice $\{0,1\}_k^{2k}$. Let $h : [2k] \to [k]$ be a uniformly random 2-to-1 function and define $\mathsf{C} = C_{0^{2k},h}$ (the notation is from Definition 2.2.6). Then if $R|_{\mathsf{C}}$ denotes the natural restriction of R to the subcube C as a polynomial in k variables, we have*

$$\Pr_{\mathsf{C}}[R|_{\mathsf{C}} \text{ vanishes on all of } \{0,1\}_{k/2}^k] \leqslant \mathcal{O}_d\left(\frac{1}{k^\eta}\right)$$

for some absolute constant $\eta > 0$.

Proof. Let $S \subseteq \{0,1\}_k^{2k}$ denote the set of points of $\{0,1\}_k^{2k}$ where R does not vanish. By Lemma 5.1.6, we know that $|S| \geqslant \binom{2k-2d}{k-d} = \frac{1}{2^{\mathcal{O}(d)}} \cdot \binom{2k}{k}$. Note that $R|_{\mathsf{C}}$ vanishes on $\{0,1\}_{k/2}^k$ exactly when $S \cap \mathsf{C} = \varnothing$. By Lemma 5.1.1, the probability of the latter event is at most $\mathcal{O}_d\left(\frac{1}{k^\eta}\right)$. ■

Remark 5.1.8. *We remark that the above corollary is false under the weaker assumption that R is just a non-zero polynomial. A simple example is given by the linear polynomial $\sum_{i=1}^n x_i$ in* *the setting of $G = \mathbb{F}_2$.*[22] *However, the assumption that R is non-zero on $\{0,1\}^{2k}_k$ eliminates this example. Interestingly, this is exactly the condition we need in the analysis of the error-reduction algorithm (Theorem 5.0.1) below (the reason is essentially in Observation 5.2.2 below).*

5.2 The error-reduction algorithm (Theorem 5.0.1)

As mentioned above, the proof now closely follows the proof of the analogous theorem in [ABP+24], which in turn was based on ideas from [STV01].

5.2.1 Preliminaries and useful observations

In this subsection, we give an overview of the algorithms $\mathcal{A}_1^f$ and $\Psi_1, \ldots, \Psi_{L'}$ as mentioned in Theorem 5.0.1.

We first describe a combinatorial construction from [ABP+24] that will be useful in our local list correctors. Given an embedding of a subcube C and a point $\mathbf{b}$, we would like to find a subcube C' such that C and $\mathbf{b}$ are contained in C'. For completeness, we state the definition and observations from [ABP+24] here.

Definition 5.2.1 (Subcube spanned by C and $\mathbf{b}$)**.** *[ABP+24, Definition 9]. Let $\mathsf{C} = C_{\mathbf{a},h}$ be an embedding of a subcube of dimension k (see Definition 2.2.6). For any point $\mathbf{b} \in \{0,1\}^n$, let $\mathbf{v} := \mathbf{a} \oplus \mathbf{b}$. Pick a uniformly random permutation $\sigma : [2k] \to [2k]$. Define a hash function $h' : [n] \to [2k]$ as follows: For all $i \in [n]$,*

$$h'(i) = \begin{cases} \sigma(j), & \text{if } h(i) = j \text{ and } v_i = 0 \\ \sigma(j+k), & \text{if } h(i) = j \text{ and } v_i = 1. \end{cases}$$

For every $\mathbf{z} \in \{0,1\}^{2k}$, $x(\mathbf{z})$ is defined as follows:

$$x(\mathbf{z})_i = z_{h'(i)} \oplus a_i.$$

$\mathsf{C}^{\mathbf{b}}$ is the set of points $x(\mathbf{z})$ for all $\mathbf{z} \in \{0,1\}^{2k}$, i.e. $\mathsf{C}^{\mathbf{b}} := \left\{ x(\mathbf{z}) \;\middle|\; \mathbf{z} \in \{0,1\}^{2k} \right\}$.

Since h' refines the partition induced by h, $\mathsf{C} \subset \mathsf{C}^{\mathbf{b}}$. Also define $\mathbf{w} \in \{0,1\}^{2k}$ as follows: for $j \in [k]$, $w_{\sigma(j)} = 0$ and $w_{\sigma(j+k)} = 1$. Then $x(\mathbf{w}) = \mathbf{b}$, meaning $\mathbf{b} \in \mathsf{C}^{\mathbf{b}}$. Next, we make a couple of observations that will be useful while analyzing the correctness probability of our local list correctors.

