List Decoding of Direct Sum Codes

Vedat Levi Alev[*] Fernando Granha Jeronimo[†] Dylan Quintana[‡]
Shashank Srivastava[§] Madhur Tulsiani[¶]

Abstract

We consider families of codes obtained by "lifting" a base code $\mathcal{C}$ through operations such as k-XOR applied to "local views" of codewords of $\mathcal{C}$, according to a suitable k-uniform hypergraph. The k-XOR operation yields the direct sum encoding used in works of [Ta-Shma, STOC 2017] and [Dinur and Kaufman, FOCS 2017].

We give a general framework for list decoding such lifted codes, as long as the base code admits a unique decoding algorithm, and the hypergraph used for lifting satisfies certain expansion properties. We show that these properties are indeed satisfied by the collection of length k walks on a sufficiently strong expanding graph, and by hypergraphs corresponding to high-dimensional expanders. Instantiating our framework, we obtain list decoding algorithms for direct sum liftings corresponding to the above hypergraph families. Using known connections between direct sum and direct product, we also recover (and strengthen) the recent results of Dinur et al. [SODA 2019] on list decoding for direct product liftings.

Our framework relies on relaxations given by the Sum-of-Squares (SOS) SDP hierarchy for solving various constraint satisfaction problems (CSPs). We view the problem of recovering the closest codeword to a given (possibly corrupted) word, as finding the optimal solution to an instance of a CSP. Constraints in the instance correspond to edges of the lifting hypergraph, and the solutions are restricted to lie in the base code $\mathcal{C}$. We show that recent algorithms for (approximately) solving CSPs on certain expanding hypergraphs by some of the authors also yield a decoding algorithm for such lifted codes.

We extend the framework to list decoding, by requiring the SOS solution to minimize a convex proxy for negative entropy. We show that this ensures a covering property for the SOS solution, and the "condition and round" approach used in several SOS algorithms can then be used to recover the required list of codewords.

1 Introduction

We consider the problem of list decoding binary codes obtained by starting with a binary base code $\mathcal{C}$ and amplifying its distance by "lifting" $\mathcal{C}$ to a new code $\mathcal{C}'$ using an expanding or pseudorandom structure. Examples of such constructions include *direct products* where one "lifts" (say) $\mathcal{C} \subseteq \mathbb{F}_2^n$ to $\mathcal{C}' \subseteq (\mathbb{F}_2^k)^{n^k}$ with each position in $y \in \mathcal{C}'$ being a k-tuple of bits from k positions in $z \in \mathcal{C}$. Another example is *direct sum* codes where $\mathcal{C}' \subseteq \mathbb{F}_2^{n^k}$ and each position in y is the parity of a k-tuple of bits in $z \in \mathcal{C}$. Of course, for many applications, it is interesting to consider a small “pseudorandom” set of k-tuples, instead of considering the complete set of size n^k.

This kind of distance amplification is well known in coding theory [ABN+92, IW97, GI01, TS17] and it can draw on the vast repertoire of random and pseudorandom expanding objects [HLW06, Lub18]. Such constructions are also known to have several applications to the theory of Probabilitically Checkable Proofs (PCPs) [IKW09, DS14, DDG+15, Cha16, Aro02]. However, despite having several useful properties, it might not always be clear how to *decode* the codes resulting from such constructions, especially when constructed using sparse pseudorandom structures. An important example of this phenomenon is Ta-Shma's explicit construction of binary codes of arbitrarily large distance near the (non-constructive) Gilbert-Varshamov bound [TS17]. Although the construction is explicit, efficient decoding is not known. Going beyond unique-decoding algorithms, it is also useful to have efficient list-decoding algorithms for complexity-theoretic applications [Sud00, Gur01, STV01, Tre04].

The question of list decoding such pseudorandom constructions of direct-product codes was considered by Dinur et al. [DHK+19], extending a unique-decoding result of Alon et al. [ABN+92]. While Alon et al. proved that the code is unique-decodable when the lifting hypergraph (collection of k-tuples) is a good "sampler", Dinur et al. showed that when the hypergraph has additional structure (which they called being a "double sampler") then the code is also list de-

[*] Supported by NSERC Discovery Grant 2950-120715, NSERC Accelerator Supplement 2950-120719, and partially supported by NSF awards CCF-1254044 and CCF-1718820. University of Waterloo. vlalev@uwaterloo.ca.

[†] Supported in part by NSF grants CCF-1254044 and CCF-1816372. University of Chicago. granha@uchicago.edu.

[‡] University of Chicago. dquintana@uchicago.edu

[§] TTIC. shashanks@ttic.edu

[¶] Supported by NSF grants CCF-1254044 and CCF-1816372. TTIC. madhurt@ttic.edu codable. They also posed the question of understanding structural properties of the hypergraph that might yield even unique decoding algorithms for the *direct sum* based liftings.

We develop a generic framework to understand properties of the hypergraphs under which the lifted code $\mathcal{C}'$ admits efficient list decoding algorithms, assuming only efficient unique decoding algorithms for the base code $\mathcal{C}$. Formally, let X be a downward-closed hypergraph (simplicial complex) defined by taking the downward closure of a k-uniform hypergraph, and let $g : \mathbb{F}_2^k \to \mathbb{F}_2$ be any boolean function. $X(i)$ denotes the collection of sets of size i in X and $X(\leq d)$ the collection of sets of size at most d. We consider the lift $\mathcal{C}' = \text{lift}^g_{X(k)}(\mathcal{C})$, where $\mathcal{C} \subseteq \mathbb{F}_2^{X(1)}$ and $\mathcal{C}' \subseteq \mathbb{F}_2^{X(k)}$, and each bit of $y \in \mathcal{C}'$ is obtained by applying the function g to the corresponding k bits of $z \in \mathcal{C}$. We study properties of g and X under which this lifting admits an efficient list decoding algorithm.

We consider two properties of this lifting, *robustness* and *tensoriality*, which we will be formally defined later. We will show that these properties are sufficient to yield decoding algorithms. The first property (robustness) essentially requires that for any two words in $\mathbb{F}_2^{X(1)}$ at a moderate distance, the lifting amplifies the distance between them. While the second property is of a more technical nature and is inspired by the Sum-of-Squares (SOS) SDP hierarchy used for our decoding algorithms, it is implied by some simpler combinatorial properties. Roughly speaking, this combinatorial property requires that the graph on (say) $X(k/2)$ defined by connecting $\mathfrak{s}, \mathfrak{t} \in X(k/2)$ if $\mathfrak{s} \cap \mathfrak{t} = \emptyset$ and $\mathfrak{s} \cup \mathfrak{t} \in X(k)$, is a sufficiently good expander (and similarly for graphs on $X(k/4)$, $X(k/8)$ and so on). This property requires that the k-tuples can be (recursively) split into disjoint pieces such that at each step the graph obtained between the pairs of pieces is a good expander. We refer to this property as *splittability*.

Expanding Structures. We instantiate the above framework with two specific structures: the collection of k-sized hyperedges of a high-dimensional expander (HDX) and the collection of length k walks of an expander graph. HDXs are downward-closed hypergraphs satisfying certain expansion properties. We will quantify this expansion using Dinur and Kaufman's notion of a γ-HDX [DK17].

HDXs were proved to be splittable by some of the authors [AJT19]. For the expander walk instantiation, we consider a slight variant where a walk of length k is split into two halves, which are walks of length $k/2$ (thus we do *not* consider all $k/2$ size subsets of the walk). The spectrum of the graphs obtained by this splitting can easily be related to that of the underlying expander graph. In both cases, we take the function g to be k-XOR which corresponds to the direct sum lifting. We also obtain results for direct product codes via a simple (and standard) reduction to the direct sum case.

Our Results. Now we provide a quantitative version of our main result. For this, we split the main result into two cases (due to their difference in parameters): HDXs and length k walks on expander graphs. We start with the former expanding object.

THEOREM 1.1. (HDX (INFORMAL)) *Let $\beta_0 < 1/2$ be a constant and $\beta \in (0, \beta_0)$. Suppose $X(\leq d)$ is a γ-HDX on n vertices with $\gamma \leq (\log(1/\beta))^{-O(\log(1/\beta))}$ and $d = \Omega\left((\log(1/\beta))^2/\beta^2\right)$.*

For every linear code $\mathcal{C}_1 \subset \mathbb{F}_2^n$ with relative distance $\geq 1/2 - \beta_0$, there exists a direct sum lifting $\mathcal{C}_k \subset \mathbb{F}_2^{X(k)}$ with $k = O\left(\log(1/\beta)\right)$ and relative distance $\geq 1/2 - \beta^{\Omega_{\beta_0}(1)}$ satisfying the following:

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \beta)$-close to $\mathcal{C}_k$, then we can compute the list of all the codewords of $\mathcal{C}_k$ that are $(1/2 - \beta)$-close to $\tilde{y}$ in time $n^{\beta^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.*
- *[Rate] The rate [1] r_k of $\mathcal{C}_k$ is $r_k = r_1 \cdot |X(1)| / |X(k)|$, where r_1 is the rate of $\mathcal{C}_1$.*

A consequence of this result is a method of decoding the direct product lifting on a HDX via a reduction to the direct sum case.

COROLLARY 1.1. (HDX (INFORMAL)) *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon > 0$. Suppose $X(\leq d)$ is a γ-HDX on n vertices with $\gamma \leq (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$ and $d = \Omega((\log(1/\varepsilon))^2/\varepsilon^2)$.*

For every linear code $\mathcal{C}_1 \subset \mathbb{F}_2^n$ with relative distance $\geq 1/2 - \varepsilon_0$, there exists a direct product encoding $\mathcal{C}_\ell \subset (\mathbb{F}_2^\ell)^{X(\ell)}$ with $\ell = O(\log(1/\varepsilon))$ that can be efficiently list decoded up to distance $1 - \varepsilon$.

