THE UNIVERSITY OF CHICAGO

A CONSTRAINED RANDOM WALK THROUGH CODING THEORY

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

FERNANDO GRANHA JERONIMO

CHICAGO, ILLINOIS

JULY 2021

To my wife, Renata, for her love and unwavering support.

To my parents for their love and support.

“Education is not preparation for life; education is life itself.” — John Dewey

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I thank my wife for strongly supporting me in this decision of becoming a scientist and seeing in science as much value and beauty as I do. I thank my parents for imparting to me the importance of education and for also supporting me in this decision of becoming a scientist.

I thank my truly amazing advisor, Madhur Tulsiani, for embarking on this journey with me. Besides his broad knowledge and interest for everything, he is an amazing person and has been extremely supportive. Madhur makes it feel that no topic is inaccessible. I can not imagine a best PhD advisor. I thank Arnaldo Vieira Moura for introducing me to theoretical computer science and helping me at several stages. Without Arnaldo I would not be in this fascinating field now. I thank Irit Dinur for her support at an important step in my path and for being an inspiration through her research with many beautiful results involving *expansion*. I thank Thomas Vidick for an internship opportunity and for believing that anyone can grow as researcher.

I thank Leonardo Nagami Coregliano for being a brave and resilient collaborator in attacking all types of problems with me. I thank Vedat Levi Alev for the fun times we fought with swap walks and CSPs and for sharing his love for spectral graph theory. I thank Mrinalkanti Ghosh for the generous hospitality in receiving me in Madhur's group and for many discussions and lunches together. I thank Dylan Quintana and Shashank Srivastava for their friendly and kind nature during many hours of collaboration. I thank Chris Jones and Goutham Rajandran for all the discussions about Sum-of-Squares (and to Chris also to discussions about codes). I thank Tushant Mittal for many discussions about the structure of quantum codes and for his great enthusiasm for theory (at some point giving me the extra energy to start a reading group). I thank Aaron Potechin for making the Sum-of-Squares hierarchy seem a natural and intuitive object. I thank Akash Kumar for being a frequent visitor bringing his love for the field and also some jokes.

I thank Janos Simon for giving me the opportunity to come here and for accommodating my distaste for bureaucracies in a very generous way. Here, I had the pleasure of encountering many more friendly and enthusiastic students (Hing Yin Tsang, Aritra Sen and the list is too long to include all their names). I also encountered many professors deeply passionate about their work who kindly shared their knowledge and sometimes even their unique style of doing science. I will carry fond memories of courses by László Babai, Julia Chuzhoy, Andrew Drucker, Yury Makarychev, Ketan Mulmuley, Aaron Potechin, Alexander Razborov and Madhur Tulsiani.

I thank the theoretical computer science community as a whole for creating a welcoming and vibrant environment. I greatly benefited from the Simons cluster on high-dimensional expanders and codes organized by Irit Dinur and Prahladh Harsha. I also thank Prasad Raghavendra for extending a hand during an application showing how supportive this community is.

A special thank you to my committee members Aaron Potechin, Janos Simon and Madhur Tulsiani for being generous with their time and being very flexible with their schedule. I also thank the hard work of the UChicago and TTIC staff for their prompt help in all sorts of tasks.

This disseration is based on the following papers.

- Chapter 2 is based on the paper "Approximating Constraint Satisfaction Problems on High-Dimensional Expanders" [FOCS 2019] joint work with Vedat Levi Alev and Madhur Tulsiani.

- Chapter 3 is based on the paper "List Decoding of Direct Sum Codes" [SODA 2020] joint work with Vedat Levi Alev, Dylan Quintana, Shashank Srivastava and Madhur Tulsiani.

- Chapter 4 is based on the paper "Unique Decoding of Explicit ε-balanced Codes

Near the Gilbert–Varshamov Bound" [FOCS 2020] joint work with Dylan Quintana, Shashank Srivastava and Madhur Tulsiani.

- Chapter 5 is based on the paper "Near-Linear Time Decoding of Ta-Shma's Codes via Splittable Regularity" [STOC 2021] joint work with Shashank Srivastava and Madhur Tulsiani.

ABSTRACT

We investigate some problems involving optimization, expansion, coding theory and pseudorandomness establishing some new connections among these fields.

Our first result is an approximation algorithm for constraint satisfaction problems (CSPs), where constraints are placed on the edges of "expanding" hypergraphs. This result (builds on and) generalizes known algorithms for graphs. Our algorithm is based on the Sum-of-Squares semi-definite programming hierarchy and it is a natural higher-order generalization of the graph case.