Observation 5.2.2. *[ABP+24, Observation 5.2]. Let $\mathbf{a}$ be sampled uniformly from $\{0,1\}^n$ and $h : [n] \to [k]$ be a uniformly random function. For a uniformly random $\mathbf{b} \sim \{0,1\}^n$, $\mathsf{C}^{\mathbf{b}}$ as defined above is a random embedding of a subcube of dimension $2k$ (as defined in Section 2). Note that $\mathbf{b} = x(\mathbf{w})$ for some $\mathbf{w}$ of Hamming weight exactly k.*

[22] In the linear case, [ABP+24] showed that this is essentially the only ‘bad’ example. However, for degrees 2 and higher, there are many more such examples. This is what makes Corollary 5.1.7 more challenging to prove in our setting.

Observation 5.2.3. *Assume that* $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$ *and* $h : [n] \to [k]$ *are chosen independently and uniformly at random and define* C *and* $\mathsf{C}^{\mathbf{b}}$ *as above. Conditioned on the choice of the* $2k$*-dimensional cube* $\mathsf{C}^{\mathbf{b}}$ *(which we identify with* $\{0,1\}^{2k}$*), we may define the distribution of* C *(which is a subcube of dimension* k *in* $\mathsf{C}^{\mathbf{b}}$*) as follows. We sample a random map* $\rho : [2k] \to [k]$ *that is* 2*-to-*1 *(i.e. for each* $j \in [k]$*,* $|\rho^{-1}(\{j\})|$ *is of size exactly* 2*) and identify the variables in each pair. More formally, we set* $\mathsf{C} = C_{0^{2k},\rho}$ *following the notation in Definition 2.2.6.*

Finally, we will use the following *non-local* list decoding algorithm from [ABP+24].

Theorem 5.2.4 ([ABP+24], Theorem A.2)**.** *Fix any Abelian group* G *and degree parameter* d*. There is a* $\mathrm{poly}(2^{n^{d+1}})$*-time algorithm that, given oracle access to a function* $f : \{0,1\}^n \to G$ *produces a list of all polynomials* $P \in \mathcal{P}_d$ *such that* $\delta(f, P) < 1/2^d$*.*

5.2.2 Overview of the algorithms

In this subsection, we give an overview of our local list correction algorithm for $\mathcal{P}_d(\{0,1\}^n, G)$. As mentioned before, our algorithm is mostly similar to the local list correction algorithm for [ABP+24, Section 5.2], except for some changes in the parameters to handle degree d polynomials instead of degree 1 polynomials. Nevertheless, we present the overview and the algorithm for the sake of completeness.

Similar to [STV01], the local list correction algorithm has two key steps. The first step is to produce oracles that approximate the polynomials in the list and the second step is to apply the unique local corrector from Section 3 on each of the approximating oracles. In this subsection, we describe the first step i.e. to produce oracles such that each polynomial in the list is approximated sufficiently well by an oracle.

For the overview below, let $f(x_1, \ldots, x_n)$ be the input function and $\mathsf{List}^f_\varepsilon$ denotes the set as described earlier.

1. **First step (advice):** We construct a randomized algorithm $\mathcal{A}^f_1$ that makes oracle queries to the input function f and produces a list of oracles such that with high probability (over the randomness of $\mathcal{A}^f_1$), for every polynomial $P \in \mathsf{List}^f_\varepsilon$, there exists an oracle that is $1/(10 \cdot 2^{d+1})$-close[23] to P (this is within the unique decoding radius of $\mathcal{P}_d(\{0,1\}^n, G)$). We give an overview of the algorithm $\mathcal{A}^f_1$ below:

   - Choose a random subcube C of dimension k (which is a polynomial in the list size and from the previous section we know that the list size is a constant dependent on ε). Then find all the degree d polynomials that are $(1/2^d - \varepsilon/2)$-close to f on the subcube C. Repeat this step a few times. Let's call this set of polynomials (union over all the repetitions) as T. Since C is a random subcube, with high probability, restriction of every polynomial in $\mathsf{List}^f_\varepsilon$ will be in the set T. The restriction of each $P \in \mathsf{List}^f_\varepsilon$ to C (which is in T) will be *advice* for P in the second step.
   - The algorithm $\mathcal{A}^f_1$ also samples a random permutation on σ on $2k$ variables. The oracles in the next step will use this permutation while locally list correcting (it is the same

[23] This step is essentially an error reduction because we start with a function f that is $(1/2^d - \varepsilon)$-close to P and $\mathcal{A}^f_1$ produces a list of oracles such that one of them is $1/(10 \cdot 2d)$-close to P.

permutation for every oracle). For each polynomial Q in T, the algorithm $\mathcal{A}_1$ produces an oracle $\Psi[\mathsf{C}, \sigma, Q]$ with the polynomial's evaluation on the cube C as advice.