REMARK 1.1. *List decoding the direct product lifting was first established by Dinur et al. in [DHK+19] using an expanding object introduced by them, namely, double samplers. Since constructions of double samplers are only known using HDXs, we can compare some parameters. In our setting, we obtain $d = O(\log(1/\varepsilon)^2/\varepsilon^2)$ and $\gamma = (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$ whereas in [DHK+19] $d = O(\exp(1/\varepsilon))$ and $\gamma = O(\exp(-1/\varepsilon))$.*

[1] For the rate computation, $X(k)$ is viewed as a multi-set such that each $\mathfrak{s} \in X(k)$ appears (approximately) $\propto \Pi_k(\mathfrak{s})$ times.

Given a graph G, we denote by $W_G(k)$ the collection of all length k walks of G, which plays the role of the local views $X(k)$. If G is sufficiently expanding, we have the following result.

THEOREM 1.2. (EXPANDER WALKS (INFORMAL)) *Let $\beta_0 < 1/2$ be a constant and $\beta \in (0, \beta_0)$. Suppose G is a d-regular γ-two-sided spectral expander graph on n vertices with $\gamma \le \beta^{O(1)}$.*

For every linear code $\mathcal{C}_1 \subset \mathbb{F}_2^n$ with relative distance $\ge 1/2 - \beta_0$, there exists a direct sum encoding $\mathcal{C}_k \subset \mathbb{F}_2^{\mathcal{W}_G(k)}$ with $k = O(\log(1/\beta))$ and relative distance $\ge 1/2 - \beta^{\Omega_{\beta_0}(1)}$ satisfying the following:

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \beta)$-close to $\mathcal{C}_k$, then we can compute the list of all the codewords of $\mathcal{C}_k$ that are $(1/2 - \beta)$-close to $\tilde{y}$ in time $n^{\beta^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.*
- *[Rate] The rate r_k of $\mathcal{C}_k$ is $r_k = r_1/d^k$, where r_1 is the rate of $\mathcal{C}_1$.*

The results in Theorem 1.1, Corollary 1.1, and Theorem 1.2 can all be extended (using a simple technical argument) to nonlinear base codes $\mathcal{C}_1$ with similar parameters. We also note that applying Theorem 1.1 to explicit objects derived from Ramanujan complexes [LSV05b, LSV05a] and applying Theorem 1.2 to Ramanujan graphs [LPS88] yield explicit constructions of codes with constant relative distance and rate, starting from a base code with constant relative distance and rate. With these constructions, the rate of the lifted code satisfies $r_k \ge r_1 \cdot \exp\left(-(\log(1/\beta))^{O(\log(1/\beta))}\right)$ in the HDX case and $r_k \ge r_1 \cdot \beta^{O(\log(1/\beta))}$ for expander walks. The precise parameters of these applications are given in the full version of this article.

Our techniques. We connect the question of decoding lifted codes to finding good solutions for instances of Constraint Satisfaction Problems (CSPs) which we then solve using the Sum-of-Squares (SOS) hierarchy. Consider the case of direct sum lifting, where for the lifting y of a codeword z, each bit of y is an XOR of k bits from z. If an adversary corrupts some bits of y to give $\tilde{y}$, then finding the closest codeword to $\tilde{y}$ corresponds to finding $z' \in \mathcal{C}$ such that appropriate k-bit XORs of z' agree with as many bits of $\tilde{y}$ as possible. The distance properties of the code ensure that the unique choice for z' is z (if the corruption is small). Moreover, the distance amplification (robustness) properties of the lifting can be used to show that it suffices to find *any* z' (not necessarily in $\mathcal{C}$) satisfying sufficiently many constraints. We then use results by a subset of the authors [AJT19] showing that splittability (or the tensorial nature) of the hypergraphs used for lifting can be used to yield algorithms for approximately solving the related CSPs. Of course, the above argument does not rely on the lifting being direct sum and works for any lifting function g.

For list decoding, we solve just a single SOS program whose solution is rich enough to “cover” the list of code words we intend to retrieve. In particular, the solutions to the CSP are obtained by “conditioning" the SDP solution on a small number of variables, and we try to ensure that in the list decoding case, conditioning the SOS solution on different variables yields solutions close to different elements of the list. To achieve this covering property we consider a convex proxy Ψ for negative entropy measuring how concentrated (on a few code words) the SOS solution is. Then we minimize Ψ while solving the SOS program. A similar technique was also independently used by Karmalkar, Klivans, and Kothari [KKK19] and Raghavendra and Yau [RY19] in the context of learning regression. Unfortunately, this SOS cover comes with only some weak guarantees which are, a priori, not sufficient for list decoding. However, again using the robustness property of the lifting, we are able to convert weak covering guarantees for the lifted code $\mathcal{C}'$ to strong guarantees for the base code $\mathcal{C}$, and then appeal to the unique decoding algorithm. We regard the interplay between these two properties leading to the final list decoding application as our main technical contribution. A more thorough overview is given in Section 3 after introducing some objects and notation in Section 2. In Section 3, we also give further details about the organization of the document.

Related work. The closest result to ours is the list decoding framework of Dinur et al. [DHK$^+$19] for the direct product encoding, where the lifted code is not binary but rather over the alphabet $\mathbb{F}_2^k$. Our framework instantiated for the direct sum encoding on HDXs (cf. Theorem 1.1) captures and strengthens some of their parameters in Corollary 1.1. While Dinur et al. also obtain list decoding by solving an SDP for a specific CSP (Unique Games), the reduction to CSPs in their case uses the combinatorial nature of the instances (double sampler) and is also specific to the direct product encoding. They recover the list by iteratively solving many CSP instances, reducing the alphabet size by one each time. On the other hand, the reduction to CSPs is somewhat generic in our framework and the recovery of the list is facilitated by including an entropic proxy in the convex relation. As mentioned earlier, a similar entropic proxy was also (independently) used by Karmalkar et al. [KKK19] and Raghavendra and Yau [RY19] in the context of list decoding for linear regression and mean estimation. Direct products on expanders were also used as a building block by Guruswami and Indyk [GI03] who used these to construct *linear time* list decodable codes over large alphabets. They gave an algorithm for recovering the list based on spectral partitioning techniques.

2 Preliminaries

2.1 Simplicial Complexes It will be convenient to work with hypergraphs satisfying a certain downward-closed property (which is straightforward to obtain).

Definition 2.1. *A* simplicial complex *X with ground set $[n]$ is a downward-closed collection of subsets of $[n]$, i.e., for all sets $\mathfrak{s} \in X$ and $\mathfrak{t} \subseteq \mathfrak{s}$, we also have $\mathfrak{t} \in X$. The sets in X are referred to as* faces *of X. We use the notation $X(i)$ for the set of all faces of a simplicial complex X with cardinality i and $X(\leq d)$ for the set of all faces of cardinality at most d.* [2] *By convention, we take $X(0) := \{\varnothing\}$.*

A simplicial complex $X(\leq d)$ is said to be a pure simplicial complex *if every face of X is contained in some face of size d. Note that in a pure simplicial complex $X(\leq d)$, the top slice $X(d)$ completely determines the complex.*

Simplicial complexes are equipped with the following probability measures on their sets of faces.

Definition 2.2. (Measures $(\Pi_1, \ldots, \Pi_d)$) *Let $X(\leq d)$ be a pure simplicial complex and let Π_d be an arbitrary probability measure on $X(d)$. We define a coupled array of random variables $(\mathfrak{s}^{(d)}, \ldots, \mathfrak{s}^{(1)})$ as follows: sample $\mathfrak{s}^{(d)} \sim \Pi_d$ and (recursively) for each $i \in [d]$, take $\mathfrak{s}^{(i-1)}$ to be a uniformly random subset of $\mathfrak{s}^{(i)}$ of size $i-1$. The distributions $\Pi_{d-1}, \ldots, \Pi_1$ are then defined to be the marginal distributions of the random variables $\mathfrak{s}^{(d-1)}, \ldots, \mathfrak{s}^{(1)}$. We also define the joint distribution of $(\mathfrak{s}^{(d)}, \ldots, \mathfrak{s}^{(1)})$ as Π. Note that the choice of Π_d determines each other distribution Π_i on $X(i)$.*

In order to work with the HDX and expander walk instantiations in a unified manner, we will use also use the notation $X(k)$ to indicate the set of all length k walks on a graph G. In this case, $X(k)$ is a set of k-tuples rather than subsets of size k. This distinction will be largely irrelevant, but we will use $W_G(k)$ when referring specifically to walks rather than subsets. The set of walks $W_G(k)$ has a corresponding distribution Π_k as well.

2.2 Codes and Lifts

Codes We briefly recall some standard terminology from coding theory. Let Σ be a finite alphabet with $q \in \mathbb{N}$ symbols. We will be mostly concerned with the case $\Sigma = \mathbb{F}_2$. Given $z, z' \in \Sigma^n$, recall that the relative Hamming distance between z and z' is $\Delta(z, z') := \left|\{i \mid z_i \neq z'_i\}\right| / n$. Any set $\mathcal{C} \subset \Sigma^n$ gives rise to a q-ary code. The distance of $\mathcal{C}$ is defined as $\Delta(\mathcal{C}) := \min_{z \neq z'} \Delta(z, z')$ where $z, z' \in \mathcal{C}$. We say that $\mathcal{C}$ is a linear code [3] if $\Sigma = \mathbb{F}_q$ and $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$. The rate of $\mathcal{C}$ is $\log_q(|\mathcal{C}|)/n$.

Instead of discussing the distance of a binary code, it will often be more natural to phrase results in terms of its bias.

Definition 2.3. (Bias) *The* bias *of a word* [4] *$z \in \mathbb{F}_2^n$ is $\operatorname{bias}(z) := \left|\mathbb{E}_{i \in [n]} (-1)^{z_i}\right|$. The bias of a code $\mathcal{C}$ is the maximum bias of any codeword in $\mathcal{C}$.*

Lifts Starting from a code $\mathcal{C}_1 \subset \Sigma_1^{X(1)}$, we amplify its distance by considering a *lifting* operation defined as follows.