Next, we observe that the task of decoding some well-known families of distance amplified codes using expanding structures can be reduced to approximating suitable "expanding" CSPs. By enhancing the Sum-of-Squares hierarchy with an "entropic" potential, we can, roughly speaking, obtain list decoding guarantees out of unique decoding. In this way, we obtain a new list decoding framework which is our second result.

In a breakthrough work, Ta-Shma [STOC 2017] found the first explicit family of ε-balanced binary codes near the so-called Gilbert–Varshamov bound (hence achieving nearly optimal distance versus rate trade-off). Put it simply, Ta-Shma's codes are distance amplified codes using carefully desinged expanding structures. We obtain our third result by overcoming the far from optimal trade-offs of our list decoding framework allowing us to provide the first polynomial time decoder for Ta-Shma's codes.

Finally, using pseudorandomness techniques based on our new weak regularity decomposition of sparse tensors (supported on expanding structures) we can also approximate the CSPs arising in the decoding tasks mentioned above. Thanks to the much faster computational time of finding weak regularity decompositions compared to solving Sum-of-Squares programs, we obtain our fourth result: a near-linear time decoder for Ta-Shma's codes.

CHAPTER 1

INTRODUCTION

A central concept in our work is *expansion*. In theoretical computer science (TCS) and mathematics, expansion comes in a few flavors with the most notable example, perhaps, being expander graphs [HLW06]. Roughly speaking, expander graphs are graphs mimicking some of the properties of complete graphs such as being well-connected. Fortunately, contrary to complete graphs which have as many edges as possible, exapnder graphs can have much fewer edges and yet still be well-connected. By combining these opposing properties of sparseness and well-connectedness, expanders found a myriad of applications in TCS and this wide applicability continues to grow.

One way to extend the theory of expander graphs is to consider notions of expanding hypergraphs. In this direction, a theory of high-dimensional expnaders (HDXs) has recently emerged [Lub18]. Given the extra richness of hypergraphs compared to their one dimensional counterparts (graphs), the very definition of hypergraph expansion seems trickier and some (not necessarily equivalent) definitions were proposed [LM06, KKL14, EK16a, DK17]. This represents a departure from the expansion of graphs, where different definitions amount to morally the same notion of expansion[1] (be it isoperimetric, algebraic or in terms of mixing time of random walks).

Another point of departure between graphs and hypergraphs appears in the study of constraint satisfaction problems (CSPs). CSPs whose constraints lie on the edges of sparse random regular graphs admit non-trivial approximation algorithms[2] [BRS11, GS11, OGT15]. However, CSPs whose constraints lie on hyperedges of sparse random hypergraphs (even of hyperedge size 3) are believed to be hard to approximate efficiently. In fact, the later

1. This in terms of notions related to edge expansion. Vertex expansion of graphs is indeed a different notion.

2. Random d-regular graphs are expanding (even near-Ramanujan) with high probability [Fri91].

kind of CSP cannot be non-trivially approximated using powerful algorithmic techniques, e.g., [Gri01, Sch08, KMOW17].

A natural question emerges: how well can we approximate CSPs whose constraints lie on the hyperedges of (sparse) high-dimensional expanders? Given that CSPs on sparse random hypergraphs are believed to be hard to approximate, it would be conceivable to expect that CSPs on sparse HDXs are similarly hard. In our first result in this work [AJT19], we show that this intuition is false when the constraints are placed on the hyperedges of a sufficiently expanding HDX (and the variables correspond to vertices). Our approximation algorithm is a natural higher-order generalization of known algorithms for CSPs on graphs by Barak, Raghavendra and Steurer [BRS11] and Guruswami and Sinop [GS11], which are based on a convex optimization hierarchy called Sum-of-Squares semi-definite programming hierarchy.

To analyze the quality of approximation of our algorithm, we are led to consider (higher-order) versions of specific random walks in a hypergraph. As long as the operators corresponding to these walks are expanding (i.e., these walks mix fast), our algorithm can provide good additive approximation to the optimum value of the CSP. This expansion requirement can be seen as giving rise to another notion of high-dimensional expansion which we dub *splittability*. Curiously, the walk operators considered in our work are of the same as the operators independently considered for *agreement tests* [DD19], which is an important primitive in the construction of probabilistic check proofs (PCPs). This notion of expansion (splittability) rediscovered in our CSP work actually first appeared in another context [Mos10].