2. **Second step (approximation):** Suppose we want to locally list correct f at a point $\mathbf{b} \in \{0,1\}^n$. Each algorithm $\Psi[\mathsf{C}, \sigma, Q]$ creates a subcube C' of dimension $2k$ that is spanned by C and $\mathbf{b}$. Then the oracle computes all degree d polynomials that have distance at most $(1/2^d - \varepsilon/2)$ from f on the subcube C' and uses the advice on the subcube C to filter out $P(\mathbf{b})$, where $P|_{\mathsf{C}} = Q$.

5.2.3 Formal description of the algorithms

In this subsection, we describe the algorithms that will prove Theorem 5.0.1. The algorithms in this subsection are the same as in [ABP$^+$24, Section 5.2] with minor modifications in the parameters to make it work for the class $\mathcal{P}_d$. We state the algorithms here for the sake of completeness.

Notation: Let C be a k-dimensional subcube of $\{0,1\}^n$ as defined in Definition 2.2.6. Let $Q : \{0,1\}^k \to G$ a polynomial in $\mathcal{P}_d(\{0,1\}^k, G)$ where the polynomial Q is a function on the subcube C. We will use $\Psi[\mathsf{C}, \sigma, Q]$ to denote a *deterministic* algorithm that has the description of the subcube C, a permutation $\sigma : [2k] \to [2k]$, and evaluation of Q on C hardwired inside it[24].

Algorithm 4 is a randomized algorithm that outputs the descriptions of the deterministic oracles and Algorithm 3 describes the oracles themselves.

```
Algorithm 3: Approximating Algorithm $\Psi[\mathsf{C}, \sigma, Q]$
  Input: Oracle access to the function $f$, a point $\mathbf{b} \in \{0,1\}^n$
Let $\mathsf{C}'$ be a subcube spanned by $\mathsf{C}$ and $\mathbf{b}$ using $\sigma$        // see Definition 5.2.1
Let $\mathbf{w} \in \{0,1\}^{2k}$ such that $x(\mathbf{w}) \in \mathsf{C}'$ and $x(\mathbf{w}) = \mathbf{b}$        // $|\mathbf{w}| = k$
Query $f$ on the subcube $\mathsf{C}'$        // Number of queries is $2^{2k}$
Find all polynomials $R_1, \ldots, R_{L''} \in \mathcal{P}_d(\{0,1\}^{2k}, G)$ that are $\left(\frac{1}{2^d} - \frac{\varepsilon}{2}\right)$-close to $f|_{\mathsf{C}'}$
  // using Theorem 5.2.4
        // $L'' \leqslant L(\varepsilon/2)$
if there exists an $i \in [L'']$ such that $R_i|_{\mathsf{C}} = Q$ then
    pick any such $i$ and return $R_i(\mathbf{w})$
else
    return $0$        // An arbitrary value
```

Now we describe the randomized Algorithm 4 that returns the descriptions of the deterministic oracles.

[24] In the final algorithm, C will be a random subcube of dimension $\mathcal{O}_\varepsilon(1)$ and Q with high probability be equal to $P|_{\mathsf{C}}$, for some $P \in \mathsf{List}^f_\varepsilon$

```
Algorithm 4: Algorithm $\mathcal{A}_1$
    Input: Oracle access to the function $f$
Choose $k \leftarrow B_d \cdot (L(\varepsilon/2)/\varepsilon)^c$        // $B_d$ and $c$ chosen below
Set $\ell \leftarrow \log L(\varepsilon)$
$T \leftarrow \varnothing$
repeat
    Sample $\mathbf{a} \sim U_n$ and a random hash function $h : [n] \to [k]$   // the first source of randomness
    Construct the subcube $\mathsf{C} := C_{\mathbf{a},h}$        // see Definition 2.2.6
    Query $f$ on the subcube $\mathsf{C}$        // Number of queries is $2^k$
    Find all polynomials $Q_1, \ldots, Q_{L'} \in \mathcal{P}_d(\{0,1\}^k, G)$ that are $\left(\frac{1}{2^d} - \frac{\varepsilon}{2}\right)$-close to $f|_{\mathsf{C}}$   // using Theorem 5.2.4
    Pick a uniformly random permutation $\sigma : [2k] \to [2k]$      // the second source of randomness
    $T \leftarrow T \cup \{(\mathsf{C}, \sigma, Q_1), \ldots, (\mathsf{C}, \sigma, Q_{L'})\}$        // $L' \leqslant L(\varepsilon/2)$
until $\ell$ times
return $\Psi[\mathsf{C}, \sigma, Q]$ for all $(\mathsf{C}, \sigma, Q) \in T$        // Size of $T$ is $\leqslant \ell L'$
```

5.3 Analysis of the algorithms

In this subsection, we analyze Algorithm 3 and Algorithm 4.