Definition 2.4. (Lifting Function) *Let $g : \Sigma_1^k \to \Sigma_k$ and $X(k)$ be a collection of k-uniform hyperedges or walks of length k on the set $X(1)$. For $z \in \Sigma_1^{X(1)}$, we define $\operatorname{lift}_{X(k)}^g(z) = y$ such that $y_{\mathfrak{s}} = g(z|_{\mathfrak{s}})$ for all $\mathfrak{s} \in X(k)$, where $z|_{\mathfrak{s}}$ is the restriction of z to the indices in $\mathfrak{s}$.*

The lifting of a code $\mathcal{C}_1 \subseteq \Sigma_1^{X(1)}$ is

$$\operatorname{lift}_{X(k)}^g(\mathcal{C}_1) = \{\operatorname{lift}_{X(k)}^g(z) \mid z \in \mathcal{C}_1\},$$

which we will also denote $\mathcal{C}_k$. We will omit g and $X(k)$ from the notation for lifts when they are clear from context.

We will call liftings that amplify the distance of a code *robust*.

Definition 2.5. (Robust Lifting) *We say that $\operatorname{lift}_{X(k)}^g$ is (δ_0, δ)-robust if for every $z, z' \in \Sigma_1^{X(1)}$ we have*

$$\Delta(z, z') \geq \delta_0 \Rightarrow \Delta(\operatorname{lift}(z), \operatorname{lift}(z')) \geq \delta.$$

For us the most important example of lifting is when the function g is k-XOR and $\Sigma_1 = \Sigma_k = \mathbb{F}_2$, which has been extensively studied in connection with codes and otherwise [TS17, STV01, GNW95, ABN$^+$92]. In our language of liftings, k-XOR corresponds to the *direct sum lifting*.

[2] Note that it is more common to associate a geometric representation to simplicial complexes, with faces of cardinality i being referred to as faces of *dimension* $i-1$ (and the collection being denoted by $X(i-1)$ instead of $X(i)$). However, we prefer to index faces by their cardinality to improve readability of related expressions.

[3] In this case, q is required to be a prime power.

[4] Equivalently, the bias of $z \in \{\pm 1\}^n$ is $\operatorname{bias}(z) := \left|\mathbb{E}_{i \in [n]} z_i\right|$.

Definition 2.6. (Direct Sum Lifting) *Let $\mathcal{C}_1 \subseteq \mathbb{F}_2^n$ be a base code on $X(1) = [n]$. The* direct sum lifting *of a word $z \in \mathbb{F}_2^n$ on a collection $X(k)$ is $\mathrm{dsum}_{X(k)}(z) = y$ such that $y_{\mathfrak{s}} = \sum_{i \in \mathfrak{s}} z_i$ for all $\mathfrak{s} \in X(k)$.*

We will be interested in cases where the direct sum lifting reduces the bias of the base code; in [TS17], structures with such a property are called parity samplers, as they emulate the reduction in bias that occurs by taking the parity of random samples.

Definition 2.7. (Parity Sampler) *Let $g \colon \mathbb{F}_2^k \to \mathbb{F}_2$. We say that $\mathrm{lift}_{X(k)}^g$ is an $(\varepsilon_0, \varepsilon)$-parity sampler if for all $z \in \mathbb{F}_2^{X(1)}$ with* $\mathrm{bias}(z) \le \varepsilon_0$, *we have* $\mathrm{bias}(\mathrm{lift}(z)) \le \varepsilon$.

2.3 Constraint Satisfaction Problems (CSPs)

A k-CSP instance $\mathfrak{I}(H, \mathcal{P}, w)$ with alphabet size q consists of a k-uniform hypergraph H, a set of constraints

$$\mathcal{P} = \{\mathcal{P}_{\mathfrak{a}} \subseteq [q]^{\mathfrak{a}} : \mathfrak{a} \in H\},$$

and a non-negative weight function $w \in \mathbb{R}_+^H$ on the constraints satisfying $\sum_{\mathfrak{a} \in H} w(a) = 1$.

We will think of the constraints as predicates that are satisfied by an assignment σ if we have $\sigma|_{\mathfrak{a}} \in \mathcal{P}_{\mathfrak{a}}$, i.e., the restriction of σ on $\mathfrak{a}$ is contained in $\mathcal{P}_{\mathfrak{a}}$. We write $\mathsf{SAT}_{\mathfrak{I}}(\sigma)$ for the (weighted) fraction of the constraints satisfied by the assignment σ, i.e.,

$$\begin{aligned}\mathsf{SAT}_{\mathfrak{I}}(\sigma) &= \sum_{\mathfrak{a} \in H} w(\mathfrak{a}) \cdot \mathbf{1}[\sigma|_{\mathfrak{a}} \in \mathcal{P}_{\mathfrak{a}}] \\ &= \mathop{\mathbb{E}}_{\mathfrak{a} \sim w} \left[\mathbf{1}[\sigma|_{\mathfrak{a}} \in \mathcal{P}_{\mathfrak{a}}]\right] .\end{aligned}$$

We denote by $\mathsf{OPT}(\mathfrak{I})$ the maximum of $\mathsf{SAT}_{\mathfrak{I}}(\sigma)$ over all $\sigma \in [q]^{V(H)}$.

A particularly important class of k-CSPs for our work will be k-XOR: here the input consists of a k-uniform hypergraph H with weighting w, and a (right hand side) vector $r \in \mathbb{F}_2^H$. The constraint for each $\mathfrak{a} \in X(k)$ requires

$$\sum_{i \in \mathfrak{a}} \sigma(i) = r_{\mathfrak{a}} \pmod 2.$$

In this case we will use the notation $\mathfrak{I}(H, r, w)$ to refer to the k-XOR instance. When the weighting w is implicitly clear, we will just omit it and write $\mathfrak{I}(H, r)$.

Any k-uniform hypergraph H can be associated with a pure simplicial complex in a canonical way by just setting $X_{\mathfrak{I}} = \{\mathfrak{b} : \exists\, \mathfrak{a} \in H \text{ and } \mathfrak{a} \supseteq \mathfrak{b}\}$; notice that $X_{\mathfrak{I}}(k) = H$. We will refer to this complex as the constraint complex of the instance $\mathfrak{I}$. The probability distribution Π_k on $X_{\mathfrak{I}}$ will be derived from the weight function w of the constraint:

$$\Pi_k(\mathfrak{a}) = w(\mathfrak{a}) \quad \forall \mathfrak{a} \in X_{\mathfrak{I}}(k) = H.$$

2.4 Sum-of-Squares Relaxations and t-local PSD Ensembles

The Sum-of-Squares (SOS) hierarchy gives a sequence of increasingly tight semidefinite programming relaxations for several optimization problems, including CSPs. Since we will use relatively few facts about the SOS hierarchy, already developed in the analysis of Barak, Raghavendra and Steurer [BRS11], we will adapt their notation of t-local distributions to describe the relaxations. For a k-CSP instance $\mathfrak{I} = (H, \mathcal{P}, w)$ on n variables, we consider the following semidefinite relaxation given by t-levels of the SOS hierarchy, with vectors $v_{(S,\alpha)}$ for all $S \subseteq [n]$ with $|S| \le t$, and all $\alpha \in [q]^S$. Here, for $\alpha_1 \in [q]^{S_1}$ and $\alpha_2 \in [q]^{S_2}$, $\alpha_1 \circ \alpha_2 \in [q]^{S_1 \cup S_2}$ denotes the partial assignment obtained by concatenating α_1 and α_2.

$$\boxed{\begin{array}{llcll}
\text{maximize} & \displaystyle\mathop{\mathbb{E}}_{\mathfrak{a} \sim w}\left[\sum_{\alpha \in \mathcal{P}_{\mathfrak{a}}} \|v_{(\mathfrak{a},\alpha)}\|^2\right] & =: & \mathsf{SDP}(\mathfrak{I}) & \\
\text{subject to} & \left\langle v_{(S_1,\alpha_1)}, v_{(S_2,\alpha_2)} \right\rangle & = & 0 & \forall\, \alpha_1|_{S_1 \cap S_2} \ne \alpha_2|_{S_1 \cap S_2} \\
 & \left\langle v_{(S_1,\alpha_1)}, v_{(S_2,\alpha_2)} \right\rangle & = & \left\langle v_{(S_3,\alpha_3)}, v_{(S_4,\alpha_4)} \right\rangle & \forall\, S_1 \cup S_2 = S_3 \cup S_4, \\
 & & & & \alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4 \\
 & \displaystyle\sum_{j \in [q]} \|v_{(\{i\},j)}\|^2 & = & 1 & \forall i \in [n] \\
 & \|v_{(\emptyset,\emptyset)}\| & = & 1 &
\end{array}}$$

For any set S with $|S| \le t$, the vectors $v_{(S,\alpha)}$ induce a probability distribution μ_S over $[q]^S$ such that the assignment $\alpha \in [q]^S$ appears with probability $\|v_{(S,\alpha)}\|^2$. Moreover, these distributions are consistent on intersections i.e., for $T \subseteq S \subseteq [n]$, we have $\mu_{S|T} = \mu_T$, where $\mu_{S|T}$ denotes the restriction of the distribution μ_S to the set T. We use these distributions to define a collection of random variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ taking values in $[q]$, such that for any set S with $|S| \le t$, the collection of variables $\{\mathbf{Z}_i\}_{i \in S}$ have a joint distribution μ_S. Note that the entire collection $(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)$ *may not* have a joint distribution: this property is only true for sub-collections of size t. We will refer to the collection $(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)$ as a t-local ensemble of random variables.

We also have that that for any $T \subseteq [n]$ with $|T| \le t - 2$, and any $\xi \in [q]^T$, we can define a $(t - |T|)$-local ensemble $(\mathbf{Z}_1', \ldots, \mathbf{Z}_n')$ by "conditioning" the local distributions on the event $\mathbf{Z}_T = \xi$, where $\mathbf{Z}_T$ is shorthand for the collection $\{\mathbf{Z}_i\}_{i \in T}$. For any S with $|S| \le t - |T|$, we define the distribution of $\mathbf{Z}_S'$ as $\mu_S' := \mu_{S \cup T} | \{\mathbf{Z}_T = \xi\}$. Finally, the semidefinite program also ensures that for any such conditioning, the conditional covariance matrix

$$\mathsf{M}_{(S_1,\alpha_1)(S_2,\alpha_2)} = \mathrm{Cov}\left(\mathbf{1}[\mathbf{Z}_{S_1}' = \alpha_1], \mathbf{1}[\mathbf{Z}_{S_2}' = \alpha_2]\right)$$

is positive semidefinite, where $|S_1|, |S_2| \le (t - |T|)/2$. Here, for each pair S_1, S_2 the covariance is computed using the joint distribution $\mu'_{S_1 \cup S_2}$. In this paper, we will only consider t-local ensembles such that for every conditioning on a set of size at most $t-2$, the conditional covariance matrix is PSD. We will refer to these as t-local PSD ensembles. We will also need a simple corollary of the above definitions.

