List Decoding of Direct Sum Codes

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

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.

*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

Contents

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 decodable. 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 $\mathcal{C}' = \mathrm{lift}^g_{X(k)}(\mathcal{C})$, 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*, formally defined later, which 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, which we refer to as *splittability*, 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). Splittability 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.

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 [1] on 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 variant of splittability 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 (Direct Sum Lifting on HDX (Informal))**.** *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. Suppose $X(\leq d)$ is a γ-HDX on n vertices with $\gamma \leq (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$ and $d = \Omega\left((\log(1/\varepsilon))^2/\varepsilon^2\right)$.*

For every linear code $\mathcal{C}_1 \subset \mathbb{F}_2^n$ with relative distance $\geq 1/2 - \varepsilon_0$, there exists a direct sum lifting $\mathcal{C}_k \subset \mathbb{F}_2^{X(k)}$ with $k = O(\log(1/\varepsilon))$ and relative distance $\geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ satisfying the following:

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \varepsilon)$-close to $\mathcal{C}_k$, then we can compute the list of all the codewords of $\mathcal{C}_k$ that are $(1/2 - \varepsilon)$-close to $\tilde{y}$ in time $n^{\varepsilon^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.*
- *[Rate] The rate [2] 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 1.2 (Direct Product Lifting on 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 $(1 - \varepsilon)$.

[1] Actually, we will be working with length $k-1$ walks which can be represented as k-tuples, though this is an unimportant technicality. The reason is to be consistent in the number of vertices (allowing repetitions) with k-sized hyperedges.

[2] In the rate computation, $X(k)$ is viewed as a multi-set where each $\mathfrak{s} \in X(k)$ is repeated a certain number of times for technical reasons.

Remark 1.3. *List decoding the direct product lifting was first established by Dinur et al. in [DHK+19] using their notion of 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* $\gamma = O(\exp(-1/\varepsilon))$.

Given a graph G, we denote by $W_G(k)$ the collection of all length $k-1$ walks of G, which plays the role of the local views $X(k)$. If G is sufficiently expanding, we have the following result.

Theorem 1.4 (Direct Sum Lifting on Expander Walks (Informal))**.** *Let* $\varepsilon_0 < 1/2$ *be a constant and* $\varepsilon \in (0, \varepsilon_0)$. *Suppose* G *is a* d*-regular* γ*-two-sided spectral expander graph on* n *vertices with* $\gamma \leq \varepsilon^{O(1)}$.

For every linear code $\mathcal{C}_1 \subset \mathbb{F}_2^n$ *with relative distance* $\geq 1/2 - \varepsilon_0$, *there exists a direct sum encoding* $\mathcal{C}_k \subset \mathbb{F}_2^{W_G(k)}$ *with* $k = O(\log(1/\varepsilon))$ *and relative distance* $\geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ *satisfying the following:*

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

The results in Theorem 1.1, Corollary 1.2, and Theorem 1.4 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 1.4 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 $r_k \geq r_1 \cdot \exp\left(-(\log(1/\varepsilon))^{O(\log(1/\varepsilon))}\right)$ in the HDX case and $r_k \geq r_1 \cdot \varepsilon^{O(\log(1/\varepsilon))}$ for expander walks. The precise parameters of these applications are given in Corollary 7.2 of Section 7 and in Corollary 9.5 of Section 9, respectively.

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. If the corruption is small, the distance properties of the code ensure that the unique choice for z' is z. 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 codewords 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 codewords) 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–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 (c.f. Theorem 1.1) captures and strengthens some of their parameters in Corollary 1.2. 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 double sampler instances and is also specific to the direct product encoding. They recover the list by iteratively solving many CSP instances, where each newly found solution is pruned from the instance by 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–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*.* [3] *By convention, we take* $X(0) := \{\varnothing\}$*.*

[3] 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.

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 (Probability 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 $k-1$ 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 (see Definition 9.1).

2.2 Codes and Lifts

Codes

We briefly recall some standard code terminology. 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 $\Delta(z, z') := |\{i \mid z_i \neq z'_i\}| / n$. 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 [4] 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* [5] *$z \in \mathbb{F}_2^n$ is* $\text{bias}(z) := \left|\mathbb{E}_{i \in [n]} (-1)^{z_i}\right|$*. The bias of a code $\mathcal{C}$ is the maximum bias of any non-zero 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 $k-1$ on the set $X(1)$. For $z \in \Sigma_1^{X(1)}$, we define $\text{lift}^g_{X(k)}(z) = y$ 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 $\mathfrak{s}$.*

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

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

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

$$\mathrm{lift}^g_{X(k)}(\mathcal{C}_1) = \{\mathrm{lift}^g_{X(k)}(z) \mid z \in \mathcal{C}_1\},$$

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 $\mathrm{lift}^g_{X(k)}$ is (δ_0, δ)-robust if for every $z, z' \in \Sigma_1^{X(1)}$ we have*

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

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*.

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 $\mathrm{lift}^g_{X(k)}$ is an (β_0, β)-parity sampler if for all $z \in \mathbb{F}_2^{X(1)}$ with $\mathrm{bias}(z) \leq \beta_0$, we have $\mathrm{bias}(\mathrm{lift}(z)) \leq \beta$.*

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.,

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

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 $\mathfrak{a} \in H$ 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 omit it and just write $\mathfrak{I}(H, r)$.

Any k-uniform hypergraph H can be associated with a pure simplicial complex in a canonical way by setting $X_{\mathfrak{I}} = \{\mathfrak{b} : \exists\, \mathfrak{a} \in H \text{ with } \mathfrak{a} \supseteq \mathfrak{b}\}$; 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 $X_{\mathfrak{I}}(k)$ 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 $|S| \leq t$, 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.

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

For any set S with $|S| \leq t$, 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: 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 $|S| \leq t$, the collection of variables $\{\mathbf{Z}_i\}_{i \in S}$ has 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 $|T| \leq t-2$, 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 $|S| \leq t - |T|$, 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 $|S_1|, |S_2| \leq (t - |T|)/2$. 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 2.8. *Let $(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)$ be a t-local PSD ensemble, and let X be any collection with $X(1) = [n]$. Then, for all $s \leq t/2$, the collection $\{\mathbf{Z}_\mathfrak{a}\}_{\mathfrak{a} \in X(\leq s)}$ is a (t/s)-local PSD ensemble, where $X(\leq s) = \bigcup_{i=1}^{s} X(i)$.*

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 $|S| \leq t$). We also define $\mathrm{Var}[\mathbf{Z}_S]$ as

$$\mathrm{Var}\left[\mathbf{Z}_S\right] := \sum_{\alpha \in [q]^S} \mathrm{Var}\left[\mathbf{1}\left[\mathbf{Z}_S = \alpha\right]\right].$$

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 [6] $\{\mathbf{Z}_{i,a}\}_{a \in [q]}$ 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 $\mathcal{R} = \mathbb{R}[\mathbf{Z}_{1,1}, \ldots, \mathbf{Z}_{n,q}]$ be the ring of polynomials on $\{\mathbf{Z}_{i,a}\}_{i \in [n], a \in [q]}$. We will write $\mathcal{R}^{\leq d}$ 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 $\widetilde{\mathbb{E}} : \mathcal{R}^{\leq 2t} \to \mathbb{R}$ called the pseudo-expectation operator. This operator satisfies the following problem-independent constraints: (i) $\widetilde{\mathbb{E}}[1] = 1$ (normalization) and (ii) $\widetilde{\mathbb{E}}[P^2] \geq 0$ for every $P \in \mathcal{R}^{\leq t}$ (non-negative on Sum-of-Squares) [7]. It also satisfies the problem-dependent constraints

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

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

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

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

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

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

$\mathbb{P}[\mathbf{Z}_i = a] = \widetilde{\mathbb{E}}[\mathbf{Z}_{i,a}] > 0$. This can be done by considering the new operator $\widetilde{\mathbb{E}}_{|Z_i=a} \colon \mathcal{R}^{\leq 2t-2} \to \mathbb{R}$ 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 $|S| \leq t$ 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.

- $1/2 - \varepsilon_0$ is a lower bound on the distance of the base code $\mathcal{C}_1$.
- $1/2 - \varepsilon$ is a lower bound on the distance of the lifted code $\mathcal{C}_k$.
- κ is a parameter that will control the list-decodability 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.
- β, δ are arbitrary error parameters.
- $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues of a graph's adjacency matrix (in $[-1, 1]$).
- $\sigma_1 \geq \sigma_2 \geq \cdots$ are the singular values of a graph's adjacency matrix (in $[0, 1]$).

SOS is an analytic tool so we will identify [8] words over $\mathbb{F}_2$ with words over $\{\pm 1\}$. We also make some choices for words and local variables to distinguish the ground space $\mathbb{F}_2^{X(1)}$ or $\{\pm 1\}^{X(1)}$ form the lifted space $\mathbb{F}_2^{X(k)}$ or $\{\pm 1\}^{X(k)}$.

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

Unique Decoding. Given $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ with the lifting function as k-XOR, we can view the problem of finding the closest codeword to a given $\tilde{y} \in \mathbb{F}_2^{X(k)}$ 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 $\mathfrak{s} \in X(k)$. By this property, $y = \mathrm{dsum}(z)$ is the unique codeword of $\mathcal{C}_k$ closest to $\tilde{y}$. Using the results of [AJT19], it is indeed possible to find $z' \in \mathbb{F}_2^n$ such that $\Delta(\mathrm{dsum}(z'), \tilde{y}) \leq \Delta(\mathrm{dsum}(z), \tilde{y}) + \beta$ for any $\beta > 0$. We then argue

[8] For this, we can use any bijection from $\mathbb{F}_2 \to \{\pm 1\}$.

that z' or its complement $\overline{z'}$ must be close to $z \in \mathcal{C}_1$, which can then be recovered by unique decoding.

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 $\mathrm{dsum}(z - z')$ has bias close to zero, i.e., $\Delta(\mathrm{dsum}(z), \mathrm{dsum}(z')) \approx 1/2$. However, if $\Delta(\tilde{y}, \mathcal{C}_k) \leq \eta$, then we must have

$$\Delta(\mathrm{dsum}(z), \mathrm{dsum}(z')) \;\leq\; \Delta(\mathrm{dsum}(z), \tilde{y}) + \Delta(\mathrm{dsum}(z'), \tilde{y}) \;\leq\; 2\eta + \beta\,,$$

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

List Decoding. We start by describing an abstract list decoding framework which only assumes two general properties of a lifting $\mathrm{lift}^{g}_{X(k)}$: (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 $(1/2 - \sqrt{\varepsilon})$-close to a lifted code $\mathcal{C}_k = \mathrm{lift}(\mathcal{C}_1)$ where $\mathcal{C}_k$ has distance at least $1/2 - \varepsilon$ and $\mathcal{C}_1$ has distance at least $1/2 - \varepsilon_0$. By list decoding $\tilde{y}$, we mean finding a list $\mathcal{L} \subseteq \mathcal{C}_k$ of all codewords $(1/2 - \sqrt{\varepsilon})$-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 [9]. 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 $\Psi := \mathbb{E}_{\mathfrak{s},\mathfrak{t} \in X(k)} \left[\left(\widetilde{\mathbb{E}}[\mathbf{Y}_{\mathfrak{s}} \mathbf{Y}_{\mathfrak{t}}] \right)^2 \right]$.

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 [10] 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 $|\langle y', y \rangle| \geq 2 \cdot \varepsilon$. 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 $(z, y = \mathrm{lift}(z))$ and $(z', y' = \mathrm{lift}(z'))$ 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 $y = \mathrm{lift}(z)$. Furthermore, if

[9] The precise SOS program used is given in Section 6.2.

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

$\mathcal{C}_1$ 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 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.

To recapitulate, the three stages of the abstract list decoding framework are summarized in Fig. 1 along with the required assumptions on the lifting.

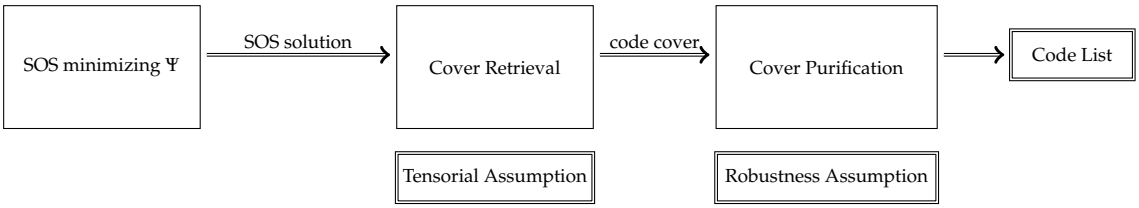

Figure 1: List decoding framework with the assumptions required in each stage.

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 $\mathcal{C}_k$ on 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 part of 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 using $X(i)$ has correlation at least ε with the word y' defined as $y'_{\mathfrak{t}} = \chi_t(\tilde{y}_{\mathfrak{s}})$ 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. Before addressing list decoding, 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 is given in Section 6. Next, we instantiate the framework with the direct sum lifting on HDXs in Section 7. As an interlude between the first and second instantiation, Section 8 describes how the first concrete instantiation of Section 7 captures the direct product lifting on HDXs via a reduction to the direct sum lifting. Finally, in Section 9, we show how to instantiate the framework with the direct sum lifting on the collection of length $k-1$ walks of an expander graph.

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 (Walks on Expanders are 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 a $(\beta_0, (\beta_0 + 2\lambda)^{\lfloor k/2 \rfloor})$-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.

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 4.2 (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 4.3 (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 4.4 (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-HDXs do exist, with flat distributions on their sets of faces. The following lemma from [DHK+19] is a refinement of [DK17].

Lemma 4.5 (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 [11]. We first show that the expectation over k-sized faces 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 4.6 (Complete complex and 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

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

Proof. Let $\mathcal{E} = \{(i_1,\dots,i_k) \in X(1)^k \mid i_1,\dots,i_k \text{ are distinct}\}$, $\delta = \mathbb{P}_{i_1,\dots,i_k\sim\Pi_1}[(i_1,\dots,i_k) \notin \mathcal{E}]$, and $\eta = \mathbb{E}_{(i_1,\dots,i_k)\in X(1)^k\setminus\mathcal{E}}[f(i_1)\cdots f(i_k)]$. Then

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

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

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

so we have

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

■

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 4.7 (Sampler)**.** *Let $G = (U, V, E)$ be a bipartite graph with a probability distribution Π_U on U. Let Π_V be the distribution on V obtained by choosing $u \in U$ according to Π_U, then a uniformly random neighbor v of u. We say that G is an (η,δ)-sampler if for every function $f\colon V \to [0,1]$ with $\mu = \mathbb{E}_{v\sim\Pi_V} f(v)$,*

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

[11] 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 4.8 (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 4.6, 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 4.9 (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 β_0, then*

$$\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) \leq (\beta_0 + \eta)^k + \frac{k^2}{d} + \delta.$$

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

$$\begin{aligned}
\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) &= \left|\mathbb{E}_{\{i_1,\dots,i_k\}\sim\Pi_k}(-1)^{z_{i_1}+\cdots+z_{i_k}}\right| \\
&= \left|\mathbb{E}_{t\sim\Pi_d}\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_t(k)}(-1)^{z_{i_1}+\cdots+z_{i_k}}\right| \\
&\leq \left|\mathbb{E}_{t\sim\Pi_d}\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_t(k)}(-1)^{z_{i_1}+\cdots+z_{i_k}}\mathbb{1}_{[\mathrm{bias}(z|_t)\leq\beta_0+\eta]}\right| \\
&\quad + \mathop{\mathbb{P}}_{t\sim\Pi_d}[\mathrm{bias}(z|_t) > \beta_0 + \eta] \\
&\leq \mathbb{E}_{t\sim\Pi_d}\mathbb{1}_{[\mathrm{bias}(z|_t)\leq\beta_0+\eta]}\left|\mathbb{E}_{\{i_1,\dots,i_k\}\in X|_t(k)}(-1)^{z_{i_1}+\cdots+z_{i_k}}\right| + \delta.
\end{aligned}$$

By Claim 4.6, the magnitude of the expectation of $(-1)^{z_i}$ over the edges of size k in the complete complex $X|_t$ is close to $\left|\mathbb{E}_{i\sim X|_t(1)}(-1)^{z_i}\right|$, which is just the bias of $z|_t$. Then

$$\begin{aligned}
\mathrm{bias}(\mathrm{dsum}_{X(k)}(z)) &\leq \mathbb{E}_{t\sim X(d)}\mathbb{1}_{[\mathrm{bias}(z|_t)\leq\beta_0+\eta]}\,\mathrm{bias}(z|_t)^k + \frac{k^2}{d} + \delta \\
&\leq (\beta_0 + \eta)^k + \frac{k^2}{d} + \delta
\end{aligned}$$

■

Now we can compute the parameters necessary for a HDX to be an (β_0, β)-parity sampler for arbitrarily small β.

Lemma 4.10 (HDXs are parity samplers). *Let $0 < \beta \leq \beta_0 < 1$, $0 < \theta < (1/\beta_0) - 1$, and $k \geq \log_{(1+\theta)\beta_0}(\beta/3)$. If $X(\leq d)$ is a γ-HDX with $d \geq \max\{3k^2/\beta, 6/(\theta^2\beta_0^2\beta)\}$ and $\gamma = O\left(1/d^2\right)$, then $X(k)$ is a (β_0, β)-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 $\eta = \theta\beta_0$ and $\delta = \beta/3$. Using Fact 4.8 to obtain a sampler with these parameters, we need the second singular value σ_2 of $M_{1,d}$ to be bounded as

$$\sigma_2 \leq \theta\beta_0\sqrt{\frac{\beta}{3}}.$$

By the upper bound on σ_2^2 from Theorem 4.3, it suffices to have

$$\frac{1}{d} + O(d\gamma) \leq \frac{\theta^2\beta_0^2\beta}{3},$$

which is satisfied by taking $d \geq 6/\left(\theta^2\beta_0^2\beta\right)$ and $\gamma = O\left(1/d^2\right)$.

By Claim 4.9, $X(k)$ is a $(\beta_0, (\beta_0+\eta)^k + k^2/d + \delta)$-parity sampler. The first term in the bias is $(\beta_0+\eta)^k = ((1+\theta)\beta_0)^k$, so we require $(1+\theta)\beta_0 < 1$ to amplify the bias by making k large. To make this term smaller than $\beta/3$, k must be at least $\log_{(1+\theta)\beta_0}(\beta/3)$. We already chose $\delta = \beta/3$, so ensuring $d \geq 3k^2/\beta$ gives us a (β_0, β)-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 β_0, parity sampling allows us to obtain a code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ with arbitrarily small bias β 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 4.11 (Rate of direct sum lifting for a HDX)**.** *Let $\beta_0 \in (0,1)$ and $\theta \in (0, (1/\beta_0) - 1)$ be constants, and let $\mathcal{C}_1$ be an β_0-biased binary linear code with relative rate r_1. For $\beta \in (0, \beta_0]$, suppose k, d, and γ satisfy the hypotheses of Lemma 4.10, 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 β constructed on a HDX with these parameters satisfies*

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

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

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

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 4.5, 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 β_0 and θ as constants, the values of k and d necessary to satisfy Lemma 4.10 are

$$k = \log_{(1+\theta)\beta_0}(\beta/3) = O(\log(1/\beta))$$

and

$$d = \max\left\{\frac{3k^2}{\beta}, \frac{6}{\theta^2\beta_0^2\beta}\right\} = O\left(\frac{(\log(1/\beta))^2}{\beta}\right).$$

Putting this expression for d into the inequality for D yields

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

from which the bounds in the lemma statement follow. ∎

From Lemma 4.11, we see that if $\mathcal{C}_1$ has constant rate, then $\mathcal{C}_k$ has a rate which is constant with respect to n. However, the dependence of the rate on the bias β 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 4.12 (Rate of direct sum lifting for expander walks [TS17])**.** *Let $\beta_0 \in (0,1)$ be a constant and $\mathcal{C}_1$ be an β_0-biased binary linear code with relative rate r_1. Fix $\beta \in (0, \beta_0]$. Suppose G is a graph with second largest singular value $\lambda = \beta_0/2$ and degree $d \leq 4/\lambda^2$. Let $k = 2\log_{2\beta_0}(\beta) + 1$ 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 β and rate $r_k \geq r_1 \cdot \beta^{O(1)}$.*

Proof. From Theorem 4.1 with this choice of λ and k, the direct sum lifting $\mathcal{C}_k$ has bias β. 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 $d \leq 16/\beta_0$, which is a constant, and $k = O(\log(1/\beta))$, the rate satisfies $r_k \geq r_1 \cdot \beta^{O(1)}$. ∎

5 Unique Decoding

In this section, we will show how parity sampling and the ability to solve k-XOR instances with $X(k)$ as their constraint complex allow us to decode the direct sum lifting $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ of a linear base code $\mathcal{C}_1 \in \mathbb{F}_2^n$. With a more technical argument, we can also handle different kinds of liftings and non-linear codes, 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 $\bar{z}$ will be close to $z^\star$. Running the unique decoding algorithm for $\mathcal{C}_1$ on z and $\bar{z}$ will recover $z^\star$, from which we can obtain $y^\star$ by applying the direct sum lifting.

Lemma 5.1. *Let $0 < \varepsilon < 1/2$ and $0 < \beta < 1/4 - \varepsilon/2$. 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 $(1/2 + 2\mu_0, 2\varepsilon)$-parity sampler. Let $\tilde{y} \in \mathbb{F}_2^{X(k)}$ be a word that has distance strictly less than $(1/4 - \varepsilon/2 - \beta)$ 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

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

we have either

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

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

Remark 5.2. *Since $\mathrm{dsum}_{X(k)}$ is a $(1/2 + 2\mu_0, 2\varepsilon)$-parity sampler, the code $\mathcal{C}_k$ has distance $\Delta(\mathcal{C}_k) \geq 1/2 - \varepsilon$. 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

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

By linearity of $\mathrm{dsum}_{X(k)}$, $\Delta(\mathrm{dsum}_{X_k}(z^\star - z), 0) < 1/2 - \varepsilon$, so $\mathrm{bias}(\mathrm{dsum}_k(z^\star - z)) > 2\varepsilon$. From the $(1/2 + 2\mu_0, 2\varepsilon)$-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 $\Delta(z^\star, \bar{z}) < 1/4 - \mu_0$. ∎

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 $\mathcal{C}_k$.