Query complexity: Algorithm 4 makes $2^k = \exp(B_d \cdot \mathrm{poly}(L(\varepsilon/2))$ queries to f and returns the description of $\ell L' = L(\varepsilon/2) \log L(\varepsilon)$ oracles. Each oracle (see Algorithm 3) makes $2^{2k} = \exp(B_d \cdot \mathrm{poly}(L(\varepsilon/2))$ queries to f. Hence the total number of queries to f is $\exp(\mathcal{O}_d(\mathrm{poly}(L(\varepsilon/2)))$.

Correctness: We want to show that with probability $\geqslant 3/4$, for every polynomial $P \in \mathsf{List}^f_\varepsilon$, there exists an output oracle $\Psi[\mathsf{C}, \sigma, Q]$ that is $(1/(10 \cdot 2^{d+1}))$-close to P. We prove this in the following steps.

- In a single iteration for Algorithm 4, the following holds: For every polynomial $P \in \mathsf{List}^f_\varepsilon$, with probability at least $9/10$, there exists a $1/(10 \cdot 2^{d+1})$-close approximating oracle $\Psi[\mathsf{C}, \sigma, Q_j]$. We prove this is in Lemma 5.3.1.
- As we have ℓ independent iterations, the probability that there is no $1/(10 \cdot 2^{d+1})$-close approximating oracle for P is at most $1/10^\ell$. By a union bound for all polynomials $P \in \mathsf{List}^f_\varepsilon$, we get the desired correctness probability in Theorem 5.0.1.
- Since each iteration produces a list of size at most $L(\varepsilon/2)$, overall we obtain a list of size $\mathcal{O}(L(\varepsilon/2) \cdot \log L(\varepsilon))$ as claimed.

We start by proving Lemma 5.3.1, which is the primary lemma for the correctness of our local list correctors.

Lemma 5.3.1 (Correctness of Local List Correction)**.** *The following holds as long as the constants B_d (depending on d) and c in Algorithm 4 are chosen to be large enough. Fix any polynomial $P \in \mathsf{List}^f_\varepsilon$. In each iteration of the loop in Algorithm 4, the probability (over the randomness of the algorithm) that there does not exist a $1 \leqslant j \leqslant L'$ such that $\delta(\Psi[\mathsf{C}, \sigma, Q_j], P) \leqslant 1/(10 \cdot 2^{d+1})$ is at most $1/10$.*

Proof. Fix an iteration of the loop in Algorithm 4. Let $\mathcal{E}_P$ denote the event that there does not exist a j such that $\delta(\Psi[\mathsf{C}, \sigma, Q_j], P) \leqslant 1/(10 \cdot 2^{d+1})$. We want to upper bound the probability of the "bad" event $\mathcal{E}_P$ by $1/10$. Recall that the sources of randomness in Algorithm 4 are point $\mathbf{a}$, hash function h, and permutation σ. We will show that

$$\mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\min_j \delta(\Psi[\mathsf{C}, \sigma, Q_j], P)] = \mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\min_j \Pr_{\mathbf{b}}[\Psi[\mathsf{C}, \sigma, Q_j](\mathbf{b}) \neq P(\mathbf{b})]] \leqslant \frac{1}{100 \cdot 2^{d+1}}, \tag{26}$$

from which Lemma 5.3.1 follows via an application of Markov's inequality.

Define the following auxiliary events, depending on the choice of $\mathbf{a}, h$ and σ, along with the choice of a random input $\mathbf{b}$.