Having the ability to efficiently approximate CSPs on (some) *expanding* hypergraphs will turn out to be extremely useful to us, and it underlies most of our subsequent results in this work. Of course, the alternative of having hard to approximate CSPs could have been interesting and valuable in itself. We point out that, by placing the constraints and

variables differently on a HDX, explicit hardness results for the Sum-of-Squares hierarchy were later obtained in [DFHT21].

We mentioned that expansion has found a myriad of applications in TCS and one particularly fertile field of this phenomenon has been coding theory [Gur04]. Two major uses of expansion in coding theory are: (i) in distance amplification of a base code (e.g., [GI05, TS17]), and (ii) in the design of parity check matrices (e.g., [Gal62, SS96]). In this work, we investigate the former case and show that the ability to approximate CSPs on expanding (splittable) structures can be leveraged to provide efficient decoding algorithms.

Our second result [AJQ+20] is a *list decoding* framework for distance amplified codes using expanding structures: HDXs and walks over expander graphs. List decoding is a relaxed decoding regime where one seeks to correct a large fraction of adversarial errors at the expense of having not just one but possibly a small list of codewords. It was introduced by Elias [Eli57] and it has more recently become a central primitive in coding theory thanks to the seminal efficient list decoding work of Sudan [Sud97] and Guruswami and Sudan [GS98]. Surprisingly, list decoding can be useful even in performing unique decoding.

A key observation is that the problem of *unique decoding* distance amplified codes using expanding structures (considered in this work) can be reduced to approximating CSPs whose constraints lie on the hyperedges of suitable hypergraphs. To perform list decoding, we can then use as starting point our Sum-of-Squares based CSP approximation algorithm for expanding hypergraphs. However, we will need to recover not only one approximate solution as in unique decoding, but a few (far apart) solutions to be able to perform list decoding. For this reason, we add to the Sum-of-Squares program an "entropic" potential function in order to make the (near) optimum solution of this convex program sufficiently rich to "encode" all the solutions we need to retrieve. This technique

of using an entropic potential in Sum-of-Squares was independently used in robust statistics [KKK19, RY20]. Whereas their results are in a continuous setting over the reals (i.e., $\mathbb{R}$), our results are over discrete (finite) alphabets.

In coding theory, special attention is given to the trade-off between the distance of a code and its rate. Naturally, when using a code we want to pay as little as possible in terms of added redundancy to be able to recover from a given fraction of adversarial errors. Despite their fundamental nature, it is surprising that simple questions about the achievable parameters and the very construction of (nearly) optimum codes remain open specially over small alphabets. For binary codes, a precise trade-off between these parameters remains a major elusive open problem.

The best existential distance versus rate trade-offs was shown by Gilbert [Gil52] for arbitrary codes and it was later shown for random linear codes by Varshamov [Var57]. Unfortunately, these results are merely existential not yielding an explicit efficient way to build these codes. The combinition of distance and rate achieved by their results is widely known as the Gilbert–Varshamov (GV) bound. The importance of this bound is that it is not too far from optimal as shown by the two linear programming (LP) bounds of McEliece, Rodemich, Rumsey and Welch [MRRW77]. For instance, in the large distance regime for binary codes, namely, distance $1/2 - \varepsilon$, the GV bound establishes the existence of codes of rate $\Omega(\varepsilon^2)$ whereas one of the LP bounds states that $O(\varepsilon^2 \log(1/\varepsilon))$ is an upper bound on the rate of any such code.

In a breakthrough work, Ta-Shma [TS17] gave the first explicit construction of a binary code near the GV bound. More precisely, Ta-Shma constructed linear binary codes of distance $1/2 - \varepsilon/2$ and rate $\Omega(\varepsilon^{2+o(1)})$. Ta-Shma's codes have the additional property of being ε-balanced, i.e., the hamming weight of every non-zero codeword is at least $1/2 - \varepsilon/2$ and at most $1/2 + \varepsilon/2$. This gives rise to ε-biased distributions of nearly optimum support size which is an important object in the theory of pseudorandomness. It was left

open whether Ta-Shma's codes admit efficient decoding. Note that efficient decoding is not guaranteed since there are families of codes that are hard to decode.