FACT 2.1. *Let $(\mathbf{Z}_1, \dots, \mathbf{Z}_n)$ be a t-local PSD ensemble, and let X be any collection with $X(1) = [n]$. Then, for all $s \le t/2$, the collection $\{\mathbf{Z}_{\mathfrak{a}}\}_{\mathfrak{a} \in X(\le s)}$ is a (t/s)-local PSD ensemble, where $X(\le s) = \bigcup_{i=1}^{s} X(i)$.*

For random variables $\mathbf{Z}_S$ in a t-local PSD ensemble, we use the notation $\{\mathbf{Z}_S\}$ to denote the distribution of $\mathbf{Z}_S$ (which exists when $|S| \le t$). We also define $\operatorname{Var}[\mathbf{Z}_S]$ as $\sum_{\alpha \in [q]^S} \operatorname{Var}[\mathbf{1}[\mathbf{Z}_S = \alpha]]$.

Pseudo-expectation Formulation An equivalent way of expressing this local PSD ensemble is through the use of a pseudo-expectation operator which is also a language commonly used in the SOS literature (e.g., [BHK+16, BKS17]). The exposition of some of our results is cleaner in this equivalent language. Each variable $\mathbf{Z}_i$ with $i \in [n]$ is modeled by a collection of indicator local random variables [5] $\{\mathbf{Z}_{i,a}\}_{a \in [q]}$ with the intent that $\mathbf{Z}_{i,a} = 1$ iff $\mathbf{Z}_i = a$. To ensure they behave similarly to indicators we add the following restrictions to the SOS formulation:

$$\begin{aligned} \mathbf{Z}_{i,a}^2 &= \mathbf{Z}_{i,a} && \forall i \in [n], a \in [q] \\ \sum_{a \in [q]} \mathbf{Z}_{i,a} &= 1 && \forall i \in [n] \end{aligned}$$

Let $\mathcal{R} = \mathbb{R}[\mathbf{Z}_{1,1}, \dots, \mathbf{Z}_{n,q}]$ be the ring of polynomials on $\{\mathbf{Z}_{i,a}\}_{i \in [n], a \in [q]}$. We will write $\mathcal{R}^{\le d}$ for the restriction of $\mathcal{R}$ to polynomials of degree at most d. A feasible solution at the $2t$-th level of the SOS hierarchy is a linear operator $\widetilde{\mathbb{E}} : \mathcal{R}^{\le 2t} \to \mathbb{R}$ called the pseudo-expectation operator. This operator satisfies the following problem independent constraints: (i) $\widetilde{\mathbb{E}}[1] = 1$ (normalization) and (ii) $\widetilde{\mathbb{E}}[P^2] \ge 0$ for every $P \in \mathcal{R}^{\le t}$ (non-negative on Sum-of-Squares) [6]. It also satisfies the problem dependent constraints

$$\widetilde{\mathbb{E}}\left[\mathbf{Z}_{i,a}^2 \cdot P\right] = \widetilde{\mathbb{E}}\left[\mathbf{Z}_{i,a} \cdot P\right] \text{ and}$$
$$\widetilde{\mathbb{E}}\left[\left(\sum_{a \in [q]} \mathbf{Z}_{i,a}\right) \cdot Q\right] = \widetilde{\mathbb{E}}\left[Q\right],$$

for every $i \in [n]$, $a \in [q]$, $P \in \mathcal{R}^{\le 2t-2}$ and $Q \in \mathcal{R}^{\le 2t-1}$. Note that for any collection of local random variables $\mathbf{Z}_{i_1}, \dots, \mathbf{Z}_{i_j}$ with $j \le 2t$ we have the joint distribution

$$\mathbb{P}(\mathbf{Z}_{i_1} = a_1, \dots, \mathbf{Z}_{i_j} = a_j) = \widetilde{\mathbb{E}}\left[\mathbf{Z}_{i_1,a_1} \dots \mathbf{Z}_{i_j,a_j}\right].$$

Even though we may not have a global distribution we can implement a form of pseudo-expectation conditioning on a random variable $\mathbf{Z}_i$ taking a given value $a \in [q]$ as long as $\mathbb{P}[\mathbf{Z}_i = a] = \widetilde{\mathbb{E}}[\mathbf{Z}_{i,a}] > 0$. This can be done by considering the new operator $\widetilde{\mathbb{E}}_{|Z_i = a}\colon \mathcal{R}^{\le 2t-2} \to \mathbb{R}$ defined as $\widetilde{\mathbb{E}}_{|\mathbf{Z}_i = a}[\cdot] = \widetilde{\mathbb{E}}[\mathbf{Z}_{i,a}^2 \cdot]/\widetilde{\mathbb{E}}[\mathbf{Z}_{i,a}^2]$, which is a valid pseudo-expectation operator at the $(2t-2)$-th level. This conditioning can be naturally generalized to a set of variables $S \subseteq [n]$ with $|S| \le t$ satisfying $\mathbf{Z}_S = \alpha$ for some $\alpha \in [q]^S$.

Notation We make some systematic choices for our parameters in order to syntactically stress their qualitative behavior.

- $1/2 - \beta_0$ is a lower bound on the distance of the base code $\mathcal{C}_1$.
- $1/2 - \beta$ is a lower bound on the distance of the lifted code $\mathcal{C}_k$.
- μ, θ, η are parameters that can be made arbitrarily small by increasing the SOS degree and/or the quality of expansion.
- ε, δ can be arbitrary error parameters.
- $\lambda_1 \ge \lambda_2 \ge \cdots$ are the eigenvalues of a graph's adjacency matrix (in $[-1, 1]$).
- $\sigma_1 \ge \sigma_2 \ge \cdots$ are the singular values of a graph's adjacency matrix (in $[0, 1]$).

We also make some choices for words and local variables to distinguish the ground space $\mathbb{F}_2^{X(1)}$ form the lifted space $\mathbb{F}_2^{X(k)}$.

- $z, z', z'', \dots$ are words in the ground space $\mathbb{F}_2^{X(1)}$.
- $y, y', y'', \dots$ are words in the lifted space $\mathbb{F}_2^{X(k)}$.
- $\mathbf{Z} := \{\mathbf{Z}_1, \dots, \mathbf{Z}_n\}$ is a local PSD ensemble on the ground set $X(1)$.
- $\mathbf{Y} := \{\mathbf{Y}_{\mathfrak{s}} := (\operatorname{lift}(\mathbf{Z}))_{\mathfrak{s}} \mid \mathfrak{s} \in X(k)\}$ is a local ensemble on $X(k)$.

3 Proof Strategy and Organization

As discussed earlier, we view the problem of finding the closest codeword(s) as that of finding suitable solution(s) to an instance of a CSP (which is k-XOR in the case of direct sum). We now discuss some of the technical ingredients required in the decoding procedure.

[5] Note that $\{\mathbf{Z}_{i,a}\}_{i \in [n], a \in [q]}$ are formal variables in the SOS formulation.

[6] From condition (ii), we can recover the PSD properties from the local PSD ensemble definition.

Unique Decoding. Given $\mathcal{C}_k = \operatorname{dsum}_{X(k)}(\mathcal{C}_1)$ with the lifting function as k-XOR, we can view the problem of finding the closest codeword to a given $\tilde{y}$ as that of finding the unique $z \in \mathcal{C}_1$ satisfying the maximum number of equations of the form $\sum_{i \in \mathfrak{s}} z_i = \tilde{y}_{\mathfrak{s}} \pmod 2$, with one equation for each $\mathfrak{s} \in X(k)$. Using the results of [AJT19], it is indeed possible to find $z' \in \mathbb{F}_2^n$ such that $\Delta(\operatorname{dsum}(z'), \tilde{y}) \leq \Delta(\operatorname{dsum}(z), \tilde{y}) + \varepsilon$. We then argue that z' or its complement $\overline{z'}$ must be close to $z \in \mathcal{C}_1$ which can then be recovered by unique decoding. Here, z is such that $y = \operatorname{dsum}(z)$ is the unique codeword closest to $\tilde{y}$.

If this is not the case, then $z - z'$ must have bias bounded away from 1, which would imply by robustness (parity sampling property of the hypergraph) that $\operatorname{dsum}(z - z')$ has bias close to zero i.e., $\Delta(\operatorname{dsum}(z), \operatorname{dsum}(z')) \approx 1/2$. However, if $\Delta(\tilde{y}, \mathcal{C}_k) \leq \eta$, then we must have

$$\begin{aligned} \Delta(\operatorname{dsum}(z), \operatorname{dsum}(z')) \;\leq\; & \Delta(\operatorname{dsum}(z), \tilde{y}) \\ & + \Delta(\operatorname{dsum}(z'), \tilde{y}) \\ \leq\; & 2\eta + \varepsilon, \end{aligned}$$

which leads to a contradiction if η is significantly below $1/4$ and ε is sufficiently small.

List Decoding. We start by describing an abstract list decoding framework which only assumes two general properties of a lifting $\operatorname{lift}^{g}_{X(k)}$: (i) it is distance amplifying (*robust*) and (ii) it is amenable to SOS rounding (*tensorial*).

Suppose $\tilde{y} \in \mathbb{F}_2^{X(k)}$ is a word promised to be $(1/2 - \sqrt{\beta})$-close to a lifted code $\mathcal{C}_k = \operatorname{lift}(\mathcal{C}_1)$ where $\mathcal{C}_k$ has distance at least $1/2 - \beta$ and $\mathcal{C}_1$ has distance at least $1/2 - \beta_0$. By list decoding $\tilde{y}$, we mean finding a list $\mathcal{L} \subseteq \mathcal{C}_k$ of all codewords $(1/2 - \sqrt{\beta})$-close to $\tilde{y}$.