Corollary 5.3. *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*

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

then unique decoding either z or $\bar{z}$ 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

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

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

5.2 Concrete Instantiations

High Dimensional Expanders

If $X(k)$ is the collection of k-faces of a sufficiently expanding γ-HDX, we can use the following algorithm to approximately solve the k-XOR instance $\mathfrak{I}(X(k), \tilde{y})$ and obtain $z \in \mathbb{F}_2^n$.

Theorem 5.4 ([AJT19]). *Let $\mathfrak{I}$ be an instance of MAX k-CSP on n variables taking values over an alphabet of size q, and let $\beta > 0$. Let the simplicial complex $X_{\mathfrak{I}}$ be a γ-HDX with $\gamma = \beta^{O(1)} \cdot (1/(kq))^{O(k)}$.*

There is an algorithm based on $(k/\beta)^{O(1)} \cdot q^{O(k)}$ levels of the Sum-of-Squares hierarchy which produces an assignment satisfying at least an $(\mathsf{OPT}_{\mathfrak{I}} - \beta)$ fraction of the constraints in time $n^{(k/\beta)^{O(1)} \cdot q^{O(k)}}$.

If X is a HDX with the parameters necessary to both satisfy this theorem and be a $(1/2 + 2\mu_0, 2\varepsilon)$ parity sampler, we can combine this with Corollary 5.3 to achieve efficient unique decodability of $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$.

Corollary 5.5. *Let $X(\leq d)$ be a d-dimensional γ-HDX satisfying the premises of Lemma 4.10 that would guarantee that $X(k)$ is a $(1/2 + 2\mu_0, 2\varepsilon)$-parity sampler, and let $\mathcal{C}_1 \subseteq \mathbb{F}_2^n$ be a linear code which is efficiently unique decodable within radius $1/4 - \mu_0$ for some $\mu_0 > 0$. Then the code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ can be unique decoded within distance $1/4 - \varepsilon/2 - \beta$ in time $n^{(k/\beta)^{O(1)} \cdot 2^{O(k)}}$,[12] where we have*

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

Proof. By Lemma 4.10, we can achieve $(1/2 + 2\mu_0, 2\varepsilon)$-parity sampling by taking $0 < \theta < \frac{2}{1+4\mu_0} - 1$, $k \geq \log_{(1+\theta)\cdot(\frac{1}{2}+2\mu_0)}(2\varepsilon/3)$, $d \geq \max\left\{\frac{3k^2}{2\varepsilon}, \frac{3}{\theta^2(1/2+2\mu_0)^2\varepsilon}\right\}$, and $\gamma = O(1/d^2)$. Let $\tilde{y} \in \mathbb{F}_2^{X(k)}$ be a received word with distance less than $(1/4 - \varepsilon/2 - \beta)$ from $\mathcal{C}_k$. Applying Theorem 5.4 to $\mathfrak{I}(X(k), \tilde{y})$ with $q = 2$ and the given value of β, we obtain a $z \in \mathbb{F}_2^n$ with $\mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z) \geq \mathsf{OPT}_{\mathfrak{I}(X(k),\tilde{y})} - \beta$. This z can be used in Corollary 5.3 to find z^* and uniquely decode $\tilde{y}$ as $y^* = \mathrm{dsum}_{X(k)}(z^*)$. ∎

Expander Walks

In Section 9, we will show that the algorithmic results of [AJT19] can be modified to work when $X(k)$ is a set of tuples of size k which is sufficiently splittable (Corollary 9.21), which occurs when $X(k)$ is a set of walks on on a suitably strong expander (Corollary 9.18). In particular, we have the following.

Theorem 5.6. *Let $G = (V, E)$ be a graph with $\sigma_2(G) = \lambda$ and k be a given parameter. Let $\mathfrak{I}$ be a k-CSP instance over an alphabet of size q whose constraint graph is the set of walks on G of length k. Let $\beta > 0$ be such that $\lambda = O(\beta^2/(k^2 \cdot q^{2k}))$.*

There exists an algorithm based on $O\left(\frac{q^{4k}k^7}{\beta^5}\right)$ levels of the Sum-of-Squares hierarchy which produces an assignment satisfying at least an $(\mathsf{OPT}_{\mathfrak{I}} - \beta)$ fraction of the constraints in time $n^{O(q^{4k} \cdot k^7/\beta^5)}$.

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

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 5.7. *Let $X(k)$ be the set of walks on a graph G with $\sigma_2(G) = \lambda$ such that $\mathrm{dsum}_{X(k)}$ is a $(1/2 + 2\mu_0, 2\varepsilon)$ parity sampler, and let $\mathcal{C}_1 \subseteq \mathbb{F}_2^n$ be a linear code which is efficiently unique decodable within radius $1/4 - \mu_0$ for some $\mu_0 > 0$. Then the code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ can be unique decoded within radius $1/4 - \varepsilon/2 - \beta$ in time $n^{O(2^{4k} \cdot k^7/\beta^5)}$, where we have*

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

Proof. By Theorem 4.1, we can obtain a $(1/2 + 2\mu_0, 2\varepsilon)$-parity sampler by ensuring $1/2 + \mu_0 + 2\lambda < 1$ and $k \geq 2\log_{1/2+\mu_0+2\lambda}(2\varepsilon) + 1$. Let $\tilde{y} \in \mathbb{F}_2^{X(k)}$ be a received word with distance less than $(1/4 - \varepsilon/2 - \beta)$ from $\mathcal{C}_k$. Applying Theorem 5.6 to $\mathfrak{I}(X(k), \tilde{y})$ with $q = 2$ and the given value of β, we obtain a $z \in \mathbb{F}_2^n$ with $\mathsf{SAT}_{\mathfrak{I}(X(k),\tilde{y})}(z) \geq \mathsf{OPT}_{\mathfrak{I}(X(k),\tilde{y})} - \beta$. This z can be used in Corollary 5.3 to find z^* and uniquely decode $\tilde{y}$ as $y^* = \mathrm{dsum}_{X(k)}(z^*)$. ∎

Remark 5.8. *In both Corollary 5.5 and Corollary 5.7, when μ_0 and ε are constants, k can be constant, which means we can decode $\mathcal{C}_k$ from a radius arbitrarily close to $1/4 - \varepsilon/2$ 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 $1/4 - \varepsilon/2$. Our choice of $(1/2 + 2\mu_0, 2\varepsilon)$-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 2ε. In this case, the maximum distance at which our unique decoding algorithm works will be smaller than $\Delta(\mathcal{C}_k)/2$.

6 Abstract List Decoding Framework

In this section, we present the abstract list decoding framework with its requirements and prove its guarantees. We introduce the entropic proxy Ψ in Section 6.1 and use it to define the SOS program for list decoding in Section 6.2. In Section 6.3, we establish key properties of Ψ capturing its importance as a list decoding tool. We recall the Propagation Rounding algorithm in Section 6.4 and formalize the notion of a slice as a set of assignments to variables in the algorithm. Then, considerations of SOS tractability of the lifting related to tensorial properties are dealt with in Section 6.5. Now, assuming we have a fractional SOS solution to our program, the analysis of its covering properties and the precise definition and correctness of the two later stages of the framework are given in Section 6.6. This abstract framework will be instantiated using the direct sum lifting: on HDXs in Section 7 and on expander walks in Section 9.

6.1 Entropic Proxy

In our list decoding framework via SOS, we will solve a single optimization program whose resulting pseudo-expectation will in a certain sense be rich enough to cover all

intended solutions at once. To enforce this covering property we rely on an analytical artifice, namely, we minimize a convex function Ψ that provides a proxy to how concentrated the SOS solution is. More precisely, we use Ψ from Definition 6.1. A similar list decoding technique was also (independently) used by Karmalkar et al. [KKK19] and Raghavendra–Yau [RY19], but in the context of learning.

Definition 6.1 (Entropic Proxy)**.** *Let $\mathbf{Y} = \{\mathbf{Y}_S\}_{S \in X(k)}$ be a t-local PSD ensemble with $t \geq 2$. We define $\Psi = \Psi\left(\{\mathbf{Y}_S\}_{S \in X(k)}\right)$ as*

$$\Psi \;:=\; \mathbb{E}_{S,T \sim \Pi_k} \left(\widetilde{\mathbb{E}}\left[\mathbf{Y}_S \mathbf{Y}_T\right]\right)^2.$$

We also denote Ψ equivalently as $\Psi = \Psi\left(\widetilde{\mathbb{E}}\right)$ where $\widetilde{\mathbb{E}}$ is the pseudo-expectation operator associated to the ensemble $\mathbf{Y}$.

6.2 SOS Program for List Decoding

Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ be a word promised to be $(1/2 - \sqrt{\varepsilon})$-close to a lifted code $\mathcal{C}_k = \mathrm{lift}(\mathcal{C}_1)$. The word $\tilde{y}$ is to be regarded as a (possibly) corrupted codeword for which we want to do list decoding. We consider the following SOS program.

minimize	$\Psi\left(\{\mathbf{Y}_S\}_{S \in X(k)}\right)$	(List Decoding Program)
subject to		
	$\mathbb{E}_{S \sim \Pi_k} \widetilde{\mathbb{E}}\left[\tilde{y}_S \cdot \mathbf{Y}_S\right] \geq 2\sqrt{\varepsilon}$	(Agreement Constraint)
	$\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ being $(L + 2k)$-local PSD ensemble	

Table 1: List decoding SOS formulation for $\tilde{y}$.

6.3 Properties of the Entropic Proxy

We establish some key properties of our negative entropic function Ψ. First, we show that Ψ is a convex function. Since the feasible set defined by the SOS List Decoding Program is convex and admits an efficient separation oracle [13], the convexity of Ψ implies that the List Decoding Program can be efficiently solved within η-optimality in time $n^{O(t)} \cdot \mathrm{polylog}(\eta^{-1})$ where t is the SOS degree.

Lemma 6.2 (Convexity)**.** *Ψ is convex, i.e., for every pair of pseudo-expectations $\widetilde{\mathbb{E}}_1$ and $\widetilde{\mathbb{E}}_2$ and $\alpha \in [0,1]$,*

$$\Psi\left(\alpha \cdot \widetilde{\mathbb{E}}_1 + (1-\alpha) \cdot \widetilde{\mathbb{E}}_2\right) \;\leq\; \alpha \cdot \Psi\left(\widetilde{\mathbb{E}}_1\right) + (1-\alpha) \cdot \Psi\left(\widetilde{\mathbb{E}}_2\right).$$

Proof. Suppose $S \cup T = \{i_1, \ldots, i_t\}$. By definition $\mathbf{Y}_S \mathbf{Y}_T = \mathrm{lift}(\mathbf{Z})_S \cdot \mathrm{lift}(\mathbf{Z})_T$, i.e., $\mathbf{Y}_S \mathbf{Y}_T$ is a function f on input $\mathbf{Z}_{i_1}, \ldots, \mathbf{Z}_{i_t} \in \{\pm 1\}$. Let

$$f(\mathbf{Z}_{i_1}, \ldots, \mathbf{Z}_{i_t}) = \sum_{S \subseteq S \cup T} \widehat{f}(S) \cdot \prod_{i \in S} \mathbf{Z}_i,$$

[13] In our setting the pseudo-expectation has trace bounded by $n^{O(t)}$ in which case semidefinite programming can be solved efficiently [GM12, RW17].

be the Fourier decomposition of f. Then

$$\widetilde{\mathbb{E}}\left[\mathbf{Y}_S\mathbf{Y}_T\right] = \widetilde{\mathbb{E}}\left[f\right] = \sum_{S\subseteq S\cup T}\widehat{f}(S)\cdot\widetilde{\mathbb{E}}\left[\prod_{i\in S}\mathbf{Z}_i\right].$$

Since $\widetilde{\mathbb{E}}\left[\mathbf{Y}_S\mathbf{Y}_T\right]$ is a linear function of $\widetilde{\mathbb{E}}$, we obtain $\left(\widetilde{\mathbb{E}}\left[\mathbf{Y}_S\mathbf{Y}_T\right]\right)^2$ is convex. Now, the convexity of Ψ follows by noting that Ψ is a convex combination of convex functions. ∎

The (sole) problem-specific constraint appearing in the SOS List Decoding Program allows us to deduce a lower bound on Ψ. This lower bound will be important later to show that a feasible solution that does not cover all our intended solutions must have Ψ bounded away from 0 so that we still have room to decrease Ψ. We note that an improvement in the conclusion of the following lemma would directly translate to stronger list decoding parameters in our framework.

Lemma 6.3 (Correlation ⇒ entropic bound)**.** *Let $\{\mathbf{Y}_S\}_{S\in X(k)}$ be t-local PSD ensemble with $t \geq 2$. If there is some $y \in \{\pm 1\}^{X(k)}$ such that*

$$\left|\mathbb{E}_{S\sim\Pi_k}\widetilde{\mathbb{E}}\left[y_S\cdot\mathbf{Y}_S\right]\right| \geq \beta,$$

then

$$\Psi\left(\{\mathbf{Y}_S\}_{S\in X(k)}\right) \;\geq\; \beta^4.$$

Proof. We calculate

$$\begin{aligned}
\mathbb{E}_{S,T\sim\Pi_k}\left(\widetilde{\mathbb{E}}\left[\mathbf{Y}_S\mathbf{Y}_T\right]\right)^2 &= \mathbb{E}_{S,T\sim\Pi_k}\left(\widetilde{\mathbb{E}}\left[\left(y_S\mathbf{Y}_S\right)\left(y_T\mathbf{Y}_T\right)\right]\right)^2 \\
&\geq \left(\mathbb{E}_{S,T\sim\Pi_k}\widetilde{\mathbb{E}}\left[\left(y_S\mathbf{Y}_S\right)\left(y_T\mathbf{Y}_T\right)\right]\right)^2 \qquad \text{(Jensen's Inequality)} \\
&= \left(\widetilde{\mathbb{E}}\left[\left(\mathbb{E}_{S\sim\Pi_k}y_S\cdot\mathbf{Y}_S\right)^2\right]\right)^2 \\
&\geq \left(\widetilde{\mathbb{E}}\left[\mathbb{E}_{S\sim\Pi_k}\left[y_S\cdot\mathbf{Y}_S\right]\right]\right)^4 \qquad \text{(Cauchy–Schwarz Inequality)} \\
&= \left(\mathbb{E}_{S\sim\Pi_k}\widetilde{\mathbb{E}}\left[y_S\cdot\mathbf{Y}_S\right]\right)^4 \;\geq\; \beta^4.
\end{aligned}$$

∎

We now show the role of Ψ in list decoding: if an intended solution is not represented in the pseudo-expectation $\widetilde{\mathbb{E}}$, we can get a new pseudo-expectation $\widetilde{\mathbb{E}}'$ which attains a smaller value of Ψ.

Lemma 6.4 (Progress lemma)**.** *Suppose there exist $z \in \{\pm 1\}^{X(1)}$ and $y = \mathrm{lift}(z) \in \{\pm 1\}^{X(k)}$ satisfying*

$$\widetilde{\mathbb{E}}\left[\left(\mathbb{E}_{S\sim\Pi_k}y_S\cdot\mathbf{Y}_S\right)^2\right] \leq \delta^2.$$

If $\Psi \geq \delta^2$, then there exists a pseudo-expectation $\widetilde{\mathbb{E}}'$ such that

$$\mathbb{E}_{S,T\sim\Pi_k}\left(\widetilde{\mathbb{E}}'\left[\mathbf{Y}_S\mathbf{Y}_T\right]\right)^2 \;\leq\; \Psi - \frac{\left(\Psi-\delta^2\right)^2}{2}.$$

In particular, if $\Psi \geq 2\delta^2$*, then*

$$\mathbb{E}_{s,t\sim\Pi_k}\left(\widetilde{\mathbb{E}}'\left[\mathbf{Y}_s\mathbf{Y}_t\right]\right)^2 \;\leq\; \Psi - \frac{\delta^4}{2}.$$

Proof. Let $\widetilde{\mathbb{E}}'$ be the pseudo-expectation [14]

$$\widetilde{\mathbb{E}}' \;:=\; (1-\alpha)\cdot\widetilde{\mathbb{E}} + \alpha\cdot\mathbb{E}_{\delta_z},$$

where $\mathbb{E}_{\delta_z}$ is the expectation of the delta distribution on z and $\alpha \in (0,1)$ is to be defined later. We have

$$\begin{aligned}
\mathbb{E}_{s,t\sim\Pi_k}\left(\widetilde{\mathbb{E}}'\left[\mathbf{Y}_s\mathbf{Y}_t\right]\right)^2 &= \mathbb{E}_{s,t\sim\Pi_k}\left((1-\alpha)\cdot\widetilde{\mathbb{E}}\left[\mathbf{Y}_s\mathbf{Y}_t\right] + \alpha\cdot y_s y_t\right)^2 \\
&= (1-\alpha)^2\cdot\Psi + \alpha^2\cdot\mathbb{E}_{s,t\sim\Pi_k}(y_s y_t)^2 + 2\alpha(1-\alpha)\cdot\mathbb{E}_{s,t\sim\Pi_k}\left[\widetilde{\mathbb{E}}[\mathbf{Y}_s\mathbf{Y}_t]y_s y_t\right] \\
&\leq (1-\alpha)^2\cdot\Psi + \alpha^2 + 2\alpha(1-\alpha)\cdot\delta^2.
\end{aligned}$$

The value of α minimizing the quadratic expression of the RHS above is

$$\alpha^\star \;=\; \frac{\Psi-\delta^2}{1+\Psi-2\delta^2}.$$

Using this value yields

$$\begin{aligned}
\mathbb{E}_{s,t\sim\Pi_k}\left(\widetilde{\mathbb{E}}'\left[\mathbf{Y}_s\mathbf{Y}_t\right]\right)^2 &\leq \Psi - \frac{\left(\Psi-\delta^2\right)^2}{1+\Psi-2\delta^2} \\
&\leq \Psi - \frac{\left(\Psi-\delta^2\right)^2}{2},
\end{aligned}$$

where in the last inequality we used $\Psi \leq 1$. ∎

6.4 Propagation Rounding

A central algorithm in our list decoding framework is the Propagation Rounding Algorithm 6.5. It was studied by Barak et al. [BRS11] in the context of approximating 2-CSPs on low threshold rank graphs and it was later generalized to HDXs (and low threshold rank hypergraphs) in the context of k-CSPs [AJT19].

Given an $(L+2k)$-local PSD ensemble $\{\mathbf{Z}_1,\ldots,\mathbf{Z}_n\}$, the Propagation Rounding Algorithm 6.5 chooses a subset of variables $S \subseteq [n]$ at random. Then it samples a joint assignment σ to the variables in S according to $\{\mathbf{Z}_S\}$. The value of the remaining variables $\mathbf{Z}_i$ are sampled according to the conditional marginal distributions $\{\mathbf{Z}_i|\mathbf{Z}_S=\sigma\}$. An important byproduct of this algorithm is the $2k$-local PSD ensemble $\mathbf{Z}' = \{\mathbf{Z}_1,\ldots,\mathbf{Z}_n|\mathbf{Z}_S=\sigma\}$.

The precise description of the Propagation Rounding Algorithm 6.5 follows.

[14] By summing the pseudo-expectation $\widetilde{\mathbb{E}}$ and actual expectation $\mathbb{E}_{\delta_z}$, we mean that we are summing $\widetilde{\mathbb{E}}$ to pseudo-expectation of the same dimensions obtained from operator $\mathbb{E}_{\delta_z}$.

Algorithm 6.5 (Propagation Rounding Algorithm).
Input *An $(L+2k)$-local PSD ensemble $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ and some distribution Π_k on $X(k)$.*
Output *A random assignment $(\sigma_1, \ldots, \sigma_n) \in [q]^n$ and $2k$-local PSD ensemble $\mathbf{Z}'$.*

1. *Choose $m \in \{1, \ldots, L/k\}$ uniformly at random.*
2. *Independently sample m k-faces, $\mathfrak{s}_j \sim \Pi_k$ for $j = 1, \ldots, m$.*
3. *Write $S = \bigcup_{j=1}^{m} \mathfrak{s}_j$, for the set of the seed vertices.*
4. *Sample assignment $\sigma : S \to [q]$ according to the local distribution $\{\mathbf{Z}_S\}$.*
5. *Set $\mathbf{Z}' = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_n | \mathbf{Z}_S = \sigma\}$, i.e. the local ensemble $\mathbf{Z}$ conditioned on agreeing with σ.*
6. *For all $j \in [n]$, sample independently $\sigma_j \sim \{\mathbf{Z}'_j\}$.*
7. *Output $(\sigma_1, \ldots, \sigma_n)$ and $\mathbf{Z}'$.*

To our list decoding task we will show that an ensemble minimizing Ψ covers the space of possible solutions in the sense that for any intended solution there will be a choice of S and σ such that the conditioned ensemble $\mathbf{Z}'$ enables the sampling of a word within the unique decoding radius in $\mathcal{C}_1$ of this intended solution.

An execution of the Algorithm 6.5 is completely determined by the tuple (m, S, σ) which we will refer to as a slice of the PSD ensemble.

Definition 6.6 (Slice). *We call a tuple (m, S, σ) obtainable by Algorithm 6.5 a slice and let Ω denote the set of all slices obtainable by Algorithm 6.5.*

We can endow Ω with a natural probability distribution, where the measure of each (m, S, σ) is defined as the probability that this slice is picked during an execution of Algorithm 6.5. We also define a pseudo-expectation operator for each slice.