Our third result in this work [JQST20] is a positive answer that Ta-Shma's codes do admit an efficient (polynomial time) decoding algorithm. Before we discuss our decoding algorithms, let's first give a high-level overview of Ta-Shma's construction so that it might be clear how it fits in our story so far. Ta-Shma's codes are distance amplified codes using a carefully constructed expanding structure. To be more specific, the expanding structure considered by Ta-Shma is the collection of walks of a fixed small length on the wide replacement product of two expander graphs [BATS08], which is a generalization of the the celebrated Zig-Zag product of Reingold, Vadhan and Wigderson [RVW00]. This special collection of walks can be seen as a derandomization of the collections of all walks of the same length on a single expander graph, which was known to achieve distance $1/2 - \varepsilon$ but rate $\Omega(\varepsilon^{4+o(1)})$.

Our list decoding framework can be applied to Ta-Shma's codes after simple modifications and some observations. Unfortunately, at least naively, this application can only be done after drastically reducing the rate to $\Theta(2^{-\mathrm{polylog}(1/\varepsilon)})$. In our third result, we have to bring extra techniques to operate on the nearly optimal regime of distance $1/2 - \varepsilon/2$ and rate $\Omega(\varepsilon^{2+o(1)})$. One such technique is to use list decoding to perform unique decoding since in our case the various parameter dependencies become much more favorable. The second technique is to create a sequence of codes (code cascading), where list decoding takes place between consecutive codes. Carefully combining these techniques, we then obtain the first polynomial time decoding algorithm for Ta-Shma's codes.

One downside of our decoding algorithm for Ta-Shma's codes is that albeit the running time being polynomial, it actually requires large degrees (in some regimes even depending on $1/\varepsilon$). In our fourth result in this work [JST21], we provide a near-linear time

(in the block length) decoder for Ta-Shma's code. One reason the algorithms of our third result our slow is that they are based on the Sum-of-Squares hierarchy. The hierarchy is known to be very powerful, but to also require substantial running time. For this reason, in our fourth result we adopt a completely different approach based on weak regularity decompositions as pioneered by Frieze and Kannan [FK98] but for sparse tensor supported on expanding structures. We show that computing these regularity decompositions can be done in near-linear time. This later approach is arguably simpler and more intuitive.

1.1 Chapter Outlines

As we alluded previously, we investigate problems involving optimization, expansion, coding theory and pseudorandomness establishing some connections along way. The work in this dissertation is based on joint results with Vedat Alev, Dylan Quintana, Shashank Srivastava and Madhur Tulsiani (see acknowledgments for more details). Next, we provide brief outlines for each subsequent chapter.

- **Approximating k-CSPs on expanding structures:** In Chapter 2, we give polynomial time approximation algorithms for k-CSPs (Constraint Satisfaction Problems) on suitably [3] expanding hypergraphs, which is a class of structures containing high-dimensional expanders (as in Dinur and Kaufman definition [DK17]) as an important special case. Naturally, the quality of approximation crucially depends on quality of expansion of these hypergraphs. Our algorithmic results are based on the SOS hierarchy and generalize the 2-CSPs results of [BRS11, GS11]. Via known connections, our algorithms translate into approximation algorithms for quantum k-CSPs,

3. More precisely, we generalize to hypergraphs the notion of threshold rank of a graph [BRS11] which is a robust version of expansion tolerating a few,i.e., $O(1)$, large eigenvalues (the rank) in the adjacency matrix of a graph.

the so-called k-local Quantum Hamiltonians.

- **List decoding framework for binary codes:** In Chapter 3, we provide a *list decoding* framework for distance amplified codes based on expanding structures: high-dimensional expanders (as in Dinur and Kaufman definition [DK17]) and walks on expander graphs. We view the problem of *unique decoding* as solving a suitable Max k-CSP (Constraint Satisfaction Problem) instance, which can be solved using our earlier work [AJT19] based on the SOS hierarchy. To obtain the list decoding framework, we maximize an entropic proxy while solving a k-CSP. This makes the SOS solution rich enough so that we can "recover" a list of all the desired codewords from it. Despite this parameter shortcoming, this framework served as our starting point for *unique decoding* results of nearly optimal codes mentioned next.