Our framework for list decoding $\tilde{y}$ consists of three stages. In the first stage, we set up and solve a natural SOS program which we treat abstractly in this discussion. One issue with using a rounding algorithm for this relaxation to do list decoding is that this natural SOS program may return a solution that is "concentrated", e.g., a SOS solution corresponding to single codeword in $\mathcal{L}$. Such a solution will of course not have enough information to recover the entire list. To address this issue we now ask not only for feasibility in our SOS program but also to minimize a convex function Ψ measuring how concentrated the SOS solution is. Specifically, if $\mathbf{Z}$ is the PSD ensemble corresponding to the solution of the SOS program and if $\mathbf{Y}$ is the lifted ensemble, then we minimize $\Psi := \mathbb{E}_{\mathfrak{s},\mathfrak{t} \in X(k)}\left[\widetilde{\mathbb{E}}(\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}})^2\right]$.

The key property of the function Ψ is that if the SOS solution "misses" any element in the list $\mathcal{L}$ then it is possible to decrease it. Since our solution is a minimizer [7] of Ψ, this is impossible. Therefore, our solution does "cover" the list $\mathcal{L}$. Even with this SOS cover of $\mathcal{L}$, the list decoding task is not complete. So far we have not talked about rounding, which is necessary to extract codewords out of the (fractional) solution. For now, we will simply assume that rounding is viable (this is handled by the second stage of the framework) and resume the discussion.

Unfortunately, the covering guarantee is somewhat weak, namely, for $y \in \mathcal{L}$ we are only able to obtain a word $y' \in \mathbb{F}_2^{X(k)}$ with weak agreement $|\langle y', y\rangle| \geq 2 \cdot \beta$. Converting a word y' from the cover into an actual codeword y is the goal of the third and final stage of the list decoding framework, dubbed *Cover Purification*. At this point we resort to the robustness properties of the lifting and the fact that we actually have "coupled" pairs $(z, y = \operatorname{lift}(z))$ and $(z', y' = \operatorname{lift}(z'))$ for some $z, z' \in \mathbb{F}_2^{X(1)}$. Due to this robustness (and up to some minor technicalities) even a weak agreement between y and y' in the lifted space translates into a much stronger agreement between z and z' in the ground space. Provided the latter agreement is sufficiently strong, z' will lie in the unique decoding ball centered at z in $\mathcal{C}_1$. In this case, we can uniquely recover z and thus also $y = \operatorname{lift}(z)$. Furthermore, if $\mathcal{C}_1$ admits an efficient unique decoder, we can show that this step in list decoding $\tilde{y}$ can be done efficiently.

Now we go back to fill in the rounding step, which constitutes the second stage of the framework, called *Cover Retrieval*. We view the SOS solution as composed of several "slices" from which the weak pairs (z', y') are to be extracted. Note that the framework handles, in particular, k-XOR liftings where it provides not just a single solution but a list of them. Hence, some structural assumption about $X(k)$ is necessary to ensure SOS tractability. Recall that random k-XOR instances are hard for SOS [Gri01, KMOW17]. For this reason, we impose a sufficient tractability condition on $X(k)$ which we denote the *two-step tensorial* property. This notion is a slight strengthening of a *tensorial* property which was (implicitly) first investigated by Barak et al. [BRS11] when $k = 2$ and later generalized for arbitrary $k \geq 2$ in [AJT19]. Roughly speaking, if $X(k)$ is tensorial then the SOS local random variables in a typical slice of the solution behave approximately as product variables from the perspective of the local views $\mathfrak{s} \in X(k)$. A two-step tensorial structure is a tensorial structure in which the local random variables between pairs of local views $\mathfrak{s}, \mathfrak{t} \in X(k)$ are also close to product

[7] Actually an approximate minimizer is enough in our application.

variables, which is an extra property required to perform rounding in this framework. With the two-step tensorial assumption, we are able to round the SOS solution to obtain a list of pairs (z', y') weakly agreeing with elements of the code list that will be refined during cover purification.

Finding suitable hypergraphs. Fortunately, objects satisfying the necessary tensorial and robustness assumptions do exist. HDXs were shown to be tensorial in [AJT19], and here we strengthen this result to two-step tensorial as well as prove that HDXs possess the particular robustness property of parity sampling. Walks on expander graphs are already known to be robust [TS17], and we use a modified version of the methods in [AJT19] to show they are also two-step tensorial. For both HDXs and expander walks, we describe how to use known constructions of these objects to get explicit direct sum encodings that can be decoded using our abstract framework.

Reduction from direct product to direct sum. Finally, we describe how to use list decoding results for direct sum codes to obtain results for direct product codes. Given a direct product lifting $\mathcal{C}_k$ with the hypergraph $X(k)$, if $\Delta(\tilde{y}, y) \leq 1 - \varepsilon$ for $y \in \mathcal{C}_k$, then we must have that

$$\Pr_{\mathfrak{s} \in X(k)} [y_{\mathfrak{s}} = \tilde{y}_{\mathfrak{s}}] \;=\; \mathop{\mathbb{E}}_{\mathfrak{s} \in X(k)} \left[\mathop{\mathbb{E}}_{\mathfrak{t} \subseteq \mathfrak{s}} [\chi_{\mathfrak{t}}(y_{\mathfrak{s}} + \tilde{y}_{\mathfrak{s}})] \right] \;\geq\; \varepsilon \,.$$

Since $\chi_{\mathfrak{t}}(y_{\mathfrak{s}})$ can be viewed as a direct sum lifting, we get by grouping subsets $\mathfrak{t}$ by size that there must exist a size i such that the direct sum lifting at $X(i)$ has correlation at least ε with the word y' defined as $y'_{\mathfrak{t}} = \chi_{\mathfrak{t}}(\tilde{y}_{\mathfrak{s}})$ for all $\mathfrak{t} \in X(i)$. We can then apply the list decoding algorithm for direct sum codes on $X(i)$. A standard concentration argument can also be used to control the size i to be approximately $k/2$.

Organization of Results In Section 4, we show how the direct sum lifting on HDXs can be used to reduce bias, establishing that HDXs are parity samplers. This will give a very concrete running example of a lifting that can be used in our framework. We remark in Section 5 how this lifting can be used in the simpler regime of unique decoding using a k-CSP algorithm on expanding instances [AJT19]. The abstract list decoding framework together with its concrete instantiation to HDXs and expander walks are given in the full version of this article.

4 Pseudorandom Hypergraphs and Robustness of Direct Sum

The main robustness property we will consider is parity sampling applied to the case of the direct sum lifting. As this section focuses on this specific instance of a lifting, here we will say that a collection $X(k)$ is a parity sampler if its associated direct sum lifting $\mathrm{dsum}_{X(k)}$ is a parity sampler. Recall that for such a parity sampler, the direct sum lifting brings the bias of a code close to zero, which means it boosts the distance almost to $1/2$.

4.1 Expander Walks and Parity Sampling

A known example of a parity sampler is the set $X(k)$ of all walks of length k in a sufficiently expanding graph, as shown by Ta-Shma.

THEOREM 4.1. (PARITY SAMPLERS [TS17]) *Suppose G is a graph with second largest singular value at most λ, and let $X(k)$ be the set of all walks of length k on G. Then $X(k)$ is an $(\varepsilon_0, (\varepsilon_0 + 2\lambda)^{\lfloor k/2 \rfloor})$-parity sampler.*

Our goal in this section is to prove a similar result for high-dimensional expanders, where $X(k)$ is the set of k-sized faces. We begin with an overview of some useful properties of HDXs.

4.2 High-dimensional Expanders

A high-dimensional expander (HDX) is a particular kind of simplicial complex satisfying an expansion requirement. We recall the notion of high-dimensional expansion considered in [DK17]. For a complex $X(\leq d)$ and $\mathfrak{s} \in X(i)$ for some $i \in [d]$, we denote by $X_{\mathfrak{s}}$ the link complex

$$X_{\mathfrak{s}} \;:=\; \{\mathfrak{t} \backslash \mathfrak{s} \mid \mathfrak{s} \subseteq \mathfrak{t} \in X\}\,.$$

When $|\mathfrak{s}| \;\leq\; d-2$, we also associate a natural weighted graph $G(X_{\mathfrak{s}})$ to a link $X_{\mathfrak{s}}$, with vertex set $X_{\mathfrak{s}}(1)$ and edge set $X_{\mathfrak{s}}(2)$. The edge weights are taken to be proportional to the measure Π_2 on the complex $X_{\mathfrak{s}}$, which is in turn proportional to the measure $\Pi_{|\mathfrak{s}|+2}$ on X. The graph $G(X_{\mathfrak{s}})$ is referred to as the skeleton of $X_{\mathfrak{s}}$.

Dinur and Kaufman [DK17] define high-dimensional expansion in terms of spectral expansion of the skeletons of the links.

DEFINITION 4.1. (γ-HDX FROM [DK17]) *A simplicial complex $X(\leq d)$ is said to be γ-*High Dimensional Expander *(γ-*HDX*) if for every $0 \leq i \leq d-2$ and for every $\mathfrak{s} \in X(i)$, the graph $G(X_{\mathfrak{s}})$ satisfies $\sigma_2(G(X_{\mathfrak{s}})) \leq \gamma$.*

We will need the following theorem relating γ to the spectral properties of the graph between two layers of an HDX.

THEOREM 4.2. (ADAPTED FROM [DK17]) *Let X be a γ-HDX and let $M_{1,d}$ be the weighted bipartite containment graph between $X(1)$ and $X(d)$, where each edge $(\{i\}, \mathfrak{s})$ has* *weight $(1/d)\Pi_d(\mathfrak{s})$. Then the second largest singular value σ_2 of $M_{1,d}$ satisfies*

$$\sigma_2^2 \leq \frac{1}{d} + O(d\gamma).$$

We will be defining codes using HDXs by associating each face in some $X(i)$ with a position in the code. The distance between two codewords does not take into account any weights on their entries, which will be problematic when decoding since the distributions Π_i are not necessarily uniform. To deal with this issue, we will work with HDXs where the distributions Π_i satisfy a property only slightly weaker than uniformity.