Definition 6.7 (Pseudo-Expectation Slice). *Given a slice (m, S, σ), we define the pseudo-expectation operator $\widetilde{\mathbb{E}}_{|_{S,\sigma}}$ which is the pseudo-expectation operator of the conditioned local PSD ensemble $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n | \mathbf{Z}_s = \sigma\}$.*

6.5 Tensorial Structures

In general, a local PSD ensemble $\mathbf{Z}' = \{\mathbf{Z}'_1, \ldots, \mathbf{Z}'_n\}$ output by the Propagation Rounding Algorithm 6.5 may be far from corresponding to any underlying joint global distribution [15]. In our application, we will be interested in the case where the ensemble approximately behaves as being composed of independent random variables over the collection of "local views" given by the hyperedges in $X(k)$. In such case, rounding the SOS solution via independent rounding is straightforward. A collection of local views admitting this property with a given SOS degree parameter L is denoted *tensorial* (variables behave as products over the local views).

Definition 6.8 (Tensorial Hypergraphs). *Let $X(k)$ be a collection of k-uniform hyperedges endowed with a distribution Π_k, $\mu \in [0, 1]$, and $L \in \mathbb{N}$. We say that $X(k)$ is (μ, L)-tensorial if*

[15] In fact, if this was the case, then we would be able to approximate any k-CSP with SOS degree $(L + 2k)$. However, even for L as large as linear in n this is impossible for SOS [Gri01, KMOW17].

the local PSD ensemble $\mathbf{Z}'$ *returned by Propagation Rounding Algorithm 6.5 with SOS degree parameter* L *satisfies*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \left\| \{\mathbf{Z}'_{\mathfrak{a}}\} - \{\mathbf{Z}'_{a_1}\} \cdots \{\mathbf{Z}'_{a_k}\} \right\|_1 \leq \mu. \tag{1}$$

To analyze the potential Ψ we will need that the variables between pairs of local views, i.e., pairs of hyperedges, behave as product.

Definition 6.9 (Two-Step Tensorial Hypergraphs)**.** *Let* $X(k)$ *be a collection of* k*-uniform hyperedges endowed with a distribution* Π_k*,* $\mu \in [0,1]$*, and* $L \in \mathbb{N}$*. We say that* $X(k)$ *is* (μ, L)*-two-step tensorial if it is* (μ, L)*-tensorial and the PSD ensemble* $\mathbf{Z}'$ *returned by Propagation Rounding Algorithm 6.5 with SOS degree parameter* L *satisfies*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k} \left\| \{\mathbf{Z}'_{\mathfrak{s}}\mathbf{Z}'_{\mathfrak{t}}\} - \{\mathbf{Z}'_{\mathfrak{s}}\}\{\mathbf{Z}'_{\mathfrak{t}}\} \right\|_1 \leq \mu.$$

In Section 7.1, we establish the relationship between the parameters μ and L and the expansion that will ensure HDXs are (μ, L)-two-step tensorial. Similarly, in Section 9.1.4 we provide this relationship when $X(k)$ is the collection of walks of an expander graph.

Tensorial over Most Slices

By choosing μ sufficiently small it is easy to show that most slices (m, S, σ) satisfy the tensorial (or two-step tensorial) statistical distance condition(s) with a slightly worse parameter $\tilde{\mu}$ such that $\tilde{\mu} \to 0$ as $\mu \to 0$. If we could construct tensorial (or two-step tensorial) objects for arbitrarily small parameter μ with $L = O_{k,q,\mu}(1)$, then we would be able to obtain $\tilde{\mu}$ arbitrarily small. Lemma 7.4 establishes that HDXs of appropriate expansion satisfy this assumption, and Lemma 9.20 does the same for walks on expanders.

We introduce two events. The first event captures when a slice (m, S, σ) leads to the conditioned local variables $\mathbf{Z}'_1, \ldots, \mathbf{Z}'_n$ being close to k-wise independent over the k-sized hyperedges.

Definition 6.10 (Ground Set Close to k-wise Independent)**.** *Let* $\mu \in (0,1]$*. We define the event* K_μ *as*

$$K_\mu := \left\{ (m, S, \sigma) \in \Omega \mid \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \left\| \{\mathbf{Z}_{\mathfrak{a}} | \mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{a_1} | \mathbf{Z}_S = \sigma\} \cdots \{\mathbf{Z}_{a_k} | \mathbf{Z}_S = \sigma\} \right\|_1 < \mu^2/2 \right\}.$$

The second event captures when the variables between pairs of hyperedges are close to independent.

Definition 6.11 (Lifted Variables Close to Pairwise Independent)**.** *Let* $\mu \in (0,1]$*. We define the event* P_μ *as*

$$P_\mu := \left\{ (m, S, \sigma) \in \Omega \mid \mathop{\mathbb{E}}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k} \left\| \{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}} | \mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{\mathfrak{s}} | \mathbf{Z}_S = \sigma\}\{\mathbf{Z}_{\mathfrak{t}} | \mathbf{Z}_S = \sigma\} \right\|_1 < \mu^2/2 \right\}.$$

These events satisfy a simple concentration property.

Claim 6.12 (Concentration). *Suppose a simplicial complex $X(\leq k)$ with $X(1) = [n]$ and an $(L + 2k)$-local PSD ensemble $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ are given as input to Propagation Rounding Algorithm 6.5. Let $\mu \in (0, 1]$. If $X(k)$ is $(\mu^4/4, L)$-two-step tensorial, then*

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[K_\mu^c\right] \leq \frac{\mu^2}{2}, \tag{2}$$

and

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[P_\mu^c\right] \leq \frac{\mu^2}{2}. \tag{3}$$

Proof. We only prove Eq. (2) since the proof of Eq. (3) is similar. Define the random variable $\mathbf{R} := \mathbb{E}_{\mathfrak{a}\sim\Pi_k} \left\|\{\mathbf{Z}'_{\mathfrak{a}}\} - \{\mathbf{Z}'_{a_1}\} \cdots \{\mathbf{Z}'_{a_k}\}\right\|_1$ on the sample space $\Omega = \{(m, S, \sigma)\}$. From our $(\mu^4/4, L)$-two-step tensorial assumption we have

$$\mathbb{E}_\Omega[\mathbf{R}] \leq \frac{\mu^4}{4}.$$

Now, we can conclude

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[K_\mu^c\right] = \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[\mathbf{R} \geq \frac{\mu^2}{2}\right] \leq \frac{\mu^2}{2},$$

using Markov's inequality. ∎

6.6 Further Building Blocks and Analysis

Before we delve into further phases of the list decoding framework, we introduce some notation for the list of codewords we want to retrieve.

Definition 6.13 (Code list). *Given $\tilde{y} \in \{\pm 1\}^{X(k)}$ and a code $\mathcal{C}$ on $X(k)$ with relative distance at least $1/2 - \varepsilon$, we define the list $\mathcal{L}(\tilde{y}, \mathcal{C})$ as*

$$\mathcal{L}(\tilde{y}, \mathcal{C}) := \left\{ y \in \mathcal{C} \mid \Delta(y, \tilde{y}) \leq \frac{1}{2} - \sqrt{\varepsilon} \right\}.$$

Under these assumptions the Johnson bound establishes that the list size is constant whenever $\varepsilon > 0$ is constant.

Remark 6.14. *The Johnson bound [GRS19] guarantees that*

$$|\mathcal{L}(\tilde{y}, \mathcal{C})| \leq \frac{1}{2 \cdot \varepsilon}$$

provided the relative distance of $\mathcal{C}$ is at least $1/2 - \varepsilon$.

In the case of lifted codes, it is more appropriate to consider a list of pairs $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ defined as follows.

Definition 6.15 (Coupled code list). *Given $\tilde{y} \in \{\pm 1\}^{X(k)}$ and a lifted code $\mathcal{C}_k$ on $X(k)$ with relative distance at least $1/2 - \varepsilon$, we define the coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ as*

$$\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k) := \left\{ (z, \mathrm{lift}(z)) \mid z \in \mathcal{C}_1 \quad \text{and} \quad \Delta(\mathrm{lift}(z), \tilde{y}) \leq \frac{1}{2} - \sqrt{\varepsilon} \right\}.$$

Recovering this list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ is the main goal of this section. This task will be accomplished by Algorithm 6.16 stated below whose building blocks and analysis we develop in this section.

Algorithm 6.16 (List Decoding Algorithm)**.**
Input *A word $\tilde{y} \in \{\pm 1\}^{X(k)}$ which is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k = \mathrm{lift}(\mathcal{C}_1)$.*
Output *Coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$.*

1. *Solve the List Decoding Program with η-accuracy, obtaining $\mathbf{Z}$, where $\eta = \varepsilon^8/2^{22}$*
2. *Let $\mathcal{M}$ be the output of the Cover Retrieval Algorithm 6.29 on $\mathbf{Z}$*
3. *Let $\mathcal{L}'$ be the output of the Cover Purification Algorithm 6.36 on $\mathcal{M}$*
4. *Let $\mathcal{L}'' = \{(z, y) \in \mathcal{L}' \mid \Delta(\tilde{y}, y) \leq 1/2 - \sqrt{\varepsilon}\}$*
5. *Output $\mathcal{L}''$*

As shown in Fig. 1 of Section 3, the first step is to solve the List Decoding Program which results in a pseudo-expectation "covering" the list $\mathcal{L}(\tilde{y}, \mathcal{C})$ as we will make precise. A precursor property to covering and some considerations about SOS rounding are treated in Section 6.6.1. Next, the formal definition of cover is presented in Section 6.6.2 and we have all the elements to present the Cover Retrieval Algorithm 6.29 with its correctness in Section 6.6.3. Then, we use the *robustness* properties of the lifting to purify the cover in Section 6.6.4. Finally, in Section 6.6.5, we assemble the building blocks and prove the main technical result, Theorem 6.17, whose proof follows easily once the properties of the building blocks are in place.

Note that Theorem 6.17 embodies an abstract list decoding framework which relies only on the *robustness* and *tensorial* properties of the lifting. We provide a concrete instantiation of the framework to the direct sum lifting on HDXs in Section 7 and to the direct sum lifting on expander walks in Section 9.2.

Theorem 6.17 (List Decoding Theorem)**.** *Suppose that* lift *is a $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust $(\varepsilon^8/2^{22}, L)$-two-step tensorial lifting from $\mathcal{C}_1$ to $\mathcal{C}_k$ which is either*

- *linear and a $(1/2 + \varepsilon_0, 2 \cdot \varepsilon)$-parity sampler; or*
- *$(1/4 - \varepsilon_0, 1/2 - \varepsilon/2)$-robust and odd.*

Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ be $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$. Then w.v.h.p. the List Decoding Algorithm 6.16 returns the coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. Furthermore, the running time is

$$n^{O(L+k)} \left(\mathrm{polylog}(\varepsilon^{-1}) + f(n) \right),$$

where $n = |X(1)|$ and $f(n)$ is the running time of a unique decoding algorithm of $\mathcal{C}_1$.

Remark 6.18. *Regarding Theorem 6.17, we stress that although the lifting is $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust and we can perform list decoding at least up to distance $1/2 - \sqrt{\varepsilon}$, our framework does not recover the Johnson bound. The issue is that our framework requires one of the additional amplification guarantees of Theorem 6.17, which both make the distance of $\mathcal{C}_k$ become $1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)} > 1/2 - \varepsilon$. Efficiently recovering the Johnson bound remains an interesting open problem.*

We observe that the algorithms themselves used in this framework are quite simple (although their analyses might not be). Moreover, the tasks of cover retrieval and purification are reasonably straightforward. However, Section 6.6.1 combines *tensorial* properties of the lifting with properties of Ψ, requiring a substantial analysis. The list decoding framework is divided into stages to make it modular so that key properties are isolated and their associated functionality can be presented in a simple manner. Most of the power of this framework comes from the combination of these blocks and the concrete expanding objects capable of instantiating it.

6.6.1 SOS Rounding and Recoverability

We show that if a slice (m, S, σ) "captures" an intended solution $y \in \{\pm 1\}^{X(k)}$ (this notion is made precise in the assumptions of Lemma 6.20), then we can retrieve a $z \in \{\pm 1\}^{X(1)}$ such that $\mathrm{lift}(z)$ has some agreement with y. This agreement is somewhat weak, but combined with the robustness of the lifting, it will be enough for our purposes. In this subsection, we first explore how to recover such words within a slice, which can be seen as local rounding in the slice. Next, we establish sufficient conditions for an intended solution to be recoverable, now not restricted to a given slice but rather with respect to the full pseudo-expectation. Finally, we use all the tools developed so far to show that by minimizing Ψ in a two-step tensorial structure we end up with a pseudo-expectation in which all intended solutions are recoverable. The interplay between weak agreement and robustness of the lifting is addressed in Section 6.6.4.

We will be working with two-step tensorial structures where the following product distribution associated to a slice naturally appears.

Definition 6.19 (Product Distribution on a Slice)**.** *We define $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}$ to be the product distribution on the marginals $\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}_{i\in X(1)}$, i.e., $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\} \coloneqq \prod_{i\in X(1)}\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}$.*

Under appropriate conditions, Lemma 6.20 shows how to round the pseudo-expectation in a slice.

Lemma 6.20 (From fractional to integral in a slice)**.** *Let $(m, S, \sigma) \in \Omega$ be a slice. Suppose*

$$\mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \|\{\mathbf{Z}_{\mathfrak{a}}|\mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{a_1}|\mathbf{Z}_S = \sigma\}\cdots\{\mathbf{Z}_{a_k}|\mathbf{Z}_S = \sigma\}\|_1 \le \mu, \tag{4}$$

and

$$\mathop{\mathbb{E}}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2} \|\{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{\mathfrak{s}}|\mathbf{Z}_S = \sigma\}\{\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\}\|_1 \le \mu. \tag{5}$$

For $\beta \in (0,1)$, if $\mu \le \beta\cdot\kappa^2/6$ and $y \in \{\pm 1\}^{X(k)}$ is such that

$$\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\widetilde{\mathbb{E}}_{|S,\sigma}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] \ge \kappa^2,$$

then

$$\mathop{\mathbb{P}}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[|\mathbb{E}_{\mathfrak{s}\sim\Pi_k} y_{\mathfrak{s}}\cdot\mathrm{lift}(z)_{\mathfrak{s}}| \ge \sqrt{1-\beta}\cdot\kappa\right] \ge \frac{\beta\cdot\kappa^2}{4}. \tag{6}$$

Proof. Let $\mu_{\mathfrak{s},\mathfrak{t}} := \|\{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\} - \prod_{i\in\mathfrak{s}}\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}\prod_{i\in\mathfrak{t}}\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}\|_1$. Using triangle inequality and simplifying, we get

$$\begin{aligned}\mu_{\mathfrak{s},\mathfrak{t}} \leq &\|\{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{\mathfrak{s}}|\mathbf{Z}_S = \sigma\}\{\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\}\|_1 \\ &+ \left\|\{\mathbf{Z}_{\mathfrak{s}}|\mathbf{Z}_S = \sigma\} - \prod_{i\in\mathfrak{s}}\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}\right\|_1 + \left\|\{\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\} - \prod_{i\in\mathfrak{t}}\{\mathbf{Z}_i|\mathbf{Z}_S = \sigma\}\right\|_1.\end{aligned}$$

From our assumptions Eq. (4) and Eq. (5), it follows that $\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\mu_{\mathfrak{s},\mathfrak{t}} \leq 3\cdot\mu$. Using the fact that $|y_{\mathfrak{s}}y_{\mathfrak{t}}| = 1$ and Hölder's inequality, we get

$$\begin{aligned}\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\mathbb{E}_{\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] &\geq \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\widetilde{\mathbb{E}}_{|S,\sigma}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] - \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\mu_{\mathfrak{s},\mathfrak{t}} \\ &\geq \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\widetilde{\mathbb{E}}_{|S,\sigma}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] - 3\cdot\mu \geq \left(1-\frac{\beta}{2}\right)\cdot\kappa^2.\end{aligned}$$

Alternatively,

$$\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\mathbb{E}_{\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] = \mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left(\mathbb{E}_{\mathfrak{s}\sim\Pi_k}y_{\mathfrak{s}}\cdot\mathrm{lift}(z)_{\mathfrak{s}}\right)^2 \geq \left(1-\frac{\beta}{2}\right)\cdot\kappa^2.$$

Define the random variable $\mathbf{R} := \left(\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_{\mathfrak{s}}\cdot\mathrm{lift}(z)_{\mathfrak{s}}]\right)^2$. Using Fact A.1 with approximation parameter $\beta/2$, we get

$$\mathbb{E}\left[\mathbf{R}\right] \geq \left(1-\frac{\beta}{2}\right)\cdot\kappa^2 \Rightarrow \Pr\left[\mathbf{R} \geq (1-\beta)\cdot\kappa^2\right] \geq \frac{\beta\cdot\kappa^2}{4},$$

from which Eq. (6) readily follows. ∎

To formalize the notion of a word being recoverable with respect to the full pseudo-expectation rather than in a given slice we will need two additional events. The first event captures correlation as follows.

Definition 6.21 (y-Correlated Event)**.** *Let $\kappa \in (0,1]$ and $y \in \{\pm1\}^{X(k)}$. We define the event $C_\kappa(y)$ as*

$$C_\kappa(y) := \left\{(m,S,\sigma) \in \Omega \mid \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2}\widetilde{\mathbb{E}}_{|S,\sigma}\left[y_{\mathfrak{s}}y_{\mathfrak{t}}\mathbf{Y}_{\mathfrak{s}}\mathbf{Y}_{\mathfrak{t}}\right] \geq \kappa^2\right\}.$$

The second event is a restriction of the first where we also require the slice to satisfy the two-step tensorial condition from Definition 6.9.

Definition 6.22 (y-Recoverable Event)**.** *Let $\kappa,\mu \in (0,1]$ and $y \in \{\pm1\}^{X(k)}$. We define the event $R_{\kappa,\mu}(y)$ as*

$$R_{\kappa,\mu}(y) := K_\mu \cap P_\mu \cap C_\kappa(y).$$

Lemma 6.20 motivates the following "recoverability" condition.

Definition 6.23 (Recoverable Word)**.** *Let $\kappa,\mu \in (0,1]$ and $y \in \{\pm1\}^{X(k)}$. We say that y is (κ,μ)-recoverable provided*

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[R_{\kappa,\mu}(y)\right] > 0.$$

One of the central results in our framework is the following "recoverability" lemma. It embodies the power SOS brings to our framework.

Lemma 6.24 (Recoverability lemma)**.** *Let $\mathcal{C}_k$ be a lifted code on $X(\leq k)$ with $X(1) = [n]$ and distance at least $1/2 - \varepsilon$. Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ be a word promised to be $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$ and let $\mathcal{L} = \mathcal{L}(\tilde{y}, \mathcal{C}_k)$ be its code list.*

Let $\theta \in (0, 1]$ be arbitrary and set $\mu = \kappa \cdot \theta/2$ and $\kappa = (4 - \theta) \cdot \varepsilon$. Suppose $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ is an $(L + 2k)$-local PSD ensemble which is a solution to the List Decoding Program with objective value Ψ within η additive value from the optimum where $0 \leq \eta \leq \theta^2 \cdot \varepsilon^4$.

If $X(k)$ is $(\mu^4/4, L)$-two-step tensorial, then every $y \in \mathcal{L}$ is (κ, μ)-recoverable. In particular, for every $\theta \in (0, 1)$ and under the preceding assumptions, we have that every $y \in \mathcal{L}$ is $((4 - \theta) \cdot \varepsilon, \theta)$-recoverable.