1. Event $\mathcal{E}_{1,P}$ (only depends on $\mathbf{a}, h$): In the current iteration of the loop in Algorithm 4, the algorithm does not find a polynomial Q_j such that $Q_j = P|_{\mathsf{C}}$.
2. Event $\mathcal{E}_{2,P}$ (depends on $\mathbf{a}, h, \sigma, \mathbf{b}$): For the triples (C, σ, Q) added to T in this iteration of Algorithm 4, the corresponding oracle $\Psi[\mathsf{C}, \sigma, Q]$ is such that when we run this oracle on input $\mathbf{b}$, there does not exist a polynomial R_i such that $R_i = P|_{\mathsf{C}'}$. Note that this event only depends on the choice of the cube C, σ and $\mathbf{b}$ but not on the specific polynomial Q. Hence, the event is exactly the same for each triple (C, σ, Q) in T. In particular, we may fix any one such triple.
3. Event $\mathcal{E}_{3,P}$ (depends on $\mathbf{a}, h, \sigma, \mathbf{b}$): For the triples (C, σ, Q) added to T in this iteration of Algorithm 4, the corresponding oracle $\Psi[\mathsf{C}, \sigma, Q]$ is such that when we run this oracle on input $\mathbf{b}$, there exist two polynomials R_{i_1} and R_{i_2} such that $R_{i_1}(\mathbf{w}) \neq R_{i_2}(\mathbf{w})$ but $R_{i_1}|_{\mathsf{C}} = R_{i_2}|_{\mathsf{C}}$. Here $\mathbf{w}$ is, as defined in Algorithm 3, the point in $\{0,1\}^{2k}$ of Hamming weight k such that $x(\mathbf{w}) \in \mathsf{C}'$ and $x(\mathbf{w}) = \mathbf{b}$. As for the event $\mathcal{E}_{2,P}$, we may fix a triple $(\mathsf{C}, \sigma, Q) \in T$ while analyzing this event.

We will need the following two claims that show that any of the aforementioned events occur with a small probability.

Claim 5.3.2. $\Pr_{\mathbf{a},h}[\mathcal{E}_{1,P}], \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{2,P}] \leqslant 1/(10000 \cdot 2^{d+1})$.

Claim 5.3.3. $\Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{3,P}] \leqslant 1/(10000 \cdot 2^{d+1})$.

Let us proceed with the proof of Lemma 5.3.1 assuming the above two claims. We first show that if $\mathcal{E}_P$ occurs, then at least one of the auxiliary events occurs. This implies that it is sufficient to upper bound the probability of the auxiliary events occurring and use a union bound.

For $\mathbf{a}, h$ such that the event $\mathcal{E}_{1,P}$ does not occur, we can fix a $j^* \in [L']$ such that $P|_{\mathsf{C}} = Q_{j*}$. Thus, we have

$$\mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\min_j \Pr_{\mathbf{b}}[\Psi[\mathsf{C},\sigma,Q_j](\mathbf{b}) \neq P(\mathbf{b})]] \leqslant \Pr_{\mathbf{a},h}[\mathcal{E}_{1,P}] + \mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\mathbf{1}_{\neg\mathcal{E}_{1,P}} \cdot \Pr_{\mathbf{b}}[\Psi[\mathsf{C},\sigma,Q_{j*}](\mathbf{b}) \neq P(\mathbf{b})]] \tag{27}$$

Fix any $\mathbf{a}, h$ such that the event $\mathcal{E}_{1,P}$ does not occur. Further, if the event $\mathcal{E}_{2,P}$ does not occur, then there is an $i^* \leqslant L''$ such that $P|_{\mathsf{C}'} = R_{i*}$. In particular, $R_{i*}|_{\mathsf{C}} = P|_{\mathsf{C}} = Q_{j*}$.

Finally, if event $\mathcal{E}_{3,P}$ also does not occur, then there is no $i \neq i^*$ such that $R_{i*}(\mathbf{w}) \neq R_i(\mathbf{w})$ but $R_i|_{\mathsf{C}} = R_{i*}|_{\mathsf{C}}$. In particular, the only possible output of the algorithm $\Psi[\mathsf{C},\sigma,Q_{j*}]$ on input $\mathbf{w}$ is $R_{i*}(\mathbf{w}) = P(x(\mathbf{w})) = P(\mathbf{b})$.