DEFINITION 4.2. (FLATNESS (FROM [DHK+19])) *We say that a distribution Π on a finite probability space Ω is D-flat if there exits N such that each singleton $\omega \in \Omega$ has probability in $\{1/N, \ldots, D/N\}$.*

Using the algebraically deep construction of Ramanujan complexes by Lubotzky, Samuels and Vishne [LSV05b, LSV05a], Dinur and Kaufman [DK17] showed that sparse γ-HDX do exist, with flat distributions on their sets of faces. The following lemma from [DHK+19] is a refinement of [DK17].

LEMMA 4.1. (EXTRACTED FROM [DHK+19]) *For every $\gamma > 0$ and every $d \in \mathbb{N}$ there exists an explicit infinite family of bounded degree d-sized complexes which are γ-HDXs. Furthermore, there exists a $D \leq (1/\gamma)^{O(d^2/\gamma^2)}$ such that*

$$\frac{|X(d)|}{|X(1)|} \leq D,$$

the distribution Π_1 is uniform, and the other distributions $\Pi_d, \ldots, \Pi_2$ are D-flat.

For a D-flat distribution Π_i, we can duplicate each face in $X(i)$ at most D times to make Π_i the same as a uniform distribution on this multiset. We will always perform such a duplication implicitly when defining codes on $X(i)$.

4.3 HDXs are Parity Samplers

To prove that sufficiently expanding HDXs are parity samplers, we establish some properties of the complete complex and then explore the fact that HDXs are locally complete [8]. We first show that the expectation over k-sized edges of a complete complex X on t vertices approximately splits into a product of k expectations over $X(1)$ provided $t \gg k^2$.

CLAIM 4.1. (NEAR INDEPENDENCE) *Suppose X is the complete complex of dimension at least k with Π_k uniform over $X(k)$ and Π_1 uniform over $X(1) = [t]$. For a function $f : X(1) \to \mathbb{R}$, let*

$$\mu_k = \mathbb{E}_{\mathfrak{s}\sim\Pi_k}\left[\prod_{i\in\mathfrak{s}} f(i)\right] \qquad \text{and} \qquad \mu_1 = \mathbb{E}_{i\sim\Pi_1}\left[f(i)\right].$$

Then

$$\left|\mu_k - \mu_1^k\right| \leq \frac{k^2}{t}\left\|f\right\|_\infty^k.$$

Proof. Let

$$\begin{aligned}
\mathcal{E} &= \{(i_1, \ldots, i_k) \in X(1)^k \mid i_1, \ldots, i_k \text{ are distinct}\},\\
\delta &= \mathop{\mathbb{P}}_{i_1,\ldots,i_k\sim\Pi_1}[(i_1, \ldots, i_k) \notin \mathcal{E}], \text{ and}\\
\eta &= \mathbb{E}_{(i_1,\ldots,i_k)\in X(1)^k\setminus\mathcal{E}}[f(i_1)\cdots f(i_k)].
\end{aligned}$$

Then

$$\begin{aligned}
\mu_1^k &= \mathbb{E}_{i_1,\ldots,i_k\sim\Pi_1}[f(i_1)\cdots f(i_k)]\\
&= (1-\delta)\cdot\mathbb{E}_{(i_1,\ldots,i_k)\in\mathcal{E}}[f(i_1)\cdots f(i_k)]\\
&\quad + \delta\cdot\mathbb{E}_{(i_1,\ldots,i_k)\in X(1)^k\setminus\mathcal{E}}[f(i_1)\cdots f(i_k)]\\
&= (1-\delta)\cdot\mu_k + \delta\cdot\eta,
\end{aligned}$$

where the last equality follows since Π_k is uniform and the product in the expectation is symmetric. As $i_1, \ldots, i_k$ are sampled independently from Π_1, which is uniform over $X(1)$,

$$\delta = 1 - \prod_{j<k}\left(1 - \frac{j}{t}\right) \leq \sum_{j<k}\frac{j}{t} = \frac{k(k-1)}{2t},$$

so we have

$$\left|\mu_k - \mu_1^k\right| = \delta\left|\mu_k - \eta\right| \leq \frac{k^2}{2t}\left(2\left\|f\right\|_\infty^k\right).$$

□

We will derive parity sampling for HDXs from their behavior as samplers. A sampler is a structure in which the average of any function on a typical local view is close to its overall average. More precisely, we have the following definition.

DEFINITION 4.3. (SAMPLER) *Let $G = (U, V, E)$ be a bipartite graph with a probability distribution Π_E on the edges E. Let Π_U and Π_V be the marginal distributions of Π_E on U and V, respectively. We say that G is an (η, δ)-sampler if for every function $f\colon V \to [0,1]$ with $\mu = \mathbb{E}_{v\sim\Pi_V} f(v)$,*

$$\mathop{\mathbb{P}}_{u\sim\Pi_U}\left[\left|\mathbb{E}_{v\sim u} f(v) - \mu\right| \geq \eta\right] \leq \delta.$$

[8] This a recurring theme in the study of HDXs [DK17].

To relate parity sampling to spectral expansion, we use the following fact establishing that samplers of arbitrarily good parameters (η, δ) can be obtained from sufficiently expanding bipartite graphs. This result is essentially a corollary of the expander mixing lemma.

FACT 4.1. (FROM DINUR ET AL. [DHK+19]) *A weighted bipartite graph with second singular value σ_2 is an $(\eta, \sigma_2^2/\eta^2)$ sampler.*

Using Claim 4.1, we show that the graph between $X(1)$ and $X(k)$ obtained from a HDX is a parity sampler, with parameters determined by its sampling properties.

CLAIM 4.2. (SAMPLER BIAS AMPLIFICATION) *Let $X(\leq d)$ be a HDX such that the weighted bipartite graph $M_{1,d}$ between $X(1) = [n]$ and $X(d)$ is an (η, δ)-sampler. For any $1 \leq k \leq d$, if $z \in \mathbb{F}_2^n$ has bias at most ε_0, then*

$$\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) \leq (\varepsilon_0 + \eta)^k + \frac{k^2}{d} + \delta.$$

Proof. By downward closure, the subcomplex $X|_{\mathfrak{t}}$ obtained by restricting to edges contained within some $\mathfrak{t} \in X(d)$ is a complete complex on the ground set $\mathfrak{t}$. Since $M_{1,d}$ is an (η, δ)-sampler, the bias of $z|_{\mathfrak{t}}$ must be within η of $\mathrm{bias}(z)$ on all but δ fraction of the edges $\mathfrak{t}$. Hence

$$\begin{aligned}
&\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) = \left|\mathbb{E}_{\{i_1,\dots,i_k\}\sim\Pi_k}(-1)^{z_{i_1}+\dots+z_{i_k}}\right| \\
&= \left|\mathbb{E}_{\mathfrak{t}\sim\Pi_d}\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_{\mathfrak{t}}(k)}(-1)^{z_{i_1}+\dots+z_{i_k}}\right| \\
&\leq \left|\mathbb{E}_{\mathfrak{t}\sim\Pi_d}\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_{\mathfrak{t}}(k)}(-1)^{z_{i_1}+\dots+z_{i_k}}\mathbb{1}_{[\mathrm{bias}(z|_{\mathfrak{t}})\leq\varepsilon_0+\eta]}\right| \\
&\qquad + \mathop{\mathbb{P}}_{t\sim\Pi_d}[\mathrm{bias}(z|_{\mathfrak{t}}) > \varepsilon_0 + \eta] \\
&\leq \mathbb{E}_{\mathfrak{t}\sim\Pi_d}\mathbb{1}_{[\mathrm{bias}(z|_{\mathfrak{t}})\leq\varepsilon_0+\eta]}\left|\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_{\mathfrak{t}}(k)}(-1)^{z_{i_1}+\dots+z_{i_k}}\right| + \delta.
\end{aligned}$$

By Claim 4.1, the magnitude of the expectation of $(-1)^{z_i}$ over the edges of size k in the complete complex $X|_{\mathfrak{t}}$ is close to $\left|\mathbb{E}_{i\sim X|_{\mathfrak{t}}(1)}(-1)^{z_i}\right|$, which is just the bias of $z|_{\mathfrak{t}}$. Then

$$\begin{aligned}
\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) &\leq \mathbb{E}_{\mathfrak{t}\sim X(d)}\mathbb{1}_{[\mathrm{bias}(z|_{\mathfrak{t}})\leq\varepsilon_0+\eta]}\,\mathrm{bias}(z|_{\mathfrak{t}})^k \\
&\quad + \frac{k^2}{d} + \delta \\
&\leq (\varepsilon_0 + \eta)^k + \frac{k^2}{d} + \delta,
\end{aligned}$$

concluding the proof. □

Now we can compute the parameters necessary for a HDX to be an $(\varepsilon_0, \varepsilon)$-parity sampler for arbitrarily small ε.

LEMMA 4.2. (HDXS ARE PARITY SAMPLERS) *Let $0 < \varepsilon \leq \varepsilon_0 < 1$, $0 < \theta < (1/\varepsilon_0) - 1$, and $k \geq \log_{(1+\theta)\varepsilon_0}(\varepsilon/3)$. If $X(\leq d)$ is a γ-HDX with $d \geq \max\{3k^2/\varepsilon, 6/(\theta^2\varepsilon_0^2\varepsilon)\}$ and $\gamma = O\left(1/d^2\right)$, then $X(k)$ is an $(\varepsilon_0, \varepsilon)$-parity sampler.*

Proof. Suppose the graph $M_{1,d}$ between $X(1)$ and $X(d)$ is an (η, δ)-sampler. We will choose d and γ so that $\eta = \theta\varepsilon_0$ and $\delta = \varepsilon/3$. Using Fact 4.1 to obtain a sampler with these parameters, we need the second singular value σ_2 of $M_{1,d}$ to be bounded as

$$\sigma_2 \leq \theta\varepsilon_0\sqrt{\frac{\varepsilon}{3}}.$$

By the upper bound on σ_2^2 from Theorem 4.2, it suffices to have

$$\frac{1}{d} + O\left(d\gamma\right) \leq \frac{\theta^2\varepsilon_0^2\varepsilon}{3},$$

which is satisfied by taking $d \geq 6/\left(\theta^2\varepsilon_0^2\varepsilon\right)$ and $\gamma = O\left(1/d^2\right)$.