- **Unique decoding near optimal binary codes:** In Chapter 4, we give polynomial time *unique decoding* algorithms for nearly optimal, in terms of redundancy (rate) versus robustness (distance) trade-off (i.e., near the so-called Gilbert–Varshamov bound [Gil52, Var57]), explicit **binary** codes of large distance. These codes are (essentially) those explicit binary codes of distance $1/2 - \varepsilon$ and rate $\Omega(\varepsilon^{2+o(1)})$ arising from the breakthrough construction of Ta-Shma [TS17]. Our algorithms are based on Sum-of-Squares hierarchy and use as a starting point a decoding framework from our earlier work [AJQ+20] discussed in Chapter 3. Our main contribution consists in overcoming the far from optimal rates from [AJQ+20] to operate in this near optimal regime for unique decoding. This result can be seen as a step towards a better understanding of the elusive case of binary codes in the general adversarial error model of Hamming [Ham50]. These algorithms are just a proof of concept showing that polynomial time algorithms exist in this previously unattained regime.

- **Near-linear time decoding near optimal binary codes:** In Chapter 5, this previous

result of decoding Ta-Shma's codes opened avenues for a near-linear time unique algorithms using a novel algorithmic weak regularity decomposition in the style of Frieze and Kannan [FK96] but for sparse tensors supported on expending hypergraphs. With this new weak regularity decomposition, we can approximate the k-CSPs, decode some distance amplified codes and even perform list decoding. We hope that these techniques can also open avenues to list decoding algorithms with near optimal parameters, this being a major open problem in the field [Gur09, Gur10].

CHAPTER 2

APPROXIMATING CSPS ON HIGH-DIMENSIONAL EXPANDERS

2.1 Introduction

We consider the problem of approximately solving constraint satisfaction problems (CSPs) on instances satisfying certain expansion properties. The role of expansion in understanding the approximability of CSPs with two variables in each constraint (2-CSPs) has been extensively studied and has led to several results, which can also be viewed as no-go results for PCP constructions (since PCPs are hard instances of CSPs). It was shown by Arora et al. [AKK+08] (and strengthened by Makarychev and Makarychev [MM11]) that the Unique Games problem is easily approximable on expanding instances, thus proving that the Unique Games Conjecture of Khot [Kho02] cannot be true for expanding instances. Their results were extended to all 2-CSPs and several partitioning problems in works by Barak, Raghavendra and Steurer [BRS11], Guruswami and Sinop [GS11], and Oveis Gharan and Trevisan [OGT15] under much weaker notions of expansion.

We consider the following question:

When are expanding instances of k-CSPs easy for $k > 2$?

At first glance, the question does not make much sense, since random instances of k-CSPs (which are also highly expanding) are known to be hard for various models of computation (see [KMOW17] for an excellent survey). However, while the kind of expansion exhibited by random instances of CSPs is useful for constructing codes, it is not sufficient for constructing primitives for PCPs, such as locally testable codes [BSHR05]. On the other hand, objects such as high-dimensional expanders, which possess a form of "structured multi-scale expansion" have been useful in constructing derandomized direct-product and direct-sum tests (which can be viewed as locally testable distance

amplification codes) [DK17], lattices with large distance [KM18], list-decodable direct product codes [DHK+18], and are thought to be intimately connected with PCPs [DK17]. Thus, from the PCP perspective, it is more relevant to ask if this form of expansion can be used to efficiently approximate constraint satisfaction problems.

Connections to coding theory. Algorithmic results related to expanding CSPs are also relevant for the problem of *decoding* locally testable codes. Consider a code C constructed via k-local operations (such as k-fold direct-sum) on a base code C_0 with smaller distance. Then, a codeword in C is simply an instance of a CSP, where each bit places a constraint on k bits (which is k-XOR in case of direct sum) of the relevant codeword in C_0. The task of decoding a noisy codeword is then equivalent to finding an assignment in C_0, satisfying the maximum number of constraints for the above instance. Thus, algorithms for solving CSPs on expanding instances may lead to new decoding algorithms for codes obtained by applying local operations to a base code. In fact, the list decoding algorithm for direct-product codes by Dinur et al. [DHK+18] also relied on algorithmic results for expanding unique games. Since all constructions of locally testable codes need to have at least some weak expansion [DK12], it is interesting to understand what notions of expansion are amenable to algorithmic techniques.