We have thus shown that

$$\mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\mathbf{1}_{\neg\mathcal{E}_{1,P}} \cdot \Pr_{\mathbf{b}}[\Psi[\mathsf{C},\sigma,Q_{j*}](\mathbf{b}) \neq P(\mathbf{b})]] \leqslant \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{2,P} \vee \mathcal{E}_{3,P}] \leqslant \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{2,P}] + \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{3,P}].$$

Plugging the above into Equation (27), we get

$$\mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\min_j \Pr_{\mathbf{b}}[\Psi[\mathsf{C},\sigma,Q_j](\mathbf{b}) \neq P(\mathbf{b})] \leqslant \Pr_{\mathbf{a},h}[\mathcal{E}_{1,P}] + \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{2,P}] + \Pr_{\mathbf{a},h,\sigma,\mathbf{b}}[\mathcal{E}_{3,P}]. \tag{28}$$

Using Claim 5.3.2 and Claim 5.3.3, we get,

$$\mathop{\mathbb{E}}_{\mathbf{a},h,\sigma}[\min_j \Pr_{\mathbf{b}}[\Psi[\mathsf{C},\sigma,Q_j](\mathbf{b}) \neq P(\mathbf{b})] \leqslant \frac{3}{10000 \cdot 2^{d+1}} \leqslant \frac{1}{100 \cdot 2^{d+1}}$$

So now it remains to prove Claim 5.3.2 and Claim 5.3.3. We start with Claim 5.3.2 which essentially follows from Lemma 2.2.7.

Proof of Claim 5.3.2. Recall that $\delta(P, f) \leqslant (1/2^d - \varepsilon)$. Equivalently, the set of points T where f and P differ has density at most $(1/2^d - \varepsilon)$ in $\{0,1\}^n$. For a cube C, the non-existence of Q_j such that $Q_j = P|_{\mathsf{C}}$ is equivalent to $\delta(P|_{\mathsf{C}}, f|_{\mathsf{C}}) > (1/2^d - \varepsilon/2)$.

For a random $\mathbf{a}$ and a random h, the subcube C is a random subcube. Using Lemma 2.2.7 for the subset T as mentioned above, we get that for $k \geqslant 1/\varepsilon^5$,

$$\Pr_{\mathsf{C}}\left[\delta(P|_{\mathsf{C}}, f|_{\mathsf{C}}) > \frac{1}{2^d} - \frac{\varepsilon}{2}\right] \leqslant \frac{1}{10000 \cdot 2^{d+1}}.$$

Here, we are assuming that B_d and c are large enough so that k as defined in Algorithm 4 satisfies the hypothesis of Lemma 2.2.7. Hence $\Pr[\mathcal{E}_{1,P}] \leqslant 1/(10000 \cdot 2^{d+1})$.

Using Observation 5.2.2, we know that for a random $\mathbf{a}, h$ and a random permutation σ, the subcube C' is a random subcube of dimension $2k$ in $\{0,1\}^n$ as required in Lemma 2.2.7. Proceeding as above, we get the stated upper bound on $\Pr[\mathcal{E}_{2,P}]$. ■

Now it remains to prove Claim 5.3.3, which we prove next.

Proof of Claim 5.3.3. We start by conditioning the choice of the subcube C'. This fixes the polynomials $R_1, \ldots, R_{L''}$ obtained in Line 4 of Algorithm 3. Fix any two distinct polynomials $R_{i_1}, R_{i_2} : \{0,1\}^{2k} \to G$ in this list that differ at *at least one point* in $\{0,1\}^{2k}$

of Hamming weight k (in particular, this includes pairs of polynomials that differ at the point $\mathbf{w}$ such that $x(\mathbf{w}) = \mathbf{b}$).

We want to upper bound the probability of the event that $R_{i_1}|_{\mathsf{C}} = R_{i_2}|_{\mathsf{C}}$. In the end, we use a union bound over the number of all possible pairs (R_{i_1}, R_{i_2}).

Define the polynomial $R := R_{i_1} - R_{i_2}$, where $R : \{0,1\}^{2k} \to G$ is a non-zero degree d polynomial, defined on the subcube C'. We want to upper bound the probability that $R|_{\mathsf{C}}$ is identically zero polynomial. Using Observation 5.2.3 and Corollary 5.1.7, we see that the probability of this is at most $\mathcal{O}_d(1/k^{\eta})$ for some absolute constant $\eta > 0$. For k as defined in Algorithm 4 where B_d is large enough (depending on d) and c is a large enough absolute constant, we get a probability of at most $1/(10000 \cdot 2^{d+1} \cdot L(\varepsilon/2)^2)$.

Since $L'' \leqslant L(\varepsilon/2)$, a union bound over all such pairs R_{i_1}, R_{i_2} yields the bound stated in the claim. ■

As discussed above, we have proved Claim 5.3.2 and Claim 5.3.3 and substituting them in Equation (28), we get the desired bound, and this concludes the correctness of the local list correction algorithm. ■

5.4 Local list corrector

Let us now see how Theorem 5.0.1 implies Theorem 1.3.4. Let us first recall Theorem 1.3.4.