By Claim 4.2, $X(k)$ is an $(\varepsilon_0, (\varepsilon_0 + \eta)^k + k^2/d + \delta)$-parity sampler. The first term in the bias is $(\varepsilon_0 + \eta)^k = ((1+\theta)\varepsilon_0)^k$, so we require $(1+\theta)\varepsilon_0 < 1$ to amplify the bias by making k large. To make this term smaller than $\varepsilon/3$, k must be at least $\log_{(1+\theta)\varepsilon_0}(\varepsilon/3)$. We already chose $\delta = \varepsilon/3$, so ensuring $d \geq 3k^2/\varepsilon$ gives us an $(\varepsilon_0, \varepsilon)$-parity sampler. □

4.4 Rate of the Direct Sum Lifting

By applying the direct sum lifting on a HDX to a base code $\mathcal{C}_1$ with bias ε_0, parity sampling allows us to obtain a code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ with arbitrarily small bias ε at the cost of increasing the length of the codewords. The following lemma gives a lower bound on the rate of the lifted code $\mathcal{C}_k$.

LEMMA 4.3. (RATE OF DIRECT SUM LIFTING FOR HDX) *Let $\varepsilon_0 \in (0,1)$ and $\theta \in (0, (1/\varepsilon_0) - 1)$ be constants, and let $\mathcal{C}_1$ be an ε_0-biased binary linear code with relative rate r_1. For $\varepsilon \in (0, \varepsilon_0]$, suppose k, d, and γ satisfy the hypotheses of Lemma 4.2, with k and d taking the smallest values that satisfy the lemma. The relative rate r_k of the code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ with bias ε constructed on a HDX with these parameters satisfies*

$$r_k \geq r_1 \cdot \gamma^{O((\log(1/\varepsilon))^4/(\varepsilon^2\gamma^2))}.$$

If $\gamma = C/d^2$ for some constant C, then this becomes

$$r_k \geq r_1 \cdot \left(\frac{\varepsilon^2}{(\log(1/\varepsilon))^4}\right)^{O((\log(1/\varepsilon))^{12}/\varepsilon^6)}.$$

Proof. Performing the lifting from $\mathcal{C}_1$ to $\mathcal{C}_k$ does not change the dimension of the code, but it does increase the length of the codewords from n to $|X(k)|$, where $|X(k)|$ is the size of the multiset of edges of size k after each edge has been copied a number of times proportional to its weight. Using the bound and flatness guarantee from Lemma 4.1, we can compute

$$r_k = \frac{r_1 n}{|X(k)|} \geq \frac{r_1}{D^2},$$

where $D \leq (1/\gamma)^{O(d^2/\gamma^2)}$. Treating ε_0 and θ as constants, the values of k and d necessary to satisfy Lemma 4.2 are

$$k = \log_{(1+\theta)\varepsilon_0}(\varepsilon/3) = O(\log(1/\varepsilon))$$

and

$$d = \max\left\{\frac{3k^2}{\varepsilon}, \frac{6}{\theta^2\varepsilon_0^2\varepsilon}\right\} = O\left(\frac{(\log(1/\varepsilon))^2}{\varepsilon}\right).$$

Putting this expression for d into the inequality for D yields

$$D \leq (1/\gamma)^{O((\log(1/\varepsilon))^4/(\varepsilon^2\gamma^2))},$$

from which the bounds in the lemma statement follow. □

From Lemma 4.3, we see that if $\mathcal{C}_1$ has constant rate, then $\mathcal{C}_k$ has a rate constant with respect to n. However, the dependence of the rate on the bias ε is quite poor. This is especially striking in comparison to the rate achievable using Ta-Shma's expander walk construction described in Section 4.1.

LEMMA 4.4. (RATE FOR WALKS [TS17]) *Let $\varepsilon_0 \in (0,1)$ be a constant and $\mathcal{C}_1$ be an ε_0-biased binary linear code with relative rate r_1. Fix $\varepsilon \in (0, \varepsilon_0]$. Suppose G is a graph with second largest singular value $\lambda = \varepsilon_0/2$ and degree $d \leq 4/\lambda^2$. Let $k = 2\log_{2\varepsilon_0}(\varepsilon) + 1$ and $X(k)$ be the set of all walks of length k on G. Then the direct sum lifting $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ has bias ε and rate $r_k \geq r_1 \cdot \varepsilon^{O(1)}$.*

Proof. From Theorem 4.1 with this choice of λ and k, the direct sum lifting $\mathcal{C}_k$ has bias ε. For the rate, observe that the lifting increases the length of the codewords from n to the number of walks of length k on G, which is nd^k. Thus the rate of $\mathcal{C}_k$ is

$$r_k = \frac{r_1 n}{nd^k} = \frac{r_1}{d^k}$$

As $d \leq 16/\varepsilon_0$, which is a constant, and $k = O(\log(1/\varepsilon))$, the rate satisfies $r_k \geq r_1 \cdot \varepsilon^{O(1)}$. □

5 Unique Decoding

In this section, we will show how the parity sampling of $\mathrm{dsum}_{X(k)}$ and the ability to solve k-XOR instances with $X(k)$ as their constraint complex allow us to decode $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$, where $\mathcal{C}_1 \in \mathbb{F}_2^n$ is a linear code. With a more technical argument, we can also handle non-linear codes and different kinds of lifting, but for clarity of exposition we restrict our attention to the preceding setting.

5.1 Unique Decoding on Parity Samplers

Our approach to unique decoding for $\mathcal{C}_k$ is as follows. Suppose a received word $\tilde{y} \in \mathbb{F}_2^{X(k)}$ is close to $y^\star \in \mathcal{C}_k$, which is the direct sum lifting of some $z^\star \in \mathcal{C}_1$ on $X(k)$. We first find an approximate solution $z \in \mathbb{F}_2^n$ to the k-XOR instance $\mathfrak{I}(X(k), \tilde{y})$ with predicates

$$\sum_{i \in \mathfrak{s}} z_i = \tilde{y}_{\mathfrak{s}} \pmod 2$$

for every $\mathfrak{s} \in X(k)$. Note that z being an approximate solution to $\mathfrak{I}(X(k), \tilde{y})$ is equivalent to its lifting $\mathrm{dsum}_{X(k)}(z)$ being close to $\tilde{y}$. In Lemma 5.1, we show that if $\mathrm{dsum}_{X(k)}$ is a sufficiently strong parity sampler, either z or its complement $\overline{z}$ will be close to $z^\star$. Running the unique decoding algorithm for $\mathcal{C}_1$ on z and $\overline{z}$ will recover $z^\star$, from which we can obtain $y^\star$ by applying the direct sum lifting.

LEMMA 5.1. *Let $0 < \beta < 1/2$ and $0 < \varepsilon < 1/4 - \beta/2$. Suppose $\mathcal{C}_1$ is a linear code that is efficiently uniquely decodable within radius $1/4 - \mu_0$ for some $\mu_0 > 0$, and $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ where $\mathrm{dsum}_{X(k)}$ is a $(1/2 + 2\mu_0, 2\beta)$-parity sampler. Let $\tilde{y} \in \mathbb{F}_2^{X(k)}$ be a word that has distance strictly less than $(1/4 - \beta/2 - \varepsilon)$ from $\mathcal{C}_k$, and let $y^\star = \mathrm{dsum}_{X(k)}(z^\star) \in \mathcal{C}_k$ be the word closest to $\tilde{y}$.*

Then, for any $z \in \mathbb{F}_2^n$ satisfying

$$\Delta(\mathrm{dsum}_{X(k)}(z), \tilde{y}) < \frac{1}{4} - \frac{\beta}{2},$$

we have either

$$\Delta(z^\star, z) \leq \frac{1}{4} - \mu_0 \quad \text{or} \quad \Delta(z^\star, \overline{z}) \leq \frac{1}{4} - \mu_0.$$

In particular, either z or $\overline{z}$ can be efficiently decoded in $\mathcal{C}_1$ to obtain $z^\star \in \mathcal{C}_1$.

REMARK 5.1. *Since $\mathrm{dsum}_{X(k)}$ is a $(1/2 + 2\mu_0, 2\beta)$-parity sampler, the code $\mathcal{C}_k$ has distance $\Delta(\mathcal{C}_k) \geq 1/2 - \beta$. This implies that $z^\star \in \mathcal{C}_1$ is unique, since its direct sum lifting $y^\star$ is within distance $\Delta(\mathcal{C}_k)/2$ of $\tilde{y}$.*

Proof. Let $y = \mathrm{dsum}_{X(k)}(z)$. We have

$$\Delta(y^\star, y) \leq \Delta(y^\star, \tilde{y}) + \Delta(y, \tilde{y}) < \frac{1}{2} - \beta.$$

By linearity of $\mathrm{dsum}_{X(k)}$, $\Delta(\mathrm{dsum}_{X_k}(z^\star - z), 0) < 1/2 - \beta$, so $\mathrm{bias}(\mathrm{dsum}_k(z^\star - z)) > 2\beta$. From the $(1/2 + 2\mu_0, 2\beta)$-parity sampling assumption, $\mathrm{bias}(z^\star - z) > 1/2 + 2\mu_0$. Translating back to distance, either $\Delta(z^\star, z) < 1/4 - \mu_0$ or $\Delta(z^\star, z) > 3/4 + \mu_0$, the latter being equivalent to $\Delta(z^\star, \overline{z}) < 1/4 - \mu_0$. □

To complete the unique decoding algorithm, we need only describe how a good enough approximate solution $z \in \mathbb{F}_2^n$ to a k-XOR instance $\mathfrak{I}(X(k), \tilde{y})$ allows us to recover $z^\star \in \mathcal{C}_1$ provided $\tilde{y}$ is sufficiently close to $\mathcal{C}_k$. We will rely on the following observation relating satisfiability and distance.