Proof. First observe that since $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$ the List Decoding Program is feasible and so the solution $\mathbf{Z}$ is well defined. Towards a contradiction with the η-optimality of the SOS solution $\mathbf{Z}$, suppose there exists a word $y \in \mathcal{L}$ that is not (κ, μ)-recoverable. Let $z \in \{\pm 1\}^{X(1)}$ be such that $y = \mathrm{lift}(z)$. Then

$$1 \;=\; \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[R_{\kappa,\mu}(y)^c\right] \;\leq\; \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[K_\mu^c\right] + \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[P_\mu^c\right] + \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[C_\kappa(y)^c\right].$$

Using Claim 6.12, we get

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[C_\kappa(y)^c\right] \;\geq\; 1 - \mu^2. \tag{7}$$

Since $\widetilde{\mathbb{E}}$ is a valid solution to the List Decoding Program, Lemma 6.3 implies the lower bound

$$\Psi\left(\{\mathbf{Y}_\mathfrak{s}\}_{\mathfrak{s}\in X(k)}\right) \;\geq\; 16 \cdot \varepsilon^2. \tag{8}$$

By definition, for $(m, S, \sigma) \in C_\kappa(y)^c$ we have

$$\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2} \widetilde{\mathbb{E}}_{|S,\sigma}\left[y_\mathfrak{s} y_\mathfrak{t} \mathbf{Y}_\mathfrak{s} \mathbf{Y}_\mathfrak{t}\right] \leq \kappa^2,$$

implying

$$\widetilde{\mathbb{E}}\left[\left(\mathbb{E}_{\mathfrak{s}\sim\Pi_k} y_\mathfrak{s} \cdot \mathbf{Y}_\mathfrak{s}\right)^2\right] \leq \mathbb{E}_{m,S,\sigma} \widetilde{\mathbb{E}}_{|S,\sigma}\left[\left(\mathbb{E}_{\mathfrak{s}\sim\Pi_k} y_\mathfrak{s} \cdot \mathbf{Y}_\mathfrak{s}\right)^2 \cdot \mathbf{1}_{C_\kappa^c(y)}\right] + \mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[C_\kappa(y)\right] \leq \kappa^2 + \mu^2.$$

Let $\widetilde{\mathbb{E}}$ be the pseudo-expectation of the ground set ensemble $\mathbf{Z}$ and let $\mathbb{E}'$ be the expectation on the delta distribution δ_z. Note that the pseudo-expectation obtained from $\mathbb{E}'$ is a valid solution to the List Decoding Program. Since

$$\kappa^2 + \mu^2 \leq \left(1 + \frac{\theta^2}{4}\right) \cdot \kappa^2 = \left(1 + \frac{\theta^2}{4}\right) \cdot (4 - \theta)^2 \cdot \varepsilon^2 \leq (16 - 2 \cdot \theta) \cdot \varepsilon^2,$$

and $\theta \geq 0$, Lemma 6.4 gives that there is a convex combination of $\widetilde{\mathbb{E}}$ and $\mathbb{E}'$ such that the new Ψ, denoted Ψ', can be bounded as

$$\Psi' \;\leq\; \Psi - \frac{\left(\Psi - (\kappa^2 + \mu^2)\right)^2}{2} \leq \Psi - 2 \cdot \theta^2 \cdot \varepsilon^4,$$

contradicting the η-optimality of the SOS solution $\mathbf{Z}$ since $\eta \leq \theta^2 \cdot \varepsilon^4$. ∎

6.6.2 Coupled Pairs, Coupled Lists, and Covers

The List Decoding Program minimizing Ψ was instrumental to ensure that every $y' \in \mathcal{L}(\tilde{y}, \mathcal{C}_k)$ is recoverable in the sense of the conclusion of Lemma 6.24. Unfortunately, this guarantee is somewhat weak, namely, associated to every $y' \in \mathcal{L}(\tilde{y}, \mathcal{C}_k)$ there is a slice (m, S, σ) from which we can sample y (our approximation of y') satisfying

$$|\mathbb{E}_{\mathfrak{s} \sim \Pi_k} y_{\mathfrak{s}} \cdot y'_{\mathfrak{s}}| > C \cdot \varepsilon, \tag{9}$$

where C is a constant strictly smaller than 4. A priori this seems insufficient for our list decoding task. However, there are two properties which will help us with list decoding. The first is that SOS finds not only y but also $z \in \{\pm 1\}^{X(1)}$ such that $y = \text{lift}(z)$. The second property is that the lifting is robust: even the weak agreement given by Eq. (9) translates into a much stronger agreement in the ground set between z and $z' \in \mathcal{C}_1$ where $y' = \text{lift}(z')$. This stronger agreement on the ground set can be used to ensure that z (or $-z$) lies inside the unique decoding ball of z' in the base code $\mathcal{C}_1$.

To study this coupling phenomenon between words in the lifted space $\{\pm 1\}^{X(k)}$ and on the ground space $\{\pm 1\}^{X(1)}$ we introduce some terminology. The most fundamental one is a coupled pair.

Definition 6.25 (Coupled Pair)**.** *Let $z \in \{\pm 1\}^{X(1)}$ and $y \in \{\pm 1\}^{X(k)}$. We say that (z, y) is a coupled pair with respect to a lift function* lift *provided $y = \text{lift}(z)$.*

Remark 6.26. *If the function* lift *is clear in the context, we may assume that the coupled pair is with respect to this function.*

Coupled pairs can be combined in a list.

Definition 6.27 (Coupled List)**.** *We say that a list $\mathcal{M} = \{(z^{(1)}, y^{(1)}), \ldots, (z^{(h)}, y^{(h)})\}$ is coupled with respect to lift function* lift *provided $(z^{(i)}, y^{(i)})$ is a coupled pair for every i in $[h]$.*

A coupled list can "cover" a list of words in the lifted space $\{\pm 1\}^{X(k)}$ as defined next.

Definition 6.28 (Coupled Bias Cover)**.** *Let $\mathcal{M} = \{(z^{(1)}, y^{(1)}), \ldots, (z^{(h)}, y^{(h)})\}$ be a coupled list and $\mathcal{L} \subset \{\pm 1\}^{X(k)}$. We say that $\mathcal{M}$ is a δ-bias cover of $\mathcal{L}$ provided*

$$\left(\forall y' \in \mathcal{L}\right) \left(\exists (z, y) \in \mathcal{M}\right) \left(|\mathbb{E}_{\mathfrak{s} \sim \Pi_k} y'_{\mathfrak{s}} \cdot y_{\mathfrak{s}}| > \delta\right).$$

A δ-bias cover for "small" δ might seem a rather weak property, but as alluded to, when combined with enough robustness of the lifting, it becomes a substantial guarantee enabling list decoding.

6.6.3 Cover Retrieval

When the code list $\mathcal{L}(\tilde{y}, \mathcal{C}_k)$ becomes recoverable in the SOS sense as per Lemma 6.24, we still need to conduct local rounding on the slices to collect a bias cover. Recall that this local rounding is probabilistic (c.f. Lemma 6.20), so we need to repeat this process a few times to boost our success probability[16]. This is accomplished by Algorithm 6.29.

[16] In fact, this process can be derandomized using standard techniques in our instantiations. See Lemma C.1 for detais.

Algorithm 6.29 (Cover Retrieval Algorithm).

Input *An $(L+2k)$-local PSD ensemble $\mathbf{Z}$ which is a $(\theta^2\varepsilon^4)$-optimal solution to the List Decoding Program.*

Output *A 2ε-bias cover $\mathcal{M}$ for $\mathcal{L}(\tilde{y}, \mathcal{C}_k)$.*

1. *Let $\mathcal{M} = \emptyset$*
2. *Let $T = 4 \cdot \ln(|\Omega|) \cdot n/(\beta \cdot \varepsilon^2)$*
3. *For $(m, S, \sigma) \in \Omega$ do*
4. *If $(m, S, \sigma) \in K_\mu \cap P_\mu$ then*
5. *Run Propagation Rounding T times conditioned on (m, S, σ)*
6. *Let $\mathcal{M}|_{m,S,\sigma} = \{(z^{(1)}, y^{(1)}), \ldots, (z^{(T)}, y^{(T)})\}$ be the coupled list*
7. *Set $\mathcal{M} = \mathcal{M} \cup \mathcal{M}|_{m,S,\sigma}$*
8. *Output $\mathcal{M}$.*

The correctness of Algorithm 6.29 follows easily given the properties established so far.

Lemma 6.30 (Cover lemma). *Let $\beta \in (0,1)$. Suppose that* lift *is a $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust $(\beta^4 \cdot \varepsilon^8/2^{18}, L)$-two-step tensorial lifting from $\mathcal{C}_1$ to $\mathcal{C}_k$. Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ be $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$. If $\theta \leq \beta \cdot \varepsilon/2^4$, then w.v.h.p.[17] the Cover Retrieval algorithm 6.29 returns a δ-bias cover $\mathcal{M}$ of the code list $\mathcal{L}(\tilde{y}, \mathcal{C}_k)$ where $\delta = (4-\beta) \cdot \varepsilon$. Furthermore, the running time is at most $n^{O(L+k)}/(\beta \cdot \varepsilon^2)$ where $n = |X(1)|$.*

Proof. Let $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ be an η-optimum solution to the List Decoding Program where $\eta \leq \theta^2 \cdot \varepsilon^4$ and $\theta = \beta \cdot \varepsilon/2^4$. By our $(\beta^4 \cdot \varepsilon^8/2^{18}, L)$-two-step tensorial assumption and our choice of SOS degree for the List Decoding Program, we can apply Lemma 6.24 to conclude that every $y \in \mathcal{L} = \mathcal{L}(\tilde{y}, \mathcal{C}_k)$ is $((4-\theta) \cdot \varepsilon, (4-\theta) \cdot \varepsilon \cdot \theta/2)$-recoverable. Then for $y \in \mathcal{L}$, there exists $(m, S, \sigma) \in \Omega$ such that Lemma 6.20 yields

$$\mathop{\mathbb{P}}_{z \sim \{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}} [|\mathbb{E}_{\mathfrak{s} \sim \Pi_k} y_{\mathfrak{s}} \cdot \mathrm{lift}(z)| \geq (4-\beta) \cdot \varepsilon] \geq \frac{\beta \cdot (4-\theta)^2 \cdot \varepsilon^2}{32} \geq \frac{\beta \cdot \varepsilon^2}{4}.$$

where $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}$ (c.f. Definition 6.19) is the product distribution of the marginal distributions after conditioning the ensemble on slice (m, S, σ). By sampling $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}$ independently T times we obtain $z^{(1)}, \ldots, z^{(T)}$ and thus also the coupled list

$$\mathcal{M}|_{m,S,\sigma} = \{(z^{(1)}, y^{(1)}), \ldots, (z^{(T)}, y^{(T)})\},$$

where $y^{(i)} = \mathrm{lift}(z^{(i)})$. Then

$$\begin{aligned} \mathop{\mathbb{P}}_{z^{(1)},\ldots,z^{(T)} \sim \{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}^{\otimes T}} \left[\forall i \in [T] : \left|\mathbb{E}_{\mathfrak{s} \sim \Pi_k} y_{\mathfrak{s}} \cdot \mathrm{lift}\left(z^{(i)}\right)\right| < (4-\beta) \cdot \varepsilon\right] &\leq \exp\left(-\frac{\beta \cdot \varepsilon^2 \cdot T}{4}\right) \\ &\leq \frac{\exp(-n)}{|\Omega|}, \end{aligned}$$

[17] The abbreviation w.v.h.p. stand for *with very high probability* and means with probability $1 - \exp(-\Theta(n))$.

where the last inequality follows from our choice of T. Then by union bound

$$\mathbb{P}\left[\mathcal{M} \text{ is not a } 2\varepsilon\text{-bias cover of } \mathcal{L}\right] \leq |\mathcal{L}| \cdot \frac{\exp(-n)}{|\Omega|} \leq \exp(-n),$$

concluding the proof. ∎

6.6.4 Cover Purification and Robustness

Now we consider the third and final stage of the list decoding framework. We show how despite the weak guarantee of the bias cover returned by the Cover Retrieval Algorithm 6.29 we can do a further processing to finally obtain the coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ provided the lifting admits some *robustness* properties. We first develop these properties and later present this process, denoted Cover Purification.

Further Lifting Properties

Given two coupled pairs $(z, y = \text{lift}(z))$ and $(z', y' = \text{lift}(z'))$ (where $z \in \mathcal{C}_1$), we show how weak agreement between y and y' on the lifted space is enough to provide non-trivial guarantees between z and z' as long as the lifting admits appropriate *robustness*.

Claim 6.31 (Coupled unique decoding from distance)**.** *Suppose that* lift *is a* $(1/4 - \varepsilon_0/2, 1/2 - \varepsilon)$*-robust lifting from* $\mathcal{C}_1$ *to* $\mathcal{C}_k$*. Let* (z, y) *and* (z', y') *be coupled pairs. If* $y \in \mathcal{C}_k$ *(equivalently* $z \in \mathcal{C}_1$*) and* $\Delta(y, y') < 1/2 - \varepsilon$*, then* $\Delta(z, z') \leq 1/4 - \varepsilon_0/2$*, i.e.,* z' *is within the unique decoding radius of* z*.*

Proof. Towards a contradiction suppose that $\Delta(z, z) \geq 1/4 - \varepsilon_0/2$. Since the lifting is $(1/4 - \varepsilon_0/2, 1/2 - \varepsilon)$-robust, this implies that $\Delta(y, y') \geq 1/2 - \varepsilon$ contradicting our assumption. ∎

From bias amplification (i.e., parity sampling), we deduce Claim 6.32.

Claim 6.32 (Coupled unique decoding from bias I)**.** *Suppose* lift *is a* $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$*-robust linear lifting from* $\mathcal{C}_1$ *to* $\mathcal{C}_k$ *which is also a* $(1/2 + \varepsilon_0, 2 \cdot \varepsilon)$*-parity sampler. Let* (z, y) *and* (z', y') *be coupled pairs. If* $y \in \mathcal{C}_k$ *(equivalently* $z \in \mathcal{C}_1$*) and* $|\mathbb{E}_{\mathfrak{s} \sim \Pi_k}[y_{\mathfrak{s}} \cdot y'_{\mathfrak{s}}]| > 2 \cdot \varepsilon$*, then*

$$|\mathbb{E}_{i \sim \Pi_1}[z_i \cdot z'_i]| \geq 1/2 + \varepsilon_0,$$

i.e., either z' *or* $-z'$ *is within the unique decoding radius of* z*.*

Proof. The verification follows easily from our assumptions. Towards a contradiction suppose that $|\mathbb{E}_{i \sim \Pi_1}[z_i \cdot z'_i]| < 1/2 + \varepsilon_0$, i.e., the word $z'' = z \cdot z'$ has bias at most $1/2 + \varepsilon_0$. Using the assumption that the lift is linear, we have $\text{lift}(z'') = \text{lift}(z) \cdot \text{lift}(z')$. Since the lifting takes bias $1/2 + \varepsilon_0$ to $2 \cdot \varepsilon$, we have

$$\text{bias}(\text{lift}(z) \cdot \text{lift}(z')) = \text{bias}(\text{lift}(z'')) \leq 2 \cdot \varepsilon,$$

or equivalently $|\mathbb{E}_{\mathfrak{s} \sim \Pi_k}[y_{\mathfrak{s}} \cdot y'_{\mathfrak{s}}]| \leq 2 \cdot \varepsilon$ contradicting our assumption. ∎

If the lifting function is odd, then we obtain Claim 6.33.

Claim 6.33 (Coupled unique decoding from bias II)**.** *Suppose* lift *is a* $(1/4 - \varepsilon_0/2, 1/2 - \varepsilon)$*-robust lifting from* $\mathcal{C}_1$ *to* $\mathcal{C}_k$ *which is odd, i.e.,* $\mathrm{lift}(-z) = -\mathrm{lift}(z)$*. Let* (z, y) *and* (z', y') *be coupled pairs. If* $y \in \mathcal{C}_k$ *(equivalently* $z \in \mathcal{C}_1$*) and* $|\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_\mathfrak{s} \cdot y'_\mathfrak{s}]| > 2 \cdot \varepsilon$*, then either* z' *or* $-z'$ *is within the unique decoding radius of* z*.*

Proof. Since $|\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_\mathfrak{s} \cdot y'_\mathfrak{s}]| > 2 \cdot \varepsilon$ and the lifting is odd, either

$$\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_\mathfrak{s} \cdot \mathrm{lift}(z')_\mathfrak{s}] > 2 \cdot \varepsilon,$$

or

$$\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_\mathfrak{s} \cdot \mathrm{lift}(-z')_\mathfrak{s}] = \mathbb{E}_{\mathfrak{s}\sim\Pi_k}\left[-y_\mathfrak{s} \cdot \mathrm{lift}(z')_\mathfrak{s}\right] > 2 \cdot \varepsilon.$$

Then either $\Delta(y, \mathrm{lift}(z')) \leq 1/2 - \varepsilon$ or $\Delta(y, \mathrm{lift}(-z')) \leq 1/2 - \varepsilon$. Using Claim 6.31 we conclude the proof. ∎

Cover Purification

A δ-bias cover $\mathcal{M}$ of $\mathcal{L}$ for small δ may require further processing in order to actually retrieve $\mathcal{L}$. Provided the lifting is sufficiently robust, trying to unique decode z for $(z, y) \in \mathcal{M}^{\pm}$, where $\mathcal{M}^{\pm}$ is the sign completion as defined next, and then lifting the decoded word yields a new coupled list that contains $\mathcal{L}$. This process is referred to as cover purification and its formalization is the object of this section.

Definition 6.34 (Sign Completion)**.** *Let* $\mathcal{M}$ *be coupled list. We say that* $\mathcal{M}^{\pm}$ *defined as*

$$\mathcal{M}^{\pm} := \{(z, \mathrm{lift}(z)), (-z, \mathrm{lift}(-z)) \mid (z, y) \in \mathcal{M}\},$$

is the sign completion of $\mathcal{M}$*.*

The correctness of the cover purification process is established next.

Lemma 6.35 (Purification lemma)**.** *Suppose* lift *is a* $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$*-robust lifting from* $\mathcal{C}_1$ *to* $\mathcal{C}_k$ *which is either*

- *linear and a* $(1/2 + \varepsilon_0, 2 \cdot \varepsilon)$*-parity sampler; or*
- $(1/4 - \varepsilon_0/2)$*-robust and odd.*

Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ *be* $(1/2 - \sqrt{\varepsilon})$*-close to* $\mathcal{C}_k$ *and* $\mathcal{L} = \mathcal{L}(\tilde{y}, \mathcal{C}_k)$ *be its code list. If* $\mathcal{M} = \{(z^{(i)}, y^{(i)}) | i \in [h]\}$ *is a* 2ε*-bias cover of* $\mathcal{L}$*, then*

$$\mathcal{L} \subseteq \left\{\mathrm{lift}(z) \mid z \in \mathrm{Dec}_{\mathcal{C}_1}\left(\mathrm{P}_{\mathbf{1}}\left(\mathcal{M}^{\pm}\right)\right)\right\} =: \mathcal{L}',$$

where $\mathrm{P}_{\mathbf{1}}$ *is the projection on the first coordinate and* $\mathrm{Dec}_{\mathcal{C}_1}$ *is a unique decoder for* $\mathcal{C}_1$*. Furthermore,* $\mathcal{L}'$ *can be computed in time* $O(|\mathcal{M}| \cdot f(n))$ *where* $f(n)$ *is the running time of a unique decoding algorithm of* $\mathcal{C}_1$*.*

Proof. Let $y \in \mathcal{L}$. By the 2ε-cover property, there exists a coupled pair $(z', y') \in \mathcal{M}$ satisfying $|\mathbb{E}_{\mathfrak{s}\sim\Pi_k}[y_\mathfrak{s} \cdot y'_\mathfrak{s}]| > 2 \cdot \varepsilon$. Combining this bound with the appropriate robustness assumptions, Claim 6.32 or Claim 6.33 yields that either z' or $-z'$ can be uniquely decoded in $\mathcal{C}_1$. Then

$$y \in \left\{\mathrm{lift}(z) \mid z \in \mathrm{Dec}_{\mathcal{C}_1}\left(\mathrm{P}_{\mathbf{1}}\left(\mathcal{M}^{\pm}\right)\right)\right\}.$$

Finally, observe that computing $\mathcal{L}'$ with the claimed running time is straightforward. ∎

Algorithmically, cover purification works by running the unique decoding algorithm of $\mathcal{C}_1$ on every element of the sign completion $\mathcal{M}^{\pm}$, described below in Algorithm 6.36.

Algorithm 6.36 (Cover Purification Algorithm)**.**
Input *A 2ε-bias cover $\mathcal{M}$ for $\mathcal{L}(\tilde{y}, \mathcal{C}_k)$.*
Output *Coupled List $\mathcal{L}'$ s.t. $\mathrm{P}_2(\mathcal{L}') \supseteq \mathcal{L}(\tilde{y}, \mathcal{C}_k)$.*

1. *Let $\mathcal{L}' = \emptyset$*
2. *For $(z', y') \in \mathcal{M}^{\pm}$ do*
3. *If z' is uniquely decodable in $\mathcal{C}_1$ then*
4. *Let $z = \mathrm{UniqueDecode}_{\mathcal{C}_1}(z')$*
5. *Let $y = \mathrm{lift}(z)$*
6. *Set $\mathcal{L}' = \mathcal{L}' \cup \{(z, y)\}$*
7. *Output $\mathcal{L}'$.*

6.6.5 Correctness of the List Decoding Algorithm

The building blocks developed so far are assembled to form the final list decoding algorithm (Algorithm 6.16), which is restated below for convenience.

Algorithm 6.37 (List Decoding Algorithm)**.**
Input *A word $\tilde{y} \in \{\pm 1\}^{X(k)}$ $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k = \mathrm{lift}(\mathcal{C}_1)$*
Output *Coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$.*

1. *Solve the List Decoding Program with η-accuracy obtaining $\mathbf{Z}$ where $\eta = \varepsilon^8/2^{22}$*
2. *Let $\mathcal{M}$ be the output of the Cover Retrieval Algorithm 6.29 on $\mathbf{Z}$*
3. *Let $\mathcal{L}'$ be the output of the Cover Purification Algorithm 6.36 on $\mathcal{M}$*
4. *Let $\mathcal{L}'' = \{(z, y) \in \mathcal{L}' \mid \Delta(\tilde{y}, y) \leq 1/2 - \sqrt{\varepsilon}\}$*
5. *Output $\mathcal{L}''$.*

We are ready to prove the main theorem of the abstract list decoding framework which follows easily from the properties developed so far.

Theorem 6.38 (List Decoding Theorem (Restatement of Theorem 6.17))**.** *Suppose that* lift *is a $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust $(\varepsilon^8/2^{22}, L)$-two-step tensorial lifting from $\mathcal{C}_1$ to $\mathcal{C}_k$ which is either*

- *linear and a $(1/2 + \varepsilon_0, 2 \cdot \varepsilon)$-parity sampler; or*
- *$(1/4 - \varepsilon_0, 1/2 - \varepsilon/2)$-robust and odd.*

Let $\tilde{y} \in \{\pm 1\}^{X(k)}$ be $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$. Then w.v.h.p. the List Decoding Algorithm 6.16 returns the coupled code list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. Furthermore, the running time is

$$n^{O(L+k)} \left(\mathrm{polylog}(\varepsilon^{-1}) + f(n) \right),$$

where $n = |X(1)|$ and $f(n)$ is the running time of a unique decoding algorithm of $\mathcal{C}_1$.