Theorem 1.3.4 (Local list correction for $\mathcal{P}_d$)**.** *For every Abelian group G and for every $\varepsilon > 0$, the space $\mathcal{P}_d$ is $(1/2^d - \varepsilon, \mathcal{O}_\varepsilon(1), \widetilde{\mathcal{O}}_\varepsilon(\log n)^d, \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)}))$-locally list correctable.*

Specifically, there is a randomized algorithm $\mathcal{A}$ that, when given oracle access to a polynomial f and a parameter $\varepsilon > 0$, outputs with probability at least $3/4$ a list of randomized algorithms $\phi_1, \ldots, \phi_L$ ($L \leqslant \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$) such that the following holds. For each $P \in \mathcal{P}_d$ that is $(1/2^d - \varepsilon)$-close to f, there is at least one algorithm ϕ_i that, when given oracle access to f, computes P correctly on every input with probability at least $3/4$.

The algorithm $\mathcal{A}$ makes $\mathcal{O}_\varepsilon(1)$ queries to f, while each ϕ_i makes $\widetilde{\mathcal{O}}_\varepsilon((\log n)^d)$ queries to f.

Proof of Theorem 1.3.4. We first run the algorithm given by Theorem 5.0.1 and it outputs $\psi_1, \ldots, \psi_{L'}$ for $L' \leqslant \exp(\mathcal{O}_d(1/\varepsilon)^{\mathcal{O}(d)})$ (by Theorem 1.3.3). Next we run our local correction algorithm for $\mathcal{P}_d$ (see Theorem 1.3.1 and Section 3) with $\psi_1, \ldots, \psi_{L'}$ as oracles, and these algorithms will be $\phi_1, \ldots, \phi_{L'}$. This completes the description of the local list correction algorithm $\mathcal{A}^f$ for $\mathcal{P}_d$, and the bound on correctness probability follows from the correctness probability of Theorem 1.3.1 and Theorem 5.0.1.

The algorithm $\mathcal{A}_1$ makes $\mathcal{O}_\varepsilon(1)$ queries to f as stated in Theorem 5.0.1, and then each ϕ_i makes $\mathcal{O}_\varepsilon(1) \cdot \tilde{\mathcal{O}}(\log n) = \tilde{\mathcal{O}}_\varepsilon(\log n)$ queries to f. ■

References

[ABP$^+$24] Prashanth Amireddy, Amik Raj Behera, Manaswi Paraashar, Srikanth Srinivasan, and Madhu Sudan. Local Correction of Linear Functions over the Boolean Cube . *Electron. Colloquium Comput. Complex.*, TR24-056, 2024. 1, 4, 6, 7, 8, 10, 11, 13, 14, 15, 24, 28, 29, 30, 39, 44, 47, 48, 54, 55, 56

[AC88] N. Alon and F.R.K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72(1):15–19, 1988. 49

[AKK$^+$05] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Trans. Inf. Theory*, 51(11):4032–4039, 2005. 14

[ALWZ21] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. *Annals of Mathematics*, 194(3):795 – 815, 2021. 39

[BCIM18] Andries E. Brouwer, Sebastian M. Cioabă, Ferdinand Ihringer, and Matt McGinnis. The smallest eigenvalues of hamming graphs, johnson graphs and other distance-regular graphs with classical parameters. *Journal of Combinatorial Theory, Series B*, 133:88–121, 2018. 52

[BDYW11] Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *ACM Symposium on Theory of Computing (STOC)*, pages 519–528, 2011. 5

[BF22] László Babai and Péter Frankl. Linear algebra methods in combinatorics 2.2. 2022. 40

[BHLR19] Abhishek Bhrushundi, Kaave Hosseini, Shachar Lovett, and Sankeerth Rao. Torus polynomials: An algebraic approach to ACC lower bounds. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 5

[BL18] Abhishek Bhowmick and Shachar Lovett. The list decoding radius for Reed-Muller codes over small fields. *IEEE Trans. Inf. Theory*, 64(6):4382–4391, 2018. 5, 10, 38

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. 14

[BSS20] Mitali Bafna, Srikanth Srinivasan, and Madhu Sudan. Local decoding and testing of polynomials over grids. *Random Struct. Algorithms*, 57(3):658–694, 2020. 6, 8, 14, 15, 25

[CLO15] David Archibald Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Cham, 2015. 38, 40

[Con] Keith Conrad. Proof of cauchy's theorem. Expository notes. 45

[Del78] Ph. Delsarte. Hahn polynomials, discrete harmonics, and t-designs. *SIAM Journal on Applied Mathematics*, 34(1):157–166, 1978. 51