REMARK 5.2.

$$\mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z) = 1 - \Delta(\mathrm{dsum}_{X(k)}(z), \tilde{y}).$$

COROLLARY 5.1. *Suppose $\mathcal{C}_1$, $X(k)$, $z^\star$, $y^\star$ and $\tilde{y}$ are as in the assumptions of Lemma 5.1. If $z \in \mathbb{F}_2^n$ is such that*

$$\mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z) \geq \mathsf{OPT}_{\mathfrak{I}(X(k),\tilde{y})} - \varepsilon,$$

then unique decoding either z or $\overline{z}$ gives $z^\star \in \mathcal{C}_1$. Furthermore, if such a z can be found efficiently, so can $z^\star$.

Proof. By the assumption on z, we have

$$\begin{aligned} 1 - \Delta(\mathrm{dsum}_{X(k)}(z), \tilde{y}) &= \mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z) \\ &\geq \mathsf{OPT}_{\mathfrak{I}(X(k),\tilde{y})} - \varepsilon \\ &\geq \mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z^\star) - \varepsilon \\ &= 1 - \Delta(y^\star, \tilde{y}) - \varepsilon, \end{aligned}$$

implying $\Delta(\mathrm{dsum}_{X(k)}(z), \tilde{y}) \leq \Delta(y^\star, \tilde{y}) + \varepsilon$. Using the assumption that $\tilde{y}$ has distance strictly less than $(1/4 - \beta/2 - \varepsilon)$ from $\mathcal{C}_k$, we get that $\Delta(\mathrm{dsum}_{X(k)}(z), \tilde{y}) < 1/4 - \beta/2$ in which case we are under all the conditions required in Lemma 5.1. □

5.2 Concrete Instantiations

High Dimensional Expanders If the collection $X(k)$ is part of a sufficiently expanding γ-HDX, we can use the following algorithm to approximately solve the k-XOR instance.

THEOREM 5.1. ([AJT19]) *Let $\mathfrak{I}$ be an instance of MAX k-CSP on n variables taking values over an alphabet of size q, and let $\varepsilon > 0$. Let the simplicial complex $X_{\mathfrak{I}}$ be a γ-HDX with $\gamma = \varepsilon^{O(1)} \cdot (1/(kq))^{O(k)}$. Then there is an algorithm based on $(k/\varepsilon)^{O(1)} \cdot q^{O(k)}$ levels of the Sum-of-Squares hierarchy which produces an assignment satisfying at least an $(\mathsf{OPT} - \varepsilon)$ fraction of the constraints.*

If X is a HDX with the parameters to both satisfy this theorem and be a $(1/2 + 2\mu_0, 2\beta)$ parity sampler, we can achieve efficient unique decodability of $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$.

COROLLARY 5.2. *Let $X(\leq d)$ be a d-dimensional γ-HDX satisfying the premises of Lemma 4.2 that would guarantee that $X(k)$ is a $(1/2 + 2\mu_0, 2\beta)$-parity sampler, namely, for some $0 < \theta < \frac{2}{1+4\mu_0} - 1$, we have $k \geq \log_{(1+\theta)\cdot(\frac{1}{2}+2\mu_0)}(2\beta/3)$, $d \geq \max\left\{\frac{3k^2}{2\beta}, \frac{3}{\theta^2(1/2+2\mu_0)^2\beta}\right\}$, and $\gamma = O(1/d^2)$.*

Then, assuming $\mathcal{C}_1$ is efficiently unique decodable within radius $1/4 - \mu_0$, one can uniquely decode $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ within distance $1/4 - \beta/2 - \varepsilon$ in time $n^{(k/\varepsilon)^{O(1)} \cdot 2^{O(k)}}$,[9] where we have

$$\varepsilon = (\gamma \cdot (2k)^{O(k)})^{\frac{1}{O(1)}}.$$

Expander Walks In the full version of this article, we show that the algorithmic results of [AJT19] can be made to work when $X(k)$ is the set of walks of length k of a suitably strong expander. In particular, we have the following.

THEOREM 5.2. *Let $G = (V, E)$ be a graph with $\sigma_2(G) = \lambda$, and k a given parameter. Let $\mathfrak{I}$ be a k-CSP instance over an alphabet of size k whose constraint graph is the set of walks in G of length k. Let $\varepsilon > 0$ be such that $\lambda = O(\varepsilon^2/(k^2 \cdot q^{2k}))$.*

Then there exists an algorithm based on $O\left(\frac{q^{4k}k^7}{\varepsilon^5}\right)$-levels of the Sum-of-Squares hierarchy which produces an assignment satisfying at least an $(\mathsf{OPT} - \varepsilon)$-fraction of the constraints.

Using this result, one can efficiently unique decode $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ when $X(k)$ is the set of walks of length k on an expander strong enough to achieve the necessary parity sampling property.

COROLLARY 5.3. *Let $\mathcal{C}_1$ be a code that is efficiently unique decodable within radius $1/4 - \mu_0$ for some $\mu_0 > 0$. Let G be a graph satisfying $\sigma_2(G) = \lambda$, with $1/2 + \mu_0 + 2\lambda < 1$. Let $k \geq 2\log_{1/2+\mu_0+2\lambda}(2\beta) + 1$, and define $X(k)$ to be the set of all walks in G of length at most k—note that $\mathrm{dsum}_{X(k)}$ is a $(1/2 + 2\mu_0, 2\beta)$-parity sampler by Theorem 4.1.*

The code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ can be efficiently decoded within radius $1/4 - \beta/2 - \varepsilon$ in time $n^{O(2^{4k} \cdot k^7/\varepsilon^5)}$, where we have

$$\varepsilon = O(\lambda \cdot k^2 \cdot 2^k).$$

[9] Here we are assuming that uniquely decoding $\mathcal{C}_1$ within radius $1/4 - \mu_0$ takes time less than this.

REMARK 5.3. *In both Corollary 5.2 and Corollary 5.3, when μ_0 and β are constants, k can be constant, which means we can decode $\mathcal{C}_k$ from a radius arbitrarily close to $1/4 - \beta/2$ if we have strong enough guarantees on the quality of the expansion of the high-dimensional expander or the graph, respectively.*

Notice, however, that the unique decodability radius of the code $\mathcal{C}_k$ is potentially larger than $1/4 - \beta/2$. Our choice of $(1/2 + 2\mu_0, 2\beta)$-parity sampling is needed to ensure that the approximate k-CSP solutions lie within the unique decoding radius of $\mathcal{C}_1$. Since the bias of the code $\mathcal{C}_1$ will generally be smaller than the parity sampling requirement of $1/2 + 2\mu_0$, the bias of the code $\mathcal{C}_k$ will be smaller than 2β. In this case, the maximum distance at which our unique decoding algorithm works will be smaller than $\Delta(\mathcal{C}_k)/2$.

References

[ABN+92] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth. Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 28:509–516, 1992. 1, 4

[AJT19] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. Manuscript, 2019. 2, 3, 7, 8, 12

[Aro02] Sanjeev Arora. How NP got a new definition: a survey of probabilistically checkable proofs. In *Proceedings of the International Congress of Mathematicians*, pages 637–648, 2002. Volume 3. 1

[BHK+16] B. Barak, S. B. Hopkins, J. Kelner, P. Kothari, A. Moitra, and A. Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem, 2016. 6

[BKS17] Boaz Barak, Pravesh K. Kothari, and David Steurer. Quantum entanglement, sum of squares, and the log rank conjecture. In *Proceedings of the 49th ACM Symposium on Theory of Computing*, pages 975–988. ACM, 2017. 6

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science*, pages 472–481, 2011. 5, 7

[Cha16] Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3), August 2016. 1

[DDG+15] Roee David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. ITCS '15, pages 327–336, New York, NY, USA, 2015. ACM. 1

[DHK+19] Irit Dinur, Prahladh Harsha, Tali Kaufman, Inbal Livni Navon, and Amnon Ta-Shma. List decoding with double samplers. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms*, pages 2134–2153, 2019. 1, 2, 3, 9, 10

[DK17] Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 974–985, 2017. 2, 8, 9

[DS14] Irit Dinur and David Steurer. Direct product testing. In *Proceedings of the 29th IEEE Conference on Computational Complexity*, CCC '14, pages 188–196, 2014. 1

[GI01] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 658–667, 2001. 1

[GI03] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, 2003. 4

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95-50, Electronic Colloquium on Computational Complexity, 1995. 4

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001. 7

[Gur01] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. PhD thesis, MIT, 2001. 1

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, August 2006. 1

[IKW09] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, STOC '09, pages 131–140, 2009. 1

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ unless E has sub-exponential circuits. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997. 1

[KKK19] Sushrut Karmalkar, Adam R. Klivans, and Pravesh K. Kothari. List-decodable linear regression. *CoRR*, abs/1905.05679, 2019. URL: `http://arxiv.org/abs/1905.05679`, `arXiv:1905.05679`. 3

[KMOW17] Pravesh Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th ACM Symposium on Theory of Computing*, 2017. 7

[LPS88] Alexander Lubotzky, R. Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988. 3

[LSV05a] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Explicit constructions of ramanujan complexes of type ad. *Eur. J. Comb.*, 26(6):965–993, August 2005. 3, 9

[LSV05b] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Ramanujan complexes of typeãd. *Israel Journal of Mathematics*, 149(1):267–299, Dec 2005. 3, 9

[Lub18] Alexander Lubotzky. High dimensional expanders. In *ICM*, 2018. 1

[RY19] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. *CoRR*, abs/1905.04660, 2019. URL: `http://arxiv.org/abs/1905.04660`, `arXiv:1905.04660`. 3

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan.

Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. 1, 4

[Sud00] Madhu Sudan. List decoding: Algorithms and applications. In *Proceedings of the International Conference IFIP on Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics*, TCS '00, pages 25–41, Berlin, Heidelberg, 2000. Springer-Verlag. 1

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004. arXiv:cs.CC/0409044. 1

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th ACM Symposium on Theory of Computing*, STOC 2017, pages 238–251, New York, NY, USA, 2017. ACM. 1, 4, 5, 8, 11