Proof. Under the assumptions of the theorem, Lemma 6.30 establishes that the Cover Retrieval Algorithm 6.29 returns w.v.h.p. a 2ε-bias cover. Then, Lemma 6.35 states that providing this 2ε-bias cover as input to the Cover Purification Algorithm 6.36 yields a coupled list containing the code list $\mathcal{L}(\widetilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. Finally, the last step in Algorithm 6.16 ensures the output is precisely $\mathcal{L}(\widetilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. ∎

7 Instantiation I: Direct Sum on HDXs

We instantiate the list decoding framework to the direct sum lifting on HDXs obtaining Theorem 7.1, which is the main result in this section. For this instantiation we need to establish that HDXs are two-step tensorial which will be done in Section 7.1.

Theorem 7.1 (Direct Sum Lifting on HDX)**.** *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. There exist universal constants $c, C > 0$ such that for any γ-HDX $X(\leq d)$ on ground set $X(1) = [n]$ and Π_1 uniform, if*

$$\gamma \leq (\log(1/\varepsilon))^{-C \cdot \log(1/\varepsilon)} \quad \text{and} \quad d \geq c \cdot \frac{(\log(1/\varepsilon))^2}{\varepsilon},$$

then the following holds:

For every binary code $\mathcal{C}_1$ with $\Delta(\mathcal{C}_1) \geq 1/2 - \varepsilon_0$ on $X(1) = [n]$, there exists a binary lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1))$ with $\Delta(\mathcal{C}_k) \geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ on $X(k)$ where $k = O(\log(1/\varepsilon))$, φ is an explicit linear projection, and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time*
$$n^{\varepsilon^{-O(1)}} \cdot f(n),$$
where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.
- *[Rate] The rate r_k of $\mathcal{C}_k$ satisfies $r_k = r_1 \cdot |X(1)| \,/\, |X(k)|$ where r_1 is the relative rate of $\mathcal{C}_1$.*
- *[Linearity] If $\mathcal{C}_1$ is linear, then φ is the identity and $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ is linear.*

In particular, invoking Theorem 7.1 on HDXs extracted from Ramanujan complexes (as in Lemma 4.5), we obtain Corollary 7.2.

Corollary 7.2. *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. There is an infinite sequence of HDXs $X_1, X_2, \ldots$ on ground sets of size $n_1, n_2, \ldots$ such that the following holds:*

For every sequence of binary codes $\mathcal{C}_1^{(i)}$ on $[n_i]$ with rate and distance uniformly bounded by r_1 and $(1/2 - \varepsilon_0)$ respectively, there exists a sequence of binary lifted codes $\mathcal{C}_k^{(i)} = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1^{(i)}))$ on a collection $X_i(k)$ with $\Delta(\mathcal{C}_k^{(i)}) \geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ where φ is an explicit linear projection and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time $n^{\varepsilon^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of a unique decoding algorithm of $\mathcal{C}_1$.*

- *[Explicit Construction] The collection $X_i(k)$ is part of an explicit γ-HDX $X_i(\leq d)$ where $k = O(\log(1/\varepsilon))$, $d = O\left((\log(1/\varepsilon))^2/\varepsilon\right)$, and $\gamma = (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$.*

- *[Rate] The rate $r_k^{(i)}$ of $\mathcal{C}_k^{(i)}$ satisfies $r_k^{(i)} \geq r_1 \cdot \exp\left(-(\log(1/\varepsilon))^{O(\log(1/\varepsilon))}\right)$.*

- *[Linearity] If $\mathcal{C}_1^{(i)}$ is linear, then φ is the identity and $\mathcal{C}_k^{(i)} = \mathrm{dsum}_{X(k)}(\mathcal{C}_1^{(i)})$ is linear.*

Proof. Efficient list decoding and linearity follow directly from Theorem 7.1, and the parameters of the explicit construction match the requirements of the theorem. The only thing left to do is to calculate the rate. Since the lifting $\mathrm{dsum}_{X_i(k)}$ needs to be a $(2\varepsilon_0, 2\varepsilon)$-parity sampler to achieve the promised distance, by Lemma 4.11 the rate $r_k^{(i)}$ of $\mathcal{C}_k^{(i)}$ satisfies

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

Since $\gamma = (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$, this reduces to

$$r_k^{(i)} \geq r_1 \cdot (\log(1/\varepsilon))^{-O((\log(1/\varepsilon))^5/(\varepsilon^2\cdot\gamma))} = r_1 \cdot \exp(1/\gamma^{O(1)}) = r_1 \cdot \exp\left(-(\log(1/\varepsilon))^{O(\log(1/\varepsilon))}\right).$$

■

7.1 HDXs are Two-Step Tensorial

Theorem 7.3 proven in [AJT19] establishes that HDXs of appropriate expansion parameter are tensorial objects for constant $L = O_{k,q,\mu}(1)$.

Theorem 7.3 (HDXs are Tensorial)**.** *There exist some universal constants $c' \geq 0$ and $C' \geq 0$ satisfying the following: If $L \geq c' \cdot (q^k \cdot k^5/\mu^4)$, $\mathrm{Supp}(\mathbf{Z}_j) \leq q$ for all $j \in [n]$, and X is a γ-HDX for $\gamma \leq C' \cdot \mu^4/(k^{8+k} \cdot 2^{6k} \cdot q^{2k})$ and size $\geq k$, then $X(k)$ endowed with a distribution Π_k is (μ, L)-tensorial.*

The next result shows that HDXs are also two-step tensorial objects with the same parameters as above.

Lemma 7.4 (HDXs are two-step tensorial)**.** *There exist some universal constants $c' \geq 0$ and $C' \geq 0$ satisfying the following: If $L \geq c' \cdot (q^k \cdot k^5/\mu^4)$, $\mathrm{Supp}(\mathbf{Z}_j) \leq q$ for all $j \in [n]$, and X is a γ-HDX for $\gamma \leq C' \cdot \mu^4/(k^{8+k} \cdot 2^{6k} \cdot q^{2k})$ and size $\geq k$, then $X(k)$ is (μ, L)-two-step tensorial.*

Proof. Under our assumptions the (μ, L)-tensorial property follows from Theorem 7.3 (this is the only place where the assumption on γ is used), so we only need to show

$$\mathop{\mathbb{E}}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k} \left\| \{\mathbf{Z}'_{\mathfrak{s}}\mathbf{Z}'_{\mathfrak{t}}\} - \{\mathbf{Z}'_{\mathfrak{s}}\}\{\mathbf{Z}'_{\mathfrak{t}}\} \right\|_1 \leq \mu,$$

which can be proven by adapting a potential argument technique from [BRS11]. First, set the potential

$$\Phi_m = \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\sigma\sim\{\mathbf{Z}_S\}} \mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_k} \mathrm{Var}\left[\mathbf{Z}_{\mathfrak{s}} \mid \mathbf{Z}_S = \sigma\right], \tag{10}$$

and consider the error term

$$\mu_m := \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\sigma\sim\{\mathbf{Z}_S\}} D(S,\sigma), \tag{11}$$

where $D(S,\sigma) := \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k}[\|\{\mathbf{Z}_\mathfrak{s}\mathbf{Z}_\mathfrak{t} \mid \mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_\mathfrak{s}|\mathbf{Z}_S = \sigma\}\{\mathbf{Z}_\mathfrak{t}|\mathbf{Z}_S = \sigma\}\|_1]$. If $\mu_m \geq \mu/2$, then

$$\mathop{\mathbb{P}}_{S\sim\Pi_k^m,\sigma\sim\{\mathbf{Z}_S\}}[D(S,\sigma) \geq \mu_m/2] \geq \frac{\mu}{4}.$$

Let $G = (V = X(k), E)$ be the weighted graph where $E = \{\{\mathfrak{s},\mathfrak{t}\} \mid \mathfrak{s},\mathfrak{t} \in X(k)\}$ and each edge $\{\mathfrak{s},\mathfrak{t}\}$ receives weight $\Pi_k(\mathfrak{s}) \cdot \Pi_k(\mathfrak{t})$. Local correlation (expectation over the edges) on this graph G is the same as to global correlation (expectation over two independent copies of vertices). Then, we obtain [18]

$$\Phi_m - \Phi_{m+1} \geq \mathop{\mathbb{P}}_{S\sim\Pi_k^m,\sigma\sim\{\mathbf{Z}_S\}}[D(S,\sigma) \geq \mu_m/2] \cdot \frac{\mu^2}{2q^{2k}}.$$

Since $1 \geq \Phi_1 \geq \cdots \geq \Phi_{L/k} \geq 0$, there can be at most $8q^{2k}/\mu^3$ indices $m \in [L/k]$ such that $\mu_m \geq \mu/2$. In particular, since the total number of indices is L/k, we have

$$\mathop{\mathbb{E}}_{m\in[L/k]} \mu_m \leq \frac{\mu}{2} + \frac{k}{L} \cdot \frac{8q^{2k}}{\mu^3}.$$

Our choice of L is more than enough to ensure $\mathbb{E}_{m\in[L/k]}[\mu_m] \leq \mu$. ∎

7.2 Instantiation to Linear Base Codes

First, we instantiate the list decoding framework to the seemingly simpler case of binary *linear* base codes in Lemma 7.5. As we show later, with a simple observation we can essentially use the proof of Lemma 7.5 to obtain Theorem 7.1 for general codes.

Lemma 7.5 (Direct sum lifting of linear biased codes)**.** *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0,\varepsilon_0)$. There exist universal constants $c, C > 0$ such that for any γ-HDX $X(\leq d)$ on ground set $X(1) = [n]$ and Π_1 uniform, if*

$$\gamma \leq \log(1/\varepsilon)^{-C\cdot(\log(1/\varepsilon))} \quad \text{and} \quad d \geq c \cdot \frac{(\log(1/\varepsilon))^2}{\varepsilon},$$

then the following holds:

For every binary $2\varepsilon_0$-biased linear code $\mathcal{C}_1$ on $X(1) = [n]$, there exists a 2ε-biased binary lifted linear code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ on $X(k)$ where $k = O(\log(1/\varepsilon))$ and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time*

  $$n^{\varepsilon^{-O(1)}} \cdot f(n),$$

  where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.

- *[Rate] The rate [19] r_k of $\mathcal{C}_k$ satisfies $r_k = r_1 \cdot |X(1)| \,/\, |X(k)|$ where r_1 is the relative rate of $\mathcal{C}_1$.*

- *[Linear] The lifted code $\mathcal{C}_k$ is linear.*

[18] See [AJT19] or [BRS11] for the details.

[19] For the rate computation, we assume that $X(k)$ can be expressed as a multi-set such that the uniform distribution on it coincides with Π_k, which is true in the case that Π_k is D-flat.

Proof. We show that under our assumption on the γ-HDX $X(\leq d)$ we obtain sufficient *robustness* and *tensorial* parameters to apply Theorem 6.17. In this application, we will rely on parity sampling for robustness. If $\mathrm{dsum}_{X(k)}$ is a $(2\varepsilon_0, 2\varepsilon)$-parity sampler, using the linearity of $\mathcal{C}_1$ we obtain a lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ which is linear and has bias 2ε; thus the lifting is indeed $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust. If we want to fully rely on parity sampling in Theorem 6.17, the lifting must be a $(\beta_0 = 1/2 + \varepsilon_0, \beta = 2\varepsilon)$-parity sampler, which is more stringent than the first parity sampling requirement. [20] To invoke Lemma 4.10 and obtain this (β_0, β)-parity sampler, we need to choose a parameter θ (where $0 < \theta < (1 - \beta_0)/\beta_0$) and

$$\begin{aligned} k &\geq \log_{(1+\theta)\beta_0}(\beta/3), \\ d &\geq \max\left(\frac{3 \cdot k^2}{\beta}, \frac{6}{\theta^2 \beta_0^2 \beta}\right), \text{ and} \\ \gamma &= O\left(\frac{1}{d^2}\right). \end{aligned}$$

To get a (μ, L)-tensorial HDX, Theorem 7.3 requires

$$L \geq \frac{c' \cdot 2^k \cdot k^5}{\mu^4} \quad \text{and} \quad \gamma \leq \frac{C' \cdot \mu^4}{k^{8+k} \cdot 2^{8k}}.$$

where we used that our alphabet is binary (i.e., $q = 2$) and $c', C' > 0$ are constants. Finally, Theorem 6.17 requires $\mu \leq \varepsilon^8/2^{22}$. The conceptual part of the proof is essentially complete and we are left to compute parameters. Set $\zeta_0 = 3/4 + \varepsilon_0 - \varepsilon_0^2$. We choose $\theta = 1/2 - \varepsilon_0$ which makes $(1+\theta)\beta_0$ equal to ζ_0 (provided $\varepsilon_0 < 1/2$ we have $\zeta_0 < 1$). This choice results in

$$k \geq \lceil \log_{\zeta_0}(2\varepsilon/3) \rceil \quad \text{and} \quad d = O\left(\max\left(\frac{\log_{\zeta_0}(2\varepsilon/3)}{\varepsilon}, \frac{1}{(1/4 - \varepsilon_0^2)^4 \cdot \varepsilon}\right)\right).$$

Combining the parity sampling and tensorial requirements and after some simplification, the expansion γ is constrained as

$$\gamma \leq C'' \cdot \min\left(\frac{\varepsilon^{32}}{k^{8+k} \cdot 2^{8k}}, \frac{\varepsilon^2}{k^4}, \left(1/4 - \varepsilon_0^2\right)^4 \cdot \varepsilon^2\right),$$

where $C'' > 0$ is a constant. We deduce that taking γ as

$$\gamma \leq C'' \cdot \frac{\left(1/4 - \varepsilon_0^2\right)^4 \cdot \varepsilon^{32}}{k^{8+k} \cdot 2^{8k}},$$

is sufficient. Further simplifying the above bound gives

$$\gamma = O\left(\frac{\left(1/4 - \varepsilon_0^2\right)^4 \cdot \varepsilon^{32}}{\left(\log_{\zeta_0}(2\varepsilon/3)\right)^{8+\log_{\zeta_0}(2\varepsilon/3)} \cdot (2\varepsilon/3)^{8/\log(\zeta_0)}}\right).$$

[20] Recall that this strengthening is used in our list decoding framework.

Now, we turn to the SOS-related parameter L which is constrained to be

$$L \geq c'' \cdot \frac{2^k \cdot k^5}{\varepsilon^{32}},$$

where $c'' > 0$. Note that in this case the exponent $O(L+k)$ appearing in the running time of Theorem 6.17 becomes $O(L)$. Similarly, further simplification leads to

$$L = O\left(\frac{\left(\log_{\zeta_0}(2\varepsilon/3)\right)^5 \cdot (3/2\varepsilon)^{-1/\log(\zeta_0)}}{\varepsilon^{32}}\right).$$

Taking ε_0 to be a constant and simplifying yields the claimed parameters. ∎

7.3 Instantiation to General Base Codes

We can extend Lemma 7.5 to an arbitrary (not necessarily linear) binary base code $\mathcal{C}_1$ with the natural caveat of no longer obtaining linear lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$. However, even if $\mathcal{C}_1$ has small bias, it might not be the case that the difference of any two codewords will have small bias, which is required for list decoding. To this end we modify the code $\mathcal{C}_1$ by employing a projection φ which converts a condition on the distance of the code to a condition on the bias of the difference of any two codewords.

Claim 7.6. *If $\mathcal{C}_1$ is binary code on $[n]$ with relative distance δ and rate r, then there exists an explicit linear projection $\varphi\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that the code $\mathcal{C}_1' = \varphi(\mathcal{C}_1)$ has relative distance at least $\delta/2$ and rate r. Furthermore, for every $z, z' \in \mathcal{C}_1'$ we have*

$$\mathrm{bias}(z - z') \leq 1 - \frac{\delta}{2}.$$

Proof. Take φ to be the projector onto $\mathbb{F}_2^{n-s} \oplus \{0\}^s$ where $s = \lfloor \delta n/2 \rfloor$. Then

$$\mathcal{C}_1' := \varphi(\mathcal{C}_1) = \{(z_1, \ldots, z_{n-s}, \underbrace{0, \ldots, 0}_{s}) \mid (z_1, \ldots, z_n) \in \mathcal{C}_1\},$$

and the claim readily follows. ∎

With this modification in mind, we can now restate and prove Theorem 7.1.

Theorem 7.7 (Direct Sum Lifting on HDX (Restatement of Theorem 7.1))**.** *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. There exist universal constants $c, C > 0$ such that for any γ-HDX $X(\leq d)$ on ground set $X(1) = [n]$ and Π_1 uniform, if*

$$\gamma \leq (\log(1/\varepsilon))^{-C \cdot \log(1/\varepsilon)} \quad \text{and} \quad d \geq c \cdot \frac{(\log(1/\varepsilon))^2}{\varepsilon},$$

then the following holds:

For every binary code $\mathcal{C}_1$ with $\Delta(\mathcal{C}_1) \geq 1/2 - \varepsilon_0$ on $X(1) = [n]$, there exists a binary lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1))$ with $\Delta(\mathcal{C}_k) \geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ on $X(k)$ where $k = O(\log(1/\varepsilon))$, φ is an explicit linear projection, and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time*
$$n^{\varepsilon^{-O(1)}} \cdot f(n),$$
where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.
- *[Rate] The rate r_k of $\mathcal{C}_k$ satisfies $r_k = r_1 \cdot |X(1)| \,/\, |X(k)|$ where r_1 is the relative rate of $\mathcal{C}_1$.*
- *[Linearity] If $\mathcal{C}_1$ is linear, then φ is the identity and $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ is linear.*

Proof. By virtue of Lemma 7.5, it is enough to consider when $\mathcal{C}_1$ is not linear. Note that in the proof of Lemma 7.5 the only assumption about linearity of $\mathcal{C}_1$ we used to obtain $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robustness was that the sum of two codewords is in the code and hence it has small bias. For a general code $\mathcal{C}_1$ of constant distance $1/2 - \varepsilon_0$, applying Claim 7.6 we obtain a new code $\mathcal{C}_1'$ with this guarantee at the expense of a distance $1/2$ times the original one. Naturally, in the current proof we no longer obtain a linear lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1')$. Excluding the two previous remarks the proof of Theorem 7.1 is now the same as the proof of Lemma 7.5. ∎

8 List Decoding Direct Product Codes

8.1 Direct Product Codes

Having developed a decoding algorithm for direct sum, a promising strategy for list decoding other lifted codes on expanding objects is reducing them to instances of direct sum list decoding. One such reduction involves the direct product lifting, which was first studied in the context of samplers by Alon et al. in [ABN+92]. The direct product lifting collects the entries of a code on each subset of size ℓ.

Definition 8.1 (Direct Product Lifting)**.** *Let $\mathcal{C}_1 \subseteq \mathbb{F}_2^n$ be a base code on $X(1) = [n]$. The* direct product lifting *of a word $z \in \mathbb{F}_2^n$ on a collection $X(\ell)$ is $\mathrm{dprod}_{X(\ell)}(z) = (x_t)_{t \in X(\ell)}$, where $x_t = (z_i)_{i \in t}$. The direct product lifting of the entire code is $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1) = \{\mathrm{dprod}_{X(\ell)}(z) \mid z \in \mathcal{C}_1\}$, which is a code of length $|X(\ell)|$ over the alphabet $\mathbb{F}_2^\ell$.*

If X is a HDX, its sampling properties ensure that the direct product lifting has very high distance. It follows from the definition that if the bipartite graph between $X(1)$ and $X(k)$ is an (η, δ)-sampler and the code $\mathcal{C}_1$ has minimum distance η, then the direct product lifting $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ has minimum distance at least $(1 - \delta)$. Recalling from Fact 4.8 that the bipartite graph between two levels of a HDX can be a sampler with arbitrarily small parameters if the expansion is good enough, we can reasonably hope to list decode the direct product lifting on a HDX up to a distance close to 1. In fact, Dinur et al. [DHK+19] provided a list decoding algorithm accomplishing exactly that. We offer a very different approach to the same list decoding problem.

8.2 Direct Product List Decoding

We will reduce direct product decoding on $X(\ell)$ to direct sum decoding on $X(k)$, where $k \approx \ell/2$. This requires converting a received word $\tilde{x} \in (\mathbb{F}_2^\ell)^{X(\ell)}$ to a word $\tilde{y} \in \mathbb{F}_2^{X(k)}$ that

we will decode using the direct sum algorithm. If we knew that $\tilde{x} = \mathrm{dprod}_{X(\ell)}(\tilde{z})$ for some $\tilde{z} \in \mathbb{F}_2^{X(1)}$, we would do so by simply taking $\tilde{y}_{\mathfrak{s}} = \sum_{i \in \mathfrak{s}} \tilde{z}_i$ to be the direct sum lifting on each edge $\mathfrak{s}$; that is, $\tilde{y} = \mathrm{dsum}_{X(k)}(\tilde{z})$.

Unfortunately, performing list decoding also involves dealing with words $\tilde{x}$ that might not have arisen from the direct product lifting. To construct a corrupted instance of direct sum $\tilde{y}$ from $\tilde{x}$, we need to assign values to each face $\mathfrak{s} \in X(k)$ based only on the information we have on the faces $X(\ell)$, as there is no word on the ground set to refer to. Since different faces $\mathfrak{t}, \mathfrak{t}' \in X(\ell)$ containing $\mathfrak{s}$ might not agree on $\mathfrak{s}$, there could be ambiguity as to what value to assign for the sum on $\mathfrak{s}$.

This is where the D-flatness of the distribution Π_ℓ (which holds for the γ-HDX construction described in Lemma 4.5) comes in. Recall that to obtain codewords in $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ without weights on their entries, we duplicate each face $\mathfrak{t} \in X(\ell)$ at most D times to make the distribution Π_ℓ uniform. To perform the same kind of duplication on $X(k)$ that makes Π_k uniform, note that each face $\mathfrak{s} \in X(k)$ has $\Pi_k(\mathfrak{s})$ proportional to $|\{\mathfrak{t} \in X(\ell) \mid \mathfrak{t} \supset \mathfrak{s}\}|$ (where $X(\ell)$ is thought of as a multiset), so we will create one copy of $\mathfrak{s}$ for each $\mathfrak{t}$ containing it. Thus we can assign a unique $\mathfrak{t} \supset \mathfrak{s}$ to each copy. By downward closure, the distribution on $X(\ell)$ obtained by choosing $\mathfrak{s}$ uniformly from the multiset $X(k)$ and then selecting its associated face $\mathfrak{t}$ will be uniform, just like Π_ℓ. With this careful duplication process, we are ready to define the function ρ_k that takes a corrupted direct product word $\tilde{x}$ to a corrupted direct sum word $\tilde{y}$.