High-dimensional expanders and our results. A d-dimensional expander is a downward-closed hypergraph (simplicial complex), say X, with edges of size at most $d+1$, such that for every hyperedge $\mathfrak{a} \in X$ (with $|\mathfrak{a}| \leq d-1$), a certain "neighborhood graph" $G(X_\mathfrak{a})$ is a spectral expander[1]. Here, the graph $G(X_\mathfrak{a})$ is defined to have the vertex set $\{i \mid \mathfrak{a} \cup \{i\} \in X\}$ and edge-set $\{i,j \mid \mathfrak{a} \cup \{i,j\} \in X\}$. If the (normalized) second singular value of each of the neighborhood graphs is bounded by γ, X is said to be a γ-high-

1. While there are several definitions of high-dimensional expanders, we consider the one by Dinur and Kaufman [DK17], which is most closely related to spectral expansion, and was also the one shown to be related to PCP applications. Our results also work for a weaker but more technical definition by Dikstein et al. [DDFH18], which we defer till later.

dimensional expander (γ-HDX).

Note that (the downward closure of) a random sparse $(d+1)$-uniform hypergraph, say with n vertices and $c \cdot n$ edges, is very unlikely to be a d-dimensional expander. With high probability, no two hyperedges share more than one vertex and thus for any $i \in [n]$, the neighborhood graph G_i is simply a disjoint union of cliques of size d, which is very far from an expander. While random hypergraphs do not yield high-dimensional expanders, such objects are indeed known to exists via (surprising) algebraic constructions [LSV05b, LSV05a, KO18a, CTZ18] and are known to have several interesting properties and applications [KKL16, DHK+18, KM17, KO18b, DDFH18, DK17, PRT16].

Expander graphs can simply be thought of as the one-dimensional case of the above definition. The results of Barak, Raghavendra and Steurer [BRS11] for 2-CSPs yield that if the constraint graph of a 2-CSP instance (with size n and alphabet size q) is a sufficiently good (one dimensional) spectral expander, then one can efficiently find solutions satisfying $\mathsf{OPT} - \varepsilon$ fraction of constraints, where OPT denotes the maximum fraction of constraints satisfiable by any assignment. Their algorithm is based on $(q/\varepsilon)^{O(1)}$ levels of the Sum-of-Squares (SoS) SDP hierarchy, and the expansion requirement on the constraint graph is that the (normalized) second singular value should be at most $(\varepsilon/q)^{O(1)}$. We show a similar result for k-CSPs when the corresponding simplicial complex $X_{\mathcal{J}}$, which is obtained by including one hyperedge for each constraint and taking a downward closure, is a sufficiently good $(k-1)$-dimensional expander.

Theorem 2.1.1 (Informal)**.** *Let $\mathcal{J}$ be an instance of MAX k-CSP on n variables taking values over an alphabet of size q, and let $\varepsilon > 0$. Let the simplicial complex $X_{\mathcal{J}}$ be a γ-HDX with $\gamma = \varepsilon^{O(1)} \cdot (1/(kq))^{O(k)}$. Then, there is an algorithm based on $(k/\varepsilon)^{O(1)} \cdot q^{O(k)}$ levels of the Sum-of-Squares hierarchy, which produces an assignment satisfying $\mathsf{OPT} - \varepsilon$ fraction of the constraints.*

Remark 2.1.2. *While the level-t relaxation for MAX k-CSP can be solved in time $(nq)^{O(t)}$*

[RW17], the rounding algorithms used by [BRS11] and our work do not need the full power of this relaxation. Instead, they are captured by the "local rounding" framework of Guruswami and Sinop [GS12] who show how to implement a local rounding algorithm based on t levels of the SoS hierarchy, in time $q^{O(t)} \cdot n^{O_k(1)}$ (where q denotes the alphabet size).

Our techniques. We start by using essentially the same argument for analyzing the SoS hierarchy as was used by [BRS11] (specialized to the case of expanders). They viewed the SoS solution as giving a joint distribution on each pair of variables forming a constraint, and proved that for sufficiently expanding graphs, these distributions can be made close to product distributions, by conditioning on a small number of variables (which governs the number of levels required). Similarly, we consider the conditions under which joint distributions on k-tuples corresponding to constraints can be made close to product distributions. Since the [BRS11] argument shows how to split a joint distribution into two marginals, we can use it to recursively split a set of size k into two smaller ones (one can think of all splitting operations as forming a binary tree with k leaves).