[DGKS08a] Irit Dinur, Elena Grigorescu, Swastik Kopparty, and Madhu Sudan. Decodability of group homomorphisms beyond the Johnson bound. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 275–284, New York, NY, USA, 2008. Association for Computing Machinery. 7

[DGKS08b] Irit Dinur, Elena Grigorescu, Swastik Kopparty, and Madhu Sudan. Decodability of group homomorphisms beyond the Johnson bound. *Electron. Colloquium Comput. Complex.*, TR08-020, 2008. 39, 44

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. 12

[DSW14] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. *Forum Math. Sigma*, 2:Paper No. e4, 24, 2014. 5

[DSW17] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Superquadratic lower bound for 3-query locally correctable codes over the reals. *Theory Comput.*, 13:Paper No. 11, 36, 2017. 5

[ER60] Paul Erdös and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 1(1):85–90, 1960. 32

[Erd45] Paul Erdős. On a lemma of littlewood and offord. *Bulletin of the American Mathematical Society*, 51(12)(4):898–902, 1945. 10, 34, 35

[FI19] Yuval Filmus and Ferdinand Ihringer. Boolean constant degree functions on the slice are juntas. *Discrete Mathematics*, 342(12):111614, 2019. 48

[Fil16] Yuval Filmus. An orthogonal basis for functions over a slice of the boolean hypercube. *The Electronic Journal of Combinatorics*, 23:1, 2016. 48

[Fil23] Yuval Filmus. Junta threshold for low degree boolean functions on the slice. *The Electronic Journal of Combinatorics*, 30, 2023. 48

[GH00] O. Geil and T. Hoholdt. Footprints or generalized bezout's theorem. *IEEE Transactions on Information Theory*, 46(2):635–641, 2000. 10, 38, 40

[GKZ08] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. List-decoding Reed-Muller codes over small fields. In *ACM Symposium on Theory of Computing (STOC)*, pages 265–274, 2008. 5, 10, 38

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *ACM Symposium on Theory of Computing (STOC)*, pages 25–32, 1989. 5, 47

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43:439–561, 2006. 49

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 50

[Juk11] Stasys Jukna. *Extremal combinatorics: with applications in computer science*, volume 571. Springer, 2011. 30

[Kar99] Howard Karloff. How good is the goemans–williamson max cut algorithm? *SIAM Journal on Computing*, 29(1):336–350, 1999. 51

[KK17] John Y. Kim and Swastik Kopparty. Decoding Reed-Muller codes over product sets. *Theory Comput.*, 13(1):1–38, 2017. 5

[Kum52] Ernst Eduard Kummer. Über die ergänzungssätze zu den allgemeinen reciprocitätsgesetzen. 1852. 25

[KY15] Philip N. Klein and Neal E. Young. On the number of iterations for dantzig-wolfe optimization and packing-covering approximation algorithms. *SIAM J. Comput.*, 44(4):1154–1172, 2015. 52

[LO38] J.E. Littlewood and A.C. Offord. On the number of real roots of a random algebraic equation. *Journal of the London Mathematical Society*, s1-13(4):288–295, oct 1938. 29, 34, 35

[Man11] Yishay Mansour. Lecture 5: Lower Bounds using Information Theory Tools. http://www.math.tau.ac.il/~mansour/advanced-agt+ml/scribe5-lower-bound-MAB.pdf, 2011. Lecture notes. 18

[MNV16] Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. *Theory of Computing*, 12(11):1–17, 2016. 10, 29, 35, 36

[MTT61] Saburo Muroga, Iwao Toda, and Satoru Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271(5):376–418, 1961. 15

[Mul54] David E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Trans. I R E Prof. Group Electron. Comput.*, 3(3):6–12, 1954. 5

[Mur71] Saburo Muroga. *Threshold logic and its applications.* Wiley, 1971. 15

[PS97] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997. 10, 38, 39

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987. (English translation in *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987). 5

[Ree54] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Trans. IRE Prof. Group Inf. Theory*, 4:38–49, 1954. 5

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. 12

[Sch86] Alexander Schrijver. *Theory of linear and integer programming.* 1986. 22

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82. ACM, 1987. 5

[Sri11] Murali K. Srinivasan. Symmetric chains, gelfand–tsetlin chains, and the terwilliger algebra of the binary hamming scheme. *Journal of Algebraic Combinatorics*, 34, 2011. 48

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001. 5, 7, 47, 54, 55

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation*, pages 216–226. Springer Berlin Heidelberg, 1979. 12