Definition 8.2 (Reduction Function)**.** *Let $k < \ell$ and X be a HDX where the distribution Π_ℓ is D-flat. Duplicate faces in $X(k)$ so that Π_k is uniform, and assign a face $\mathfrak{t}_{\mathfrak{s}} \in X(\ell)$ to each $\mathfrak{s} \in X(k)$ (after duplication) such that $\mathfrak{t}_{\mathfrak{s}}$ is distributed according to Π_ℓ when $\mathfrak{s}$ is selected uniformly from $X(k)$. The function $\rho_k : (\mathbb{F}_2^\ell)^{X(\ell)} \to \mathbb{F}_2^{X(k)}$ is defined as*

$$(\rho_k(\tilde{x}))_{\mathfrak{s}} = \sum_{i \in \mathfrak{s}} (\tilde{x}_{\mathfrak{t}_{\mathfrak{s}}})_i.$$

The reduction function ρ_k resolves the ambiguity of which face $\mathfrak{t} \supset \mathfrak{s}$ to sample the sum from by assigning a different face to each copy of $\mathfrak{s}$ in a manner compatible with the distribution Π_ℓ. Observe that if $\tilde{x} = \mathrm{dprod}_{X(\ell)}(\tilde{z})$ for some $\tilde{z} \in \mathbb{F}_2^{X(1)}$, then $\rho_k(\tilde{x}) = \mathrm{dsum}_{X(k)}(\tilde{z})$.

The following lemma shows that performing this reduction from direct product to direct sum maintains agreement between words. It essentially says that if a received word $\tilde{x}$ exhibits some agreement with $x \in \mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$, then there is a k for which $\rho_k(\tilde{x})$ and $\rho_k(x)$ have agreement larger than $1/2$.

Lemma 8.3 (Product-to-sum agreement)**.** *Fix $\varepsilon > 0$ and $C' > 2$. Let $z \in \mathcal{C}_1$, $x = \mathrm{dprod}_{X(\ell)}(z)$, and $\tilde{x} \in (\mathbb{F}_2^\ell)^{X(\ell)}$. If $\Delta(x, \tilde{x}) \leq 1 - \varepsilon$, then there exists a k satisfying*

$$|k - \ell/2| < \frac{1}{2}\sqrt{C' \ell \log(1/\varepsilon)}$$

such that

$$\Delta(y, \tilde{y}) \leq 1/2 - \varepsilon/2 + \varepsilon^{C'/2},$$

where $y = \rho_k(x)$ and $\tilde{y} = \rho_k(\tilde{x})$ are words in $\mathbb{F}_2^{X(k)}$.

Proof. For $t \in X(\ell)$ and $\mathfrak{s} \subseteq \mathfrak{t}$, define the function $\chi_{\mathfrak{s},\mathfrak{t}} : \mathbb{F}_2^{\mathfrak{t}} \to \{-1, 1\}$ by

$$\chi_{\mathfrak{s},\mathfrak{t}}(w) = \prod_{i \in \mathfrak{s}} (-1)^{w_i}.$$

For each face $\mathfrak{t} \in X(\ell)$, consider the expectation $\mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}}[\chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})]$, where $\mathfrak{s}$ is a subset of $\mathfrak{t}$ of any size chosen uniformly. If $x_\mathfrak{t} = \tilde{x}_\mathfrak{t}$, which happens for at least ε fraction of faces $\mathfrak{t}$, the expression in the expectation is always 1. Otherwise, this expectation is zero, so taking the expectation over the faces yields

$$\mathbb{E}_{\mathfrak{t} \sim \Pi_\ell} \mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}}[\chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})] = \mathop{\mathbb{P}}_{t \sim \Pi_\ell}[x_\mathfrak{t} = \tilde{x}_\mathfrak{t}] \geq \varepsilon.$$

We would like to restrict to a fixed size of faces $\mathfrak{s}$ for which this inequality holds; as this will be the size of the direct sum faces, we need to make sure it's large enough to give us the expansion required for decoding later. Using a Chernoff bound (Fact A.2 with $a = \sqrt{C'\ell \log(1/\varepsilon)}$), we see that the size of the faces is highly concentrated around $\ell/2$:

$$\mathop{\mathbb{P}}_{\mathfrak{s} \subseteq \mathfrak{t}} \left[\left| |\mathfrak{s}| - \frac{\ell}{2} \right| \geq \frac{1}{2} \sqrt{C'\ell \log(1/\varepsilon)} \right] \leq 2e^{-C' \log(1/\varepsilon)/2} \leq 2\varepsilon^{C'/2}.$$

Let I be the interval

$$I = \left(\frac{\ell}{2} - \frac{1}{2}\sqrt{C'\ell \log(1/\varepsilon)}, \frac{\ell}{2} + \frac{1}{2}\sqrt{C'\ell \log(1/\varepsilon)} \right).$$

The expectation inequality becomes

$$\begin{aligned} \varepsilon &\leq \mathbb{E}_{\mathfrak{t} \sim \Pi_\ell}[\mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}}[\mathbb{1}_{|\mathfrak{s}| \in I} \cdot \chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})] + \mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}}[\mathbb{1}_{|\mathfrak{s}| \notin I} \cdot \chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})]] \\ &\leq \mathbb{E}_{\mathfrak{t} \sim \Pi_\ell} \mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}, |\mathfrak{s}| \in I}[\chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})] + 2\varepsilon^{C'/2}. \end{aligned}$$

Thus there exists a $k \in I$ such that

$$\varepsilon - 2\varepsilon^{C'/2} \leq \mathbb{E}_{\mathfrak{t} \sim \Pi_\ell} \mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}, |\mathfrak{s}| = k}[\chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})].$$

Choosing a face $\mathfrak{t}$ and then a uniformly random $\mathfrak{s} \subseteq \mathfrak{t}$ of size k results in choosing $\mathfrak{s}$ according to Π_k. Moreover, the edge $\mathfrak{t}_\mathfrak{s}$ containing $\mathfrak{s}$ from Definition 8.2 is distributed according to Π_ℓ. Bearing in mind the definitions of $y_\mathfrak{s}$ and $\tilde{y}_\mathfrak{s}$, we have

$$\begin{aligned} \varepsilon - 2\varepsilon^{C'/2} &\leq \mathbb{E}_{\mathfrak{t} \sim \Pi_\ell} \mathbb{E}_{\mathfrak{s} \subseteq \mathfrak{t}, |\mathfrak{s}| = k}[\chi_{\mathfrak{s},\mathfrak{t}}(x_\mathfrak{t} - \tilde{x}_\mathfrak{t})] \\ &= \mathbb{E}_{\mathfrak{s} \sim \Pi_k}[\chi_{\mathfrak{s},\mathfrak{t}_\mathfrak{s}}(x_{\mathfrak{t}_\mathfrak{s}} - \tilde{x}_{\mathfrak{t}_\mathfrak{s}})] \\ &= \mathbb{E}_{\mathfrak{s} \sim \Pi_k}[(-1)^{(\rho_k(x))_\mathfrak{s} - (\rho_k(\tilde{x}))_\mathfrak{s}}] \\ &= \mathrm{bias}(y - \tilde{y}) \end{aligned}$$

which translates to a Hamming distance of $\Delta(y, \tilde{y}) \leq 1/2 - \varepsilon/2 + \varepsilon^{C'/2}$. ∎

With Lemma 8.3 in hand to reduce a direct product list decoding instance to a direct sum list decoding instance, we can decode by using a direct sum list decoding algorithm as a black box.

Algorithm 8.4 (Direct Product List Decoding Algorithm).

Input *A word* $\tilde{x} \in (\mathbb{F}_2^\ell)^{X(\ell)}$ *with distance at most* $(1-\varepsilon)$ *from* $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$

Output *The list* $\mathcal{L}' = \{z \in \mathbb{F}_2^n \mid \Delta(\mathrm{dprod}_{X(\ell)}(z), \tilde{x}) \leq 1-\varepsilon\}$

1. *Let* I *be the interval* $\left(\ell/2 - \sqrt{C'\ell\log(1/\varepsilon)}/2, \ell/2 + \sqrt{C'\ell\log(1/\varepsilon)}/2\right)$.
2. *For each integer* $k \in I$*, run the direct sum list decoding algorithm on the input* $\tilde{y} = \rho_k(\tilde{x}) \in \mathbb{F}_2^{X(k)}$ *to obtain a coupled list* $\mathcal{L}_k$ *of all pairs* (z, y) *with* $\Delta(y, \tilde{y}) \leq 1/2 - \varepsilon/2 + \varepsilon^{C'/2}$.
3. *Let* $\mathcal{L} = \cup_{k\in I}\{z \in \mathcal{C}_1 \mid (z, y) \in \mathcal{L}_k\}$.
4. *Let* $\mathcal{L}' = \{z \in \mathcal{L} \mid \Delta(\mathrm{dprod}_{X(\ell)}(z), \tilde{x}) \leq 1-\varepsilon\}$.
5. *Output* $\mathcal{L}'$.

Theorem 8.5 (Product-to-sum Reduction). *Let* $\varepsilon > 0$ *and* $C' > 2$*. Let* $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ *be the direct product lifting of a base code* $\mathcal{C}_1$ *on a simplicial complex* X*. If the direct sum lifting* $\mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ *is list decodable up to distance* $\left(1/2 - \varepsilon/2 + \varepsilon^{C'/2}\right)$ *in time* $\tilde{f}(n)$ *for all* k *satisfying* $|k - \ell/2| < \sqrt{C'\ell\log(1/\varepsilon)}/2$*, then Algorithm 8.4 list decodes* $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ *up to distance* $(1-\varepsilon)$ *in running time*

$$\sqrt{C'\ell\log(1/\varepsilon)}\tilde{f}(n) + |X(\ell)||\mathcal{L}|.$$

Proof. Let $\tilde{x} \in (\mathbb{F}_2^\ell)^{X(\ell)}$ be a received word and let $z \in \mathcal{C}_1$ satisfy $\Delta(\mathrm{dprod}_{X(\ell)}(z), \tilde{x}) \leq 1-\varepsilon$. By Lemma 8.3, there exists a $k \in I$ such that $\Delta(y, \tilde{y}) \leq 1/2 - \varepsilon/2 + \varepsilon^{C'/2}$. Thanks to this distance guarantee, the pair (z, y) will appear on the list $\mathcal{L}_k$ when the direct sum list decoding algorithm is run for this k. Then z will be on the combined list $\mathcal{L}$ and the trimmed list $\mathcal{L}'$, with the trimming ensuring that no elements of $\mathcal{C}_1$ appear on this list beyond those with the promised distance. The set $\{\mathrm{dprod}_{X(\ell)}(z) \mid z \in \mathcal{L}'\}$ thus contains all words in $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ with distance at most $(1-\varepsilon)$ from $\tilde{x}$.

To obtain the promised the running time, note that Algorithm 8.4 runs the direct sum list decoding algorithm $\sqrt{C'\ell\log(1/\varepsilon)}$ times and then computes the direct product lifting of each element of $\mathcal{L}$ in the trimming step. ∎

Combining the parameters in the reduction with those required for our direct sum list decoding algorithm, we obtain the following. Note that for very small values of ε, we can choose the constant C' to be close to 2, and we will be list decoding the direct sum code up to distance $1/2 - \sqrt{\beta} \approx 1/2 - \varepsilon/4$.

Corollary 8.6 (Direct Product List Decoding). *Let* $\varepsilon_0 < 1/2$ *be a constant, and let* $\varepsilon > 0$*,* $C' \geq 2 + 4/\log(1/\varepsilon)$*, and* $\beta = (\varepsilon/2 - \varepsilon^{C'/2})^2$*. There exist universal constants* $c, C > 0$ *such that for any* γ*-HDX* $X(\leq d)$ *on ground set* $[n]$ *and* Π_1 *uniform, if*

$$\gamma \leq \log(1/\beta)^{-C\log(1/\beta)} \quad \text{and} \quad d \geq c \cdot \frac{\log(1/\beta)^2}{\beta},$$

then the following holds:

For every binary code $\mathcal{C}_1$ *with* $\Delta(\mathcal{C}_1) \geq 1/2 - \varepsilon_0$ *on* $X(1) = [n]$*, there exists a lifted code* $\mathcal{C}_\ell = \mathrm{dprod}_{X(\ell)}(\varphi(\mathcal{C}_1))$ *on* C_ℓ *where* $\ell = O(\log(1/\beta))$*,* φ *is an explicit linear projection, and*

- *[Efficient List Decoding] If $\tilde{x}$ is $(1-\varepsilon)$-close to $\mathcal{C}_\ell$, then we can compute the list of all codewords of $\mathcal{C}_\ell$ that are $(1-\varepsilon)$-close to $\tilde{x}$ in time $n^{\varepsilon^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of the unique decoding algorithm for $\mathcal{C}_1$.*
- *[Rate] The rate r_ℓ of $\mathcal{C}_\ell$ satisfies $r_\ell = r_1 \cdot |X(1)| / (\ell\,|X(\ell)|)$, where r_1 is the relative rate of $\mathcal{C}_1$.*
- *[Linearity] If $\mathcal{C}_1$ is linear, then φ is the identity and $\mathcal{C}_\ell$ is linear.*

Proof. Let $k = \ell/2 - \sqrt{C'\ell\log(1/\varepsilon)}/2$. The choice of parameters ensures that $\mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ is list decodable up to distance $1/2 - \sqrt{\beta} = 1/2 - \varepsilon/2 + \varepsilon^{C'/2}$ in running time $g(n) = n^{\beta^{-O(1)}} f(n)$ by Theorem 7.1 (noting that the bound on C' implies $\beta \geq \varepsilon^2/16$). Since increasing k increases the list decoding radius of the direct sum lifting, this holds for any value of k with $|k - \ell/2| \leq \sqrt{C'\ell\log(1/\varepsilon)}/2$. By Theorem 8.5, the direct product lifting $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ is list decodable up to distance $(1-\varepsilon)$ in running time

$$\sqrt{C'\ell\log(1/\varepsilon)}\, n^{\beta^{-O(1)}} f(n) + |X(\ell)|\,|\mathcal{L}|\,.$$

The HDX has $|X(\ell)| \leq \binom{n}{\ell} = n^{O(\log(1/\beta))}$, and the list size $|\mathcal{L}|$ is bounded by the sum of the sizes of the lists $\mathcal{L}_k$ obtained from each direct sum decoding. Each of these lists has $|\mathcal{L}_k| \leq 1/(2\beta)$ by the Johnson bound (see Remark 6.14) and the number of lists is constant with respect to n, so the overall running time is dominated by the first term, $n^{\beta^{-O(1)}} f(n) = n^{\varepsilon^{-O(1)}} f(n)$.

The rate and linearity guarantees follow in the same manner as they do in Theorem 7.1, where the rate calculation requires a slight modification for dealing with the increased alphabet size and φ is the projection from Claim 7.6. ∎

Using Corollary 8.6 with HDXs obtained from Ramanujan complexes as in Corollary 7.2, we can perform list decoding with an explicit construction up to distance $(1-\varepsilon)$ with HDX parameters $d = O(\log(1/\varepsilon)^2/\varepsilon^2)$ and $\gamma = (\log(1/\varepsilon))^{-O(\log(1/\varepsilon))}$. The direct product list decoding algorithm of Dinur et al. [DHK+19] is based on a more general expanding object known as a double sampler. As the only known double sampler construction is based on a HDX, we can compare our parameters to their HDX requirements of $d = O(\exp(1/\varepsilon))$ and $\gamma = O(\exp(-1/\varepsilon))$.

9 Instantiation II: Direct Sum on Expander Walks

We instantiate the list decoding framework to the direct sum lifting where the sum is taken over the collection $X(k)$ of length k walks of a sufficiently expanding graph G. To stress the different nature of this collection and its dependence on G we equivalently denote $X(k)$ by $W_G(k)$ and endow it with a natural measure in Definition 9.1.

Definition 9.1 (Walk Collection)**.** *Let $G = (V, E, w)$ be a weighted graph with weight distribution $w\colon E \to [0,1]$. For $k \in \mathbb{N}^+$, we denote by $W_G(k)$ the collection of all walks of length k in G, i.e.,*

$$W_G(k) \coloneqq \{w = (w_1, \ldots, w_k) \mid w \text{ is a walk of length } k \text{ in } G\}.$$

We endow $W_G(k)$ with the distribution Π_k arising from taking a random vertex w_1 according to the stationary distribution on V and then taking $k-1$ steps according to the normalized random walk operator of G.

One simple difference with respect to the HDX case is that now we are working with a collection of (ordered) tuples instead of subsets. The Propagation Rounding Algorithm 6.5 remains the same, but we need to establish the tensorial properties of $W_G(k)$ which is done in Section 9.1.

The main result of this section follows.

Theorem 9.2 (Direct Sum Lifting on Expander Walks)**.** *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. There exists a universal constant $C > 0$ such that for any d-regular γ-two-sided expander graph G on ground set $W_G(1) = [n]$, if $\gamma \leq \varepsilon^C$, then the following holds:*

For every binary code $\mathcal{C}_1$ with $\Delta(\mathcal{C}_1) \geq 1/2 - \varepsilon_0$ on $W_G(1) = [n]$, there exists a binary lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1))$ with $\Delta(\mathcal{C}_k) \geq 1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}$ on $W_G(k)$ where $k = O(\log(1/\varepsilon))$, φ is an explicit linear projection, and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time*
$$n^{\varepsilon^{-O(1)}} \cdot f(n),$$
where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.
- *[Rate] The rate r_k of $\mathcal{C}_k$ satisfies $r_k = r_1/d^{k-1}$ where r_1 is the relative rate of $\mathcal{C}_1$.*
- *[Linearity] If $\mathcal{C}_1$ is linear, then φ is the identity and $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ is linear.*

In particular, we apply Theorem 9.2 to the explicit family of Ramanujan expanders of Lubotzky et al. from Theorem 9.3.

Theorem 9.3 (Lubotzky-Phillips-Sarnak abridged [LPS88])**.** *Let $p \equiv 1 \pmod 4$ be a prime. Then there exists an explicit infinite family of $(p+1)$-regular Ramanujan graphs $G_1, G_2, \ldots$ on $n_1 < n_2 < \cdots$ vertices, i.e., $\sigma_2(G_i) \leq 2 \cdot \sqrt{p}/(p+1)$.*

In order to construct Ramanujan expanders with arbitrarily good expansion, we will use the following lemma for finding primes.

Lemma 9.4 (From [TS17])**.** *For every $\alpha > 0$ and sufficiently large n, there exists an algorithm that given a and m relatively prime, runs in time $\mathrm{poly}(n)$ and outputs a prime number p with $p \equiv a \pmod m$ in the interval $[(1-\alpha)n, n]$.*

This results in Corollary 9.5.

Corollary 9.5. *Let $\varepsilon_0 < 1/2$ be a constant and $\varepsilon \in (0, \varepsilon_0)$. There is an infinite sequence of explict Ramanujan expanders $G_1, G_2, \ldots$ on ground sets of size $n_1 < n_2 < \cdots$ such that the following holds:*

For every sequence of binary codes $\mathcal{C}_1^{(i)}$ on $[n_i]$ with rate and distance uniformly bounded by $r_1^{(i)}$ and $(1/2 - \varepsilon_0)$ respectively, there exists a sequence of binary lifted codes $\mathcal{C}_k^{(i)} = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1^{(i)}))$ on a collection $X_i(k)$ with distance $\left(1/2 - \varepsilon^{\Omega_{\varepsilon_0}(1)}\right)$ where φ is an explicit linear projection and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2 - \sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y}, \mathcal{C}_1, \mathcal{C}_k)$ (c.f. Definition 6.15) in time $n^{\varepsilon^{-O(1)}} \cdot f(n)$, where $f(n)$ is the running time of a unique decoding algorithm of $\mathcal{C}_1$.*
- *[Explicit Construction] The collection $W_{G_i}(k)$ is obtained from length k walks on a Ramanujan d-regular expander G_i where $k = O(\log(1/\varepsilon))$, $d = 8 \cdot \varepsilon^{-O(1)}$ and $\gamma = \varepsilon^{O(1)}$.*
- *[Rate] The rate $r_k^{(i)}$ of $\mathcal{C}_k^{(i)}$ satisfies $r_k^{(i)} \geq r_1^{(i)} \cdot \varepsilon^{O(\log(1/\varepsilon))}$.*
- *[Linearity] If $\mathcal{C}_1^{(i)}$ is linear, then φ is the identity and $\mathcal{C}_k^{(i)} = \mathrm{dsum}_{X(k)}(\mathcal{C}_1^{(i)})$ is linear.*

Proof. Using Lemma 9.4 with $a = 1$ and $m = 4$, we see that given n, α, a prime p such that $p \equiv 1 \pmod 4$ may be found in the interval $[(1-\alpha)n, n]$ for large enough n. For Ramanujan expanders, the condition that $\gamma \leq \varepsilon^C$ translates to $p \geq 4 \cdot \varepsilon^{-2C}$. Choose $\alpha = 1/2$ and $n > 8 \cdot \varepsilon^{-2C}$ so that we find a prime greater than $4 \cdot \varepsilon^{-2C}$, but at most $8 \cdot \varepsilon^{-2C}$.

Based on this prime, we use the above Theorem 9.3 to get a family of Ramanujan graphs $G_1, G_2, \ldots$ with $n_1 < n_2 < \cdots$ vertices, such that the degree is bounded by $8\varepsilon^{-2C}$. Using the parameters of this family in Theorem 9.2, we obtain the desired claims. ∎

9.1 Expander Walks are Two-Step Tensorial

To apply the list decoding framework we need to establish the tensorial parameters of expander walks $W_G(k)$ for a γ-two-sided expander graph G. Although the tensorial property is precisely what the abstract list decoding framework uses, when faced with a concrete object such as $W_G(k)$ it will be easier to prove that it satisfies a *splittable* property defined in [AJT19] for complexes which implies the tensorial property. In turn, this splittable property is defined in terms of some natural operators denoted *Swap* operators whose definition is recalled in Section 9.1.2 in a manner tailored to the present case $X(k) = W_G(k)$. Then, in Section 9.1.3, we formally define the splittable property and show that the expansion of the Swap operator is controlled by the expansion parameter γ of G allowing us to deduce the splittable parameters of $W_G(k)$. Finally, in Section 9.1.4, we show how $W_G(k)$ being splittable gives the tensorial parameters. Some results are quite similar to the hypergraph case in [AJT19] (which built on [BRS11]). The key contribution in this new case of $W_G(k)$ is observing the existence of these new Swap operators along with their expansion properties.

9.1.1 Emergence of Swap Operators

To motivate the study of Swap operators on $W_G(k)$, we show how they naturally emerge from the study of k-CSPs. The treatment is quite similar to the hypergraph case developed in [AJT19], but this will give us the opportunity to formalize the details that are specific to $W_G(k)$. Suppose that we solve a k-CSP instance as defined in Section 2.4 whose constraints were placed on the tuples corresponding to walks in $W_G(k)$. The result is a local PSD ensemble $\{\mathbf{Z}\}$ which can then be fed to the Propagation Rounding Algorithm 6.5. It is easy to show that the tensorial condition of Eq. (12) (below) is sufficient to guarantee an approximation to this k-CSP on $W_G(k)$ within μ additive error. The precise parameters are

given in Section 9.1.5. For now, we take this observation for granted and use it to show how the Swap operators emerge in obtaining the inequality

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \{\mathbf{Z}'_{w_1}\} \cdots \{\mathbf{Z}'_{w_k}\} \right\|_1 \leq \mu \tag{12}$$

present in the definition of tensoriality.

The following piece of notation will be convenient when referring to sub-walks of a given walk.

Definition 9.6 (Sub-Walk)**.** *Given $1 \leq i \leq j \leq k$ and $w = (w_1, \ldots, w_k) \in W_G(k)$, we define the sub-walk $w(i,j)$ from w_i to w_j as*

$$w(i,j) := (w_i, w_{i+1}, \ldots, w_j).$$

We will need the following simple observation about marginal distributions of Π_k on sub-walks.

Claim 9.7 (Marginals of the walk distribution)**.** *Let $k \in \mathbb{N}^+$ and $1 \leq i \leq j \leq k$. Then sampling $w \sim \Pi_k$ in $W_G(k)$ and taking $w(i,j)$ induces the distribution Π_{j-i+1} on $W_G(j-i+1)$.*

Proof. Let $w = (w_1, \ldots, w_i, \ldots, w_j, \ldots, w_k) \sim \Pi_k$. Since $w_1 \sim \Pi_1$ where Π_1 is the stationary measure of G and $w_2, \ldots, w_i$ are obtained by $(i-1)$ successive steps of a random walk on G, the marginal distribution on w_i is again the stationary measure Π_1. Then by taking $(j-i)$ successive random walk steps from w_i on G, we obtain a walk $(w_i, \ldots, w_j)$ distributed according to Π_{j-i+1}. ∎

We also need the notion of a *splitting tree* as follows.

Definition 9.8 (Splitting Tree [AJT19])**.** *We say that a binary tree $\mathcal{T}$ is a k-splitting tree if it has exactly k leaves and*

- *the root of $\mathcal{T}$ is labeled with k and all other vertices are labeled with positive integers,*
- *the leaves are labeled with 1, and*
- *each non-leaf vertex satisfies the property that its label is the sum of the labels of its two children.*

The Swap operators arise naturally from the following triangle inequality where the quantity $\mathbb{E}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^k \left\{\mathbf{Z}'_{w(i)}\right\} \right\|_1$ is upper bounded by a sum of terms of the form

$$\mathop{\mathbb{E}}_{w\sim W_G(k_1+k_2)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w(1,k_1)}\right\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} \right\|_1.$$

We view the above expectation as taking place over the edges $W_G(k_1+k_2)$ of a bipartite graph on vertex bipartition $(W_G(k_1), W_G(k_2))$. This graph gives rise to a Swap operator which we formally define later in Section 9.1.2. The following claim shows how a splitting tree defines all terms (and hence also their corresponding graphs and operators) that can appear in this upper bound.

Claim 9.9 (Triangle inequality). *Let $k \in \mathbb{N}^+$ and $\mathcal{T}$ be a k-splitting tree. Then*

$$\mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^{k} \left\{\mathbf{Z}'_{w(i)}\right\} \right\|_1 \leq \sum_{(k_1,k_2)} \mathop{\mathbb{E}}_{w\sim W_G(k_1+k_2)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w(1,k_1)}\right\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} \right\|_1,$$

where the sum $\sum_{(k_1,k_2)}$ is taken over all pairs of labels of the two children of each internal node of $\mathcal{T}$.

Proof. We prove the claim by induction on k. Let (k_1, k_2) be the labels of the children of the root of the splitting tree $\mathcal{T}$. Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ are the corresponding splitting trees rooted at these children with labels k_1 and k_2, respectively. By this choice, we have $k = k_1 + k_2$. Applying the triangle inequality yields

$$\begin{aligned}\mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^{k} \{\mathbf{Z}'_{w_i}\} \right\|_1 \leq & \mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w(1,k_1)}\right\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} \right\|_1 + \\ & \mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \left\{\mathbf{Z}'_{w(1,k_1)}\right\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} - \prod_{i=1}^{k_1} \{\mathbf{Z}'_{w_i}\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} \right\|_1 + \\ & \mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \prod_{i=1}^{k_1} \{\mathbf{Z}'_{w_i}\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} - \prod_{i=1}^{k} \{\mathbf{Z}'_{w_i}\} \right\|_1.\end{aligned}$$

Using the marginalization given by Claim 9.7 on the second and third terms and simplifying, we get

$$\begin{aligned}\mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^{k} \{\mathbf{Z}'_{w_i}\} \right\|_1 \leq & \mathop{\mathbb{E}}_{w\sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w(1,k_1)}\right\} \left\{\mathbf{Z}'_{w(k_1+1,k_2)}\right\} \right\|_1 + \\ & \mathop{\mathbb{E}}_{w\sim W_G(k_1)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^{k_1} \{\mathbf{Z}'_{w_i}\} \right\|_1 + \mathop{\mathbb{E}}_{w\sim W_G(k_2)} \left\| \{\mathbf{Z}'_w\} - \prod_{i=1}^{k_2} \{\mathbf{Z}'_{w_i}\} \right\|_1.\end{aligned}$$

Applying the induction hypothesis to the second term with tree $\mathcal{T}_1$ and to the third term with tree $\mathcal{T}_2$ finishes the proof. ∎

9.1.2 Swap Operators Arising from Expander Walks

We define the Swap operator associated to walks on a given graph G as follows.

Definition 9.10 (Graph Walk Swap Operator). *Let $G = (V, E, w)$ be a weighted graph. Let $k_1, k_2 \in \mathbb{N}^+$ be such that $k = k_1 + k_2$. We define the graph walk Swap operator*

$$\mathsf{S}^{\circ}_{k_1,k_2} \colon \mathbb{R}^{W_G(k_2)} \to \mathbb{R}^{W_G(k_1)}$$

such that for every $f \in \mathbb{R}^{W_G(k_2)}$,

$$\left(\mathsf{S}^{\circ}_{k_1,k_2}(f)\right)(w) \coloneqq \mathbb{E}_{w' : ww' \in W(k)}[f(w')],$$

where ww' denotes the concatenation of the walks w and w'. The operator $\mathsf{S}^{\circ}_{k_1,k_2}$ can be defined more concretely in matrix form such that for every $w \in W_G(k_1)$ and $w' \in W_G(k_2)$,

$$\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)_{w,w'} \coloneqq \frac{\Pi_k(ww')}{\Pi_{k_1}(w)}.$$

Remark 9.11. *Swap operators are Markov operators, so the largest singular value of a Swap operator is bounded by* 1.

Unlike the Swap operators for HDXs described in [AJT19], which are defined using unordered subsets of hyperedges, the Swap operators $\mathsf{S}^{\circ}_{k_1,k_2}$ use sub-walks and are thus directed operators. Instead of analyzing such an operator directly, we will examine the symmetrized version

$$\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2}) = \begin{pmatrix} 0 & \mathsf{S}^{\circ}_{k_1,k_2} \\ \left(\mathsf{S}^{\circ}_{k_1,k_2}\right)^{\dagger} & 0 \end{pmatrix}$$

and show that $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ is the normalized random walk operator of an undirected graph. In particular, $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ defines an undirected weighted bipartite graph on the vertices $W_G(k_1) \cup W_G(k_2)$, where each edge ww' in this graph is weighted according to the transition probability from one walk to the other whenever one of w, w' is in $W_G(k_1)$ and the other is in $W_G(k_2)$. This becomes clear when taking a closer look at the adjoint operator $(\mathsf{S}^{\circ}_{k_1,k_2})^{\dagger}$.

Claim 9.12. *Let $k_1, k_2 \in \mathbb{N}$ and $k = k_1 + k_2$. Define the operator $\mathfrak{S}_{k_1,k_2} \colon \mathbb{R}^{W_G(k_1)} \to \mathbb{R}^{W_G(k_2)}$ such that for every $f \in \mathbb{R}^{W_G(k_1)}$,*

$$\left(\mathfrak{S}_{k_1,k_2}(f)\right)(w') := \mathbb{E}_{w:ww' \in W(k)}[f(w)]$$

for every $w' \in W_G(k_2)$. Then

$$\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)^{\dagger} = \mathfrak{S}_{k_1,k_2}.$$

Proof. Let $f \in C^{W_G(k_1)}$ and $g \in C^{W_G(k_2)}$. We show that $\langle f, \mathsf{S}^{\circ}_{k_1,k_2} g\rangle = \langle \mathfrak{S}_{k_1,k_2} f, g\rangle$. On one hand we have

$$\begin{aligned}
\langle f, \mathsf{S}^{\circ}_{k_1,k_2} g\rangle &= \mathbb{E}_{w \in W_G(k_1)}\left[f(w)\mathbb{E}_{w':ww' \in W_G(k)}[g(w')]\right] \\
&= \mathbb{E}_{w \in W_G(k_1)}\left[f(w) \sum_{w' \in W_G(k_2)} \frac{\Pi_k(ww')}{\Pi_{k_1}(w)} g(w')\right] \\
&= \sum_{w \in W_G(k_1)} \Pi_{k_1}(w) f(w) \sum_{w' \in W_G(k_2)} \frac{\Pi_k(ww')}{\Pi_{k_1}(w)} g(w') \\
&= \sum_{ww' \in W_G(k)} f(w) g(w') \Pi_k(ww').
\end{aligned}$$

On the other hand we have

$$\begin{aligned}
\langle \mathfrak{S}_{k_1,k_2} f, g\rangle &= \mathbb{E}_{w' \in W_G(k_2)}\left[\mathbb{E}_{w:ww' \in W_G(k)}[f(w)] g(w')\right] \\
&= \mathbb{E}_{w' \in W_G(k_2)}\left[\sum_{w \in W_G(k_1)} \frac{\Pi_k(ww')}{\Pi_{k_2}(w')} f(w) g(w')\right] \\
&= \sum_{w' \in W_G(k_2)} \Pi_{k_2}(w') \sum_{w \in W_G(k_1)} \frac{\Pi_k(ww')}{\Pi_{k_2}(w')} f(w) g(w') \\
&= \sum_{ww' \in W_G(k)} f(w) g(w') \Pi_k(ww').
\end{aligned}$$

Hence, $\mathfrak{S}_{k_1,k_2} = (\mathsf{S}^{\circ}_{k_1,k_2})^{\dagger}$ as claimed. ∎

9.1.3 Swap Operators are Splittable

At a high level, the expansion of a certain collection of Swap walks $\mathsf{S}^{\circ}_{k_1,k_2}$ ensures that we can round the SOS solution and this gives rise to the *splittable* notion, which we tailor to the $W_G(k)$ case after recalling some notation.

Remark 9.13. *We establish the definitions in slightly greater generality than needed for our coding application since this generality is useful for solving k-CSP instances on $W_G(k)$ for more general graphs G that are not necessarily expanders (c.f. Section 9.1.5). Solving these kinds of k-CSPs might be of independent interest. For the coding application, the threshold rank (Definition 9.14) will be one, i.e., we will be working with expander graphs.*

Definition 9.14 (Threshold Rank of Graphs (from [BRS11]))**.** *Let $G = (V, E, w)$ be a weighted graph on n vertices and A be its normalized random walk matrix. Suppose the eigenvalues of A are $1 = \lambda_1 \geq \cdots \geq \lambda_n$. Given a parameter $\tau \in (0, 1)$, we denote the threshold rank of G by $\mathrm{rank}_{\geq\tau}(\mathsf{A})$ (or $\mathrm{rank}_{\geq\tau}(G)$) and define it as*

$$\mathrm{rank}_{\geq\tau}(\mathsf{A}) \coloneqq |\{i \mid \lambda_i \geq \tau\}|.$$

Let $\mathrm{Swap}(\mathcal{T}, W_G(\leq k))$ be the set of all swap graphs over $W_G(\leq k)$ finding representation in the splitting tree $\mathcal{T}$, i.e., for each internal node with leaves labeled k_1 and k_2 we associate the undirected Swap operator $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$.

Given a threshold parameter $\tau \leq 1$ and a set of normalized adjacency matrices $\mathcal{A} = \{\mathsf{A}_1, \ldots, \mathsf{A}_s\}$, we define the threshold rank $\mathrm{rank}_{\geq\tau}(\mathcal{A})$ of $\mathcal{A}$ as

$$\mathrm{rank}_{\geq\tau}(\mathcal{A}) \coloneqq \max_{\mathsf{A}\in\mathcal{A}} \mathrm{rank}_{\geq\tau}(\mathsf{A}),$$

where $\mathrm{rank}_{\geq\tau}(\mathsf{A})$ denotes the usual threshold rank of A as in Definition 9.14.

Definition 9.15 ($(\mathcal{T}, \tau, r)$-splittability [AJT19])**.** *A collection $W_G(\leq k)$ is said to be $(\mathcal{T}, \tau, r)$-splittable if $\mathcal{T}$ is a k-splitting tree and*

$$\mathrm{rank}_{\geq\tau}(\mathrm{Swap}(\mathcal{T}, W_G)) \leq r.$$

If there exists some k-splitting tree $\mathcal{T}$ such that $W_G(\leq k)$ is $(\mathcal{T}, \tau, r)$-splittable, the instance $W_G(\leq k)$ will be called a (τ, r)-splittable instance.

We show that the expansion of $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ is inherited from the expansion of its defining graph G. To this end we will have to overcome the hurdle that $W_G(k) \subseteq V^k$ is not necessarily a natural product space, but it can be made so with the proper representation.

Lemma 9.16. *Let $G = (V = [n], E)$ be a d-regular graph with normalized random walk operator A_G. Then for every $k_1, k_2 \in \mathbb{N}^+$, there are representations of $\mathsf{S}^{\circ}_{k_1,k_2}$ and A_G as matrices such that*

$$\mathsf{S}^{\circ}_{k_1,k_2} = \mathsf{A}_G \otimes \mathsf{J}/d^{k_2-1},$$

where $\mathsf{J} \in \mathbb{R}^{[d]^{k_1-1}\times[d]^{k_2-1}}$ is the all ones matrix.

Proof. Partition the set of walks $W_G(k_1)$ into the sets $W_1, \ldots, W_n$, where $w \in W_i$ if the last vertex of the walk is $w_{k_1} = i$. Similarly, partition $W_G(k_2)$ into the sets $W'_1, \ldots, W'_n$, where

$w' \in W'_j$ if the first vertex of the walk is $w'_1 = j$. Note that $|W_i| = d^{k_1-1}$ for all i and $\left|W'_j\right| = d^{k_2-1}$ for all j.

Now order the rows of the matrix $\mathsf{S}^{\circ}_{k_1,k_2}$ so that all of the rows corresponding to walks in W_1 appear first, followed by those for walks in W_2, and so on, with an arbitrary order within each set. Do a similar re-ordering of the columns for the sets $W'_1, \ldots, W'_n$. Observe that

$$\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)_{w,w'} = \frac{\Pi_{k_1+k_2}(ww')}{\Pi_{k_1}(w)} = \frac{\mathbf{1}\left[w_{k_1} \text{ is adjacent to } w'_1\right]}{d^{k_2-1}},$$

which only depends on the adjacency of the last vertex of w and the first vertex of w'. If the vertices i and j are adjacent, then $\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)_{w,w'} = 1/d^{k_2-1}$ for every $w \in W_i$ and $w' \in W'_j$; otherwise, $\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)_{w,w'} = 0$. Since the walks in the rows and columns are sorted according to their last and first vertices, respectively, the matrix $\mathsf{S}^{\circ}_{k_1,k_2}$ exactly matches the tensor product $\mathsf{A}_G \otimes \mathsf{J}/d^{k_2-1}$, where the rows and columns of A_G are sorted according to the usual ordering on $[n]$. ∎

Corollary 9.17. *Let $G = (V, E)$ be a γ-two-sided spectral expander with normalized random walk operator A_G. Then for every $k_1, k_2 \in \mathbb{N}^+$,*

$$\lambda_2(\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})) \leq \gamma.$$

Proof. To make the presentation reasonably self-contained, we include the proof of the well-known connection between the singular values of $\mathsf{S}^{\circ}_{k_1,k_2}$ and the eigenvalues of $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$. Using Lemma 9.16 and the fact that $\sigma_i(\mathsf{A}_G \otimes \mathsf{J}/d^{k_2-1}) = \sigma_i(\mathsf{A}_G)$, we have $\sigma_i(\mathsf{S}^{\circ}_{k_1,k_2}) = \sigma_i(\mathsf{A}_G)$. Since

$$\left(\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})^{\dagger}\right)\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2}) = \begin{pmatrix} \mathsf{S}^{\circ}_{k_1,k_2}\left(\mathsf{S}^{\circ}_{k_1,k_2}\right)^{\dagger} & 0 \\ 0 & \left(\mathsf{S}^{\circ}_{k_1,k_2}\right)^{\dagger}\mathsf{S}^{\circ}_{k_1,k_2} \end{pmatrix},$$

the nonzero singular values of $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ are the same as the nonzero singular values of $\mathsf{S}^{\circ}_{k_1,k_2}$. As $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ is the random walk operator of a bipartite graph, the spectrum of $\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})$ is symmetric around 0 implying that its nonzero eigenvalues are

$$\pm\sigma_1(\mathsf{S}^{\circ}_{k_1,k_2}), \pm\sigma_2(\mathsf{S}^{\circ}_{k_1,k_2}), \ldots = \pm\sigma_1(\mathsf{A}_G), \pm\sigma_2(\mathsf{A}_G), \ldots$$

Hence, the second-largest of these is $\lambda_2(\mathcal{U}(\mathsf{S}^{\circ}_{k_1,k_2})) = \sigma_2(\mathsf{A}_G) \leq \gamma$. ∎

Applying this spectral bound on $\mathcal{U}(\mathsf{S}^{\circ}_{k,k})$ to each internal node of any splitting tree readily gives the splittability of $W_G(k)$.

Corollary 9.18. *If G is a γ-two-sided spectral expander, then for every $k \in \mathbb{N}^+$ the collection $W_G(k)$ endowed with Π_k is $(\gamma, 1)$-splittable (for all choices of splitting trees).*

9.1.4 Splittable Implies Tensorial

By a simple adaptation of an argument in [AJT19] for hypergraphs which built on [BRS11], we can use the splittable property to obtain tensorial properties for $W_G(k)$. More precisely, we can deduce Theorem 9.19.

Theorem 9.19 (Adapted from [AJT19])**.** *Suppose $W_G(\leq k)$ with $W_G(1) = [n]$ and an $(L+2k)$-local PSD ensemble $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ are given. There exist some universal constants $c_4 \geq 0$ and $C'' \geq 0$ satisfying the following: If $L \geq C'' \cdot (q^{4k} \cdot k^7 \cdot r/\mu^5)$, $\mathrm{Supp}(\mathbf{Z}_j) \leq q$ for all $j \in [n]$, and $W_G(\leq k)$ is $(c_4 \cdot (\mu/(4k \cdot q^k))^2, r)$-splittable, then*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \{\mathbf{Z}'_{w_1}\} \cdots \{\mathbf{Z}'_{w_k}\} \right\|_1 \leq \mu, \tag{13}$$

where $\mathbf{Z}'$ is as defined in Algorithm 6.5 on the input of $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ and Π_k.

Using Theorem 9.19, we can establish conditions on a γ-two-sided expander graph $G = (V, E, w)$ in order to ensure that $W_G(k)$ is (μ, L)-two-step tensorial.

Lemma 9.20 (Expander walks are two-step tensorial)**.** *There exist some universal constants $c' \geq 0$ and $C' \geq 0$ satisfying the following: If $L \geq c' \cdot (q^{4k} \cdot k^7/\mu^5)$, $\mathrm{Supp}(\mathbf{Z}_j) \leq q$ for all $j \in [n]$, and G is a γ-two-sided expander for $\gamma \leq C' \cdot \mu^2/\left(k^2 \cdot q^{2k}\right)$ and size $\geq k$, then $W_G(k)$ is (μ, L)-two-step tensorial.*

Proof. The proof is similar to the proof of Lemma 7.4 for HDXs, so we omit it. ∎

9.1.5 Interlude: Approximating k-CSP on Walk Constraints

Now, we digress to show how using Theorem 9.19 it is possible to deduce parameters for approximating k-CSPs on $W_G(k)$. We believe this result might be of independent interest and note that it is not required in the list decoding application.

Corollary 9.21. *Suppose $\mathfrak{I}$ is a q-ary k-CSP instance with constraints on $W_G(k)$. There exist absolute constants $C'' \geq 0$ and $c_4 \geq 0$ satisfying the following:*

If $W_G(k)$ is $(c_4 \cdot (\mu/(4k \cdot q^k))^2, r)$-splittable, then there is an algorithm that runs in time $n^{O\left(q^{4k} \cdot k^7 \cdot r/\mu^5\right)}$ based on $(C'' \cdot k^5 \cdot q^k \cdot r/\mu^4)$-levels of SOS-hierarchy and Algorithm 6.5 that outputs a random assignment $\xi : [n] \to [q]$ that in expectation ensures $\mathsf{SAT}_{\mathfrak{I}}(\xi) = \mathsf{OPT}(\mathfrak{I}) - \mu$.

Proof. The algorithm will just run Algorithm 6.5 on the local PSD-ensemble $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ given by the SDP relaxation of $\mathfrak{I}$ strengthened by $L = (C'' \cdot k^5 \cdot q^{2k}/\mu^4)$-levels of SOS-hierarchy and Π_k, where $C'' \geq 0$ is the constant from Theorem 9.19. $\mathbf{Z}$ satisfies

$$\mathsf{SDP}(\mathfrak{I}) = \mathop{\mathbb{E}}_{w \sim \Pi_k} \left[\mathop{\mathbb{E}}_{\{\mathbf{Z}_w\}} [\mathbf{1}[\mathbf{Z}_w \in \mathcal{P}_w]] \right] \geq \mathsf{OPT}(\mathfrak{I}). \tag{14}$$

Since the conditioning done on $\{\mathbf{Z}'\}$ is consistent with the local distribution, by law of total expectation and Eq. (14) we have

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim \Pi_k} \mathbf{1}[\mathbf{Z}'_w \in \mathcal{P}_w] = \mathsf{SDP}(\mathfrak{I}) \geq \mathsf{OPT}(\mathfrak{I}). \tag{15}$$

By Theorem 9.19 we know that

$$\mathop{\mathbb{E}}_{\Omega}\mathop{\mathbb{E}}_{w\sim\Pi_k}\left\|\{\mathbf{Z}'_w\}-\{\mathbf{Z}'_{w_1}\}\cdots\{\mathbf{Z}'_{w_k}\}\right\|_1\le\mu. \tag{16}$$

Now, the fraction of constraints satisfied by the algorithm in expectation is

$$\mathop{\mathbb{E}}_{\xi}[\mathsf{SAT}_{\mathcal{J}}(\xi)]=\mathop{\mathbb{E}}_{\Omega}\mathop{\mathbb{E}}_{w\sim\Pi_k}\mathop{\mathbb{E}}_{(\xi_1,\ldots,\xi_n)\sim\{\mathbf{Z}'_1\}\cdots\{\mathbf{Z}'_n\}}[\mathbf{1}[\xi|_w\in\mathcal{P}_w]].$$

By using Eq. (16), we can obtain

$$\mathop{\mathbb{E}}_{\xi}[\mathsf{SAT}_{\mathcal{J}}(\xi)]\ge\mathop{\mathbb{E}}_{\Omega}\left[\mathop{\mathbb{E}}_{\{\mathbf{Z}_w\}}\mathbf{1}[\mathbf{Z}'_w\text{ satisfies the constraint on }w]\right]-\mu.$$

Using Eq. (15), we conclude

$$\mathop{\mathbb{E}}_{\xi}[\mathsf{SAT}_{\mathcal{J}}(\xi)]\ge\mathsf{SDP}(\mathcal{J})-\mu=\mathsf{OPT}(\mathcal{J})-\mu.$$

■

9.2 Instantiation to Linear Base Codes

We instantiate the list decoding framework to the direct sum lifting given by the collection $W_G(k)$ of length k walks on a sufficiently expanding graph $G=(V,E,w)$. For parity sampling of expander walks, we will rely on the following fact.

Theorem 9.22 (Walks on Expanders are Parity Samplers [TS17] (Restatement of Theorem 4.1))**.** *Suppose G is a graph with second-largest eigenvalue in absolute value at most λ, and let $X(k)$ be the set of all walks of length k on G. Then $X(k)$ is a $(\beta_0,(\beta_0+2\lambda)^{\lfloor k/2\rfloor})$-parity sampler. In particular, for any $\beta>0$, if $\beta_0+2\lambda<1$ and k is sufficiently large, then $X(k)$ is a (β_0,β)-parity sampler.*

First, we instantiate the framework to linear codes which already encompasses most of the ideas need for general binary codes.

Lemma 9.23 (Direct sum lifting of linear biased codes II)**.** *Let $\varepsilon_0<1/2$ be a constant and $\varepsilon\in(0,\varepsilon_0)$. There exists a universal constant $C>0$ such that for any d-regular γ-two-sided expander graph G on ground set $W_G(1)=[n]$, if $\gamma\le\varepsilon^C$, then the following holds:*

For every binary $2\varepsilon_0$-biased linear code $\mathcal{C}_1$ on $W_G(1)=[n]$, there exists a 2ε-biased binary lifted linear code $\mathcal{C}_k=\mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ on $W_G(k)$ where $k=O(\log(1/\varepsilon))$ and

- *[Efficient List Decoding] If $\tilde{y}$ is $(1/2-\sqrt{\varepsilon})$-close to $\mathcal{C}_k$, then we can compute the list $\mathcal{L}(\tilde{y},\mathcal{C}_1,\mathcal{C}_k)$ (c.f. Definition 6.15) in time*
$$n^{\varepsilon^{-O(1)}}\cdot f(n),$$
where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}_1$.
- *[Rate] The rate r_k of $\mathcal{C}_k$ satisfies $r_k=r_1/d^{k-1}$ where r_1 is the relative rate of $\mathcal{C}_1$.*
- *[Linear] The lifted code $\mathcal{C}_k$ is linear.*

Proof. The proof is analogous to the one given in Lemma 7.5. We want to define parameters for a γ-two-sided expander $G = (V, E, w)$ so that $W_G(k)$ satisfies strong enough *robust* and *tensorial* assumptions and we can apply Theorem 6.17. In this application, we will rely on parity sampling for robustness. If $\mathrm{dsum}_{W_G(k)}$ is a $(2\varepsilon_0, 2\varepsilon)$-parity sampler, using the linearity of $\mathcal{C}_1$, we obtain a lifted code $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ which is linear and has bias 2ε; thus the lifting is indeed $(1/2 - \varepsilon_0, 1/2 - \varepsilon)$-robust. If we want to fully rely on parity sampling in Theorem 6.17, the lifting must be a $(\beta_0 = 1/2 + \varepsilon_0, \beta = 2\varepsilon)$-parity sampler, which is more stringent than the first parity sampling requirement. [21] To invoke Theorem 9.22 and obtain this (β_0, β)-parity sampler, we need to choose a parameter θ (where $0 < \theta < (1 - \beta_0)/\beta_0$) such that

$$k \geq 2 \cdot \log_{(1+\theta)\beta_0}(\beta) + 2 \text{ and}$$
$$\gamma \leq \frac{\theta \cdot \beta_0}{2},$$

which will ensure that

$$(\beta_0 + 2\gamma)^{\lfloor k/2 \rfloor} \leq ((1+\theta)\beta_0)^{\lfloor k/2 \rfloor} \leq \beta.$$

To get a (μ, L)-tensorial collection of walks, Lemma 9.20 requires

$$L \geq \frac{c' \cdot 2^{4k} \cdot k^7}{\mu^5} \quad \text{and} \quad \gamma \leq \frac{C' \cdot \mu^2}{k^2 \cdot 2^{2k}}.$$

where we used that our alphabet is binary (i.e., $q = 2$) and $c', C' > 0$ are constants. Finally, Theorem 6.17 requires $\mu \leq \varepsilon^8/2^{22}$. The conceptual part of the proof is essentially complete and we are left to compute parameters. We choose $\theta = 1/2 - \varepsilon_0$, so that provided $\varepsilon_0 < 1/2$ we have $(1+\theta)\beta_0 = 3/4 + \varepsilon_0 - \varepsilon_0^2 < 1$. Combining the parity sampling and tensorial requirements and after some simplification, the expansion γ is constrained as

$$\gamma \leq C'' \cdot \min\left(\frac{\varepsilon^{16}}{k^2 \cdot 2^{2k}}, \left(1/4 - \varepsilon_0^2\right)\right),$$

where $C'' > 0$ is a constant. We deduce that taking γ as

$$\gamma \leq C'' \cdot \frac{\left(1/4 - \varepsilon_0^2\right) \cdot \varepsilon^{16}}{k^2 \cdot 2^{2k}}$$

is sufficient. Further simplifying the above bound gives γ as in the statement of the theorem. Now, we turn to the SOS related parameter L which is constrained to be

$$L \geq c'' \cdot \frac{2^{4k} \cdot k^7}{\varepsilon^{40}},$$

where $c'' > 0$. Note that in this case the exponent $O(L + k)$ appearing in the running time of Theorem 6.17 becomes $O(L)$. Further simplification of the bound on L leads to a running time of $n^{\varepsilon^{-O(1)}} \cdot f(n)$ as in the statement of the theorem. ∎

9.3 Instantiation to General Base Codes

The proof of Theorem 9.2 follows from Lemma 9.23 in the same way that Theorem 7.7 follows from Lemma 7.5 in the case of HDXs.

[21] Recall that this strengthening is used in our list decoding framework.

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, 6, 41

[AJT19] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In *Proceedings of the 60th IEEE Symposium on Foundations of Computer Science*, pages 180–201, 2019. 2, 3, 9, 11, 12, 19, 23, 37, 38, 47, 48, 50, 51, 53

[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. 8

[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. 8

[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. 7, 11, 23, 37, 38, 47, 51, 53

[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, 3, 4, 13, 15, 41, 45

[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, 12, 13, 14

[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

[GM12] Bernd Gärtner and Jiri Matousek. *Approximation Algorithms and Semidefinite Programming*. Applications of Mathematics. Springer-Verlag Berlin Heidelberg, 2012. 21

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95-50, Electronic Colloquium on Computational Complexity, 1995. 6

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001. 11, 24

[GRS19] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. Available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/index.html, 2019. 26

[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. 4, 21

[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. 11, 24

[LPS88] Alexander Lubotzky, R. Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988. 3, 46

[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, 13

[LSV05b] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Ramanujan complexes of typeãd. *Israel Journal of Mathematics*, 149(1):267–299, Dec 2005. 3, 13

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

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2017. 58

[RW17] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 21

[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. 4, 21

[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, 6

[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, 6, 11, 12, 17, 46, 54

A Auxiliary Basic Facts of Probability

In this section, we collect some basic facts of probability used in the text.

Fact A.1 (First Moment Bound)**.** *Let* $\mathbf{R}$ *be a random variable in* $[0,1]$ *with* $\mathbb{E}[\mathbf{R}] = \alpha$*. Let* $\beta \in (0,1)$ *be an arbitrary approximation parameter. Then*

$$\mathbb{P}[\mathbf{R} \geq (1-\beta)\cdot\alpha] \geq \beta\cdot\alpha.$$

In particular,

$$\mathbb{P}\left[\mathbf{R} \geq \frac{\alpha}{2}\right] \geq \frac{\alpha}{2}.$$

Fact A.2 (Chernoff Bound [MU17])**.** *Let* $\mathbf{R}_1,\ldots,\mathbf{R}_n$ *be independent and identically distributed random variables where* $\mathbf{R}_i$ *is uniformly distributed on* $\{\pm 1\}$*. For every* $a > 0$*,*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n}\mathbf{R}_i\right| \geq a\right] \leq 2\cdot\exp\left(-\frac{a^2}{2n}\right).$$

Fact A.3 (Hoeffding Bound [MU17])**.** *Let* $\mathbf{R}_1,\ldots,\mathbf{R}_n$ *be independent random variables such that* $\mathbb{E}[\mathbf{R}_i] = \mu$ *and* $\mathbb{P}[a \leq \mathbf{R}_i \leq b] = 1$ *for* $i \in [n]$*. For every* $\beta > 0$*,*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{R}_i - \mu\right| \geq \beta\right] \leq 2\cdot\exp\left(-\frac{2\cdot\beta^2\cdot n}{(a-b)^2}\right).$$

B Further Properties of Liftings

We show that a uniformly random odd function $g\colon \{\pm 1\}^k \to \{\pm 1\}$ yields a parity lifting w.v.h.p. in k. Thus, parity liftings abound and we are not restricted to k-XOR in the framework. In fact, SOS abstracts the specific combinatorial properties of the lifting function being able to handle them in a unified way.

Lemma B.1. *Let $k \in \mathbb{N}^+$ be odd. For every $p, \beta, \theta > 0$ satisfying $\theta \geq \sqrt{\log(2/\beta)}/\sqrt{pk}$,*

$$\mathop{\mathbb{P}}_{g}\left[\left|\mathbb{E}_{x\sim \mathrm{Bern}(p)^{\otimes k}}\left[g(\chi^{\otimes k}(x))\right]\right| \geq \beta\right] \leq 2\cdot k \cdot \exp\left(-\beta^2 \cdot \binom{k}{\lfloor (1-\theta)pk\rfloor}/8\right),$$

where $g\colon \{\pm 1\}^k \to \{\pm 1\}$ is a uniformly random odd function and $\chi\colon (\mathbb{F}_2, +) \to (\{\pm 1\}, \cdot)$ is the non-trivial character.

Proof. It is enough to consider $p \in (0, 1/2]$ since the case $p \in [1/2, 1)$ can be reduced to the current case by taking the complement of the bit strings appearing in this analysis. Applying the Hoeffding bound Fact A.3 yields

$$\begin{aligned}\mathbb{E}_{x\sim \mathrm{Bern}(p)^{\otimes k}}\left[g(x)\right] &= \mathbb{E}_{w\sim \mathrm{Binom}(k,p)}\left[g(\chi^{\otimes k}(x))\mathbf{1}_{w\in[pk\pm C\cdot pk]}\right] \pm 2\cdot\exp(-C^2)\\ &= \mathbb{E}_{w\sim \mathrm{Binom}(k,p)}\left[g(\chi^{\otimes k}(x))\mathbf{1}_{w\in[pk\pm C\cdot pk]}\right] \pm \frac{\beta}{2},\end{aligned}$$

where the last equality follows from choosing $C = \theta\sqrt{pk}$ and the assumption that $\theta \geq \sqrt{\log(2/\beta)}/\sqrt{pk}$.

Since $p \leq 1/2$, $\ell = \binom{k}{\lfloor(1-\theta)\cdot p\cdot k\rfloor}$ is a lower bound on the number of binary strings of the Boolean k-hypercube in a single layer of Hamming weight in the interval $[pk \pm C\cdot pk]$. A second application of the Hoeffding bound Fact A.3 gives that the bias within this layer is

$$\mathop{\mathbb{P}}_{g}\left[\left|\mathbb{E}_{x\in\mathbb{F}_2^k\colon \|x\|=\ell}\left[g(\chi^{\otimes k}(x))\right]\right| \geq \beta/2\right] \leq 2\cdot\exp\left(\beta^2\cdot\ell/8\right).$$

By union bound over the layers the result follows. ∎

C Derandomization

We show how to derandomize the list decoding framework (which amounts to derandomize Algorithm 6.29) when the lifting function is a parity sampler and it satisfies a bounded degree condition (cf Eq. (17)). We observe that this is the setting of our two concrete instantiations, namely, for HDXs and expander walks. In the former case, we work with D-flat distributions and in the latter case with walk length and graph degree that are both functions of ε. Roughly speaking, we show that replacing a random sample by the majority works as long as parity sampling is sufficiently strong.

Lemma C.1 (Majority Word)**.** *Let $z^* \in \{\pm 1\}^{X(1)}$ where $X(1) = [n]$. Suppose that $y^* = \mathrm{lift}_{X(k)}(z^*)$ satisfy*

$$\mathbb{E}_{z\sim\{\mathbf{z}^{\otimes}|_{(S,\sigma)}\}}\left[\left|\mathbb{E}_{\mathfrak{s}\sim\Pi_k} y^*_{\mathfrak{s}}\cdot \mathrm{lift}(z)_{\mathfrak{s}}\right|\right] \geq 3\cdot\varepsilon,$$

and

$$\mathop{\mathbb{P}}_{\mathfrak{s}\sim\Pi_k}[\mathfrak{s} \ni i] \leq \frac{g(\varepsilon)}{n}. \tag{17}$$

If also $\mathrm{lift}_{X(k)}$ *is a* $(1-\xi, 2\varepsilon)$*-parity sampler for some* $\xi \in (0,1)$, $\xi \geq 2\exp\left(-C\cdot\varepsilon^2\cdot g(\varepsilon)^2\cdot n\right) = o_n(1)$ *where* $C > 0$ *is an universal constant and* $\xi \geq 1/(n(1-\xi-o_n(1)))$*, then*

$$\left|\mathbb{E}_{i\in[n]} z_i^*\cdot z_i'\right| \geq 1 - 7\sqrt{\xi},$$

where $z' \in \{\pm 1\}^n$ *is the majority defined as* $z_i' = \mathrm{argmax}_{b\in\{\pm1\}}\Pr_{\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}[\mathbf{Z}_i = b]$.

Proof. Define $f(z) := |\mathbb{E}_{\mathfrak{s}\sim\Pi_k} y_{\mathfrak{s}}^*\cdot \mathrm{lift}(z)_{\mathfrak{s}}|$. Then, using Eq. (17) we claim that $f(z)$ is $O(g(\varepsilon)/n)$-Lipschitz with respect to ℓ_1 since

$$|f(z) - f(\tilde{z})| \leq \sum_{i\in X(1)} 2\cdot \mathop{\mathbb{P}}_{\mathfrak{s}\sim\Pi_k}[\mathfrak{s}\ni i]\cdot|z_i - \tilde{z}_i| \leq O\left(\frac{g(\varepsilon)}{n}\right)\cdot\|z-\tilde{z}\|_1.$$

Since the underlying distribution of $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}$ is a product distribution on $\{\pm1\}^n$ and f is $O(g(\varepsilon)/n)$-Lipschitz, applying Hoeffding's inequality yields

$$\mathop{\mathbb{P}}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}[f(z)\leq\varepsilon] \leq \mathop{\mathbb{P}}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[\left|f(z) - \mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}} f(z)\right| \geq \varepsilon\right] \leq \exp\left(-\Theta(g'(\varepsilon)\cdot n)\right),$$

where $g'(\varepsilon) = \varepsilon^2\cdot g(\varepsilon)^2$.

Using the assumption that lift is a $(1-\xi, 2\varepsilon)$-parity sampler, we obtain

$$\mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[|\langle z^*, z\rangle|\right] \geq 1 - \xi - 2\exp\left(-\Theta(g'(\varepsilon)\cdot n)\right).$$

By Jensen's inequality,

$$\mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[\langle z^*, z\rangle^2\right] \geq \left(\mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[|\langle z^*, z\rangle|\right]\right)^2 \geq (1-\xi-2\exp\left(-\Theta(g'(\varepsilon)\cdot n)\right))^2.$$

Using indepdence, we get

$$\mathbb{E}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}\left[\mathbb{E}_{i,j\in[n]} z_i^* z_i z_j z_j^*\right] \leq \mathbb{E}_{i,j\in[n]} z_i^*\mathbb{E}\left[z_i\right]\mathbb{E}\left[z_j\right] z_j^* + \frac{1}{n} = \left(\mathbb{E}_{i\in[n]} z_i^*\mathbb{E}\left[z_i\right]\right)^2 + \frac{1}{n}.$$

Thus, in particular $\left|\mathbb{E}_{i\in[n]} z_i^*\mathbb{E}\left[z_i\right]\right| \geq (1-\xi-o_n(1)) - 1/((1-\xi-o_n(1))n) \geq 1 - 3\xi$ which implies

$$\begin{aligned} 1 - 3\xi &\leq \left|\mathbb{E}_{i\in[n]} z_i^*\left(\Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = 1] - \Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = -1]\right)\right| \\ &\leq \mathbb{E}_{i\in[n]}\left|\Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = 1] - \Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = -1]\right|. \end{aligned}$$

Since

$$\mathbb{E}_{i\in[n]} 1 - \left|\Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = 1] - \Pr_{|_{(S,\sigma)}}[\mathbf{Z}_i = -1]\right| \leq 3\xi,$$

Markov's inequality yields

$$\mathop{\mathbb{P}}_{i\in[n]}\left[1-\sqrt{\xi}\geq\left|\Pr_{|_{(S,\sigma)}}\left[\mathbf{Z}_i=1\right]-\Pr_{|_{(S,\sigma)}}\left[\mathbf{Z}_i=-1\right]\right|\right]\leq 3\sqrt{\xi}.$$

Now, let $z'\in\{\pm 1\}^n$ be as in the statement of the lemma. Then,

$$1-3\xi-4\sqrt{\xi}\leq\left|\mathbb{E}_{i\in[n]}z_i^*\cdot z_i'\right|.$$

Hence, we conclude that $\left|\mathbb{E}_{i\in[n]}z_i^*\cdot z_i'\right|\geq 1-7\sqrt{\xi}$. ■

Remark C.2. *The parity sampling requierment might be slightly stronger with this derandomized version but it does not change the asymptotic nature of our results. More precisely, we are only asking for $(1-\xi,2\varepsilon)$-parity sampler for a different constant value $\xi>0$.*