However, our arguments differ in the kind of expansion required to perform the above splitting operations. In the case of the 2-CSP, one splits along the edges of the constraint graph, and thus we only need the expansion of the contraint graph (which is part of the assumption). However, in the case of k-CSPs, we may split a set of size $(\ell_1 + \ell_2)$ into disjoint sets of size ℓ_1 and ℓ_2. This requires understanding the expansion of the following family of (weighted) bipartite graphs arising from the complex $X_{\mathcal{J}}$: The vertices in the graph are sets of variables of size ℓ_1 and ℓ_2 that occur in some constraint, and the weight of an edge $\{\mathfrak{a}_1, \mathfrak{a}_2\}$ for $\mathfrak{a}_1 \cap \mathfrak{a}_2 = \emptyset$, is proportional to the probability that a random constraint contains $\mathfrak{a}_1 \sqcup \mathfrak{a}_2$. Note that this graph may be weighted even if the k-CSP instance $\mathcal{J}$ is unweighted.

We view the above graphs as random walks, which we call "swap walks" on the hyperedges (faces) in the complex X. While several random walks on high-dimensional

expanders have been shown to have rapid mixing [KM17, KO18b, DK17, LLP17], we need a stronger condition. To apply the argument from [BRS11], we not only need that the second singular value is bounded away from one, but require it to be an arbitrarily small constant (as a function of ε, k and q). We show that this is indeed ensured by the condition that $\mathfrak{a}_1 \cap \mathfrak{a}_2 = \emptyset$, and obtain a bound of $k^{O(k)} \cdot \gamma$ on the second singular value. This bound, which constitutes much of the technical work in the paper, is obtained by first expressing these walks in terms of more canonical walks, and then using the beautiful machinery of harmonic analysis on expanding posets by Dikstein et al. [DDFH18] to understand their spectra.

The swap walks analyzed above represent natural random walks on simplicial complexes, and their properties may be of independent interest for other applications. Just as the high-dimensional expanders are viewed as "derandomized" versions of the complete complex (containing all sets of size at most k), one can view the swap walks as derandomized versions of (bipartite) Kneser graphs, which have vertex sets $\binom{[n]}{\ell_1}$ and $\binom{[n]}{\ell_2}$, and edges $(\mathfrak{a}, \mathfrak{b})$ iff $\mathfrak{a} \cap \mathfrak{b} = \emptyset$. We provide a more detailed and technical overview in Section 2.3 after discussing the relevant preliminaries in Section 2.2.

High-dimensional threshold rank. The correlation breaking method in [BRS11] can be applied as long as the graph has low threshold rank i.e., the number of singular values above a threshold $\tau = (\varepsilon/q)^{O(1)}$ is bounded. Similarly, the analysis described above can be applied, as long as all the swap walks which arise when splitting the k-tuples have bounded threshold rank. This suggests a notion of high-dimensional threshold rank for hypergraphs (discussed in Section 2.7), which can be defined in terms of the threshold ranks of the relevant swap walks. We remark that it is easy to show that dense hypergraphs (with $\Omega(n^k)$ hyperedges) have small-threshold rank according to this notion, and thus it can be used to recover known algorithms for approximating k-CSPs on dense instances [FK96] (as was true for threshold rank in graphs).

Other related work. While we extend the approach taken by [BRS11] for 2-CSPs, somewhat different approaches were considered by Guruswami and Sinop [GS11], and Oveis-Gharan and Trevisan [OGT15]. The work by Guruswami and Sinop relied on the expansion of the label extended graph, and used an analysis based on low-dimensional approximations of the SDP solution. Oveis-Gharan and Trevisan used low-threshold rank assumptions to obtain a regularity lemma, which was then used to approximate the CSP. For the case of k-CSPs, the Sherali-Adams hierarchy can be used to solve instances with bounded treewidth [WJ04] and approximately dense instances [YZ14, MR17]. Brandao and Harrow [BH13] also extended the results by [BRS11] for 2-CSPs to the case of 2-local Hamiltonians. We show that their ideas can also be used to prove a similar extension of our results to k-local Hamiltonians on high-dimensional expanders.

In case of high-dimensional expanders, in addition to canonical walks described here, a "non-lazy" version of these walks (moving from $\mathfrak{s}$ to $\mathfrak{t}$ only if $\mathfrak{s} \neq \mathfrak{t}$) was also considered by Kaufman and Oppenheim [KO18b], Anari et al. [ALGV18] and Dikstein et al. [DDFH18]. The swap walks studied in this paper were also considered independently in a very recent work of Dikstein and Dinur [DD19] (under the name "complement walks").

In a recent follow-up work [AJQ+20], the algorithms developed here were also used to obtain new unique and list decoding algorithms for direct sum and direct product codes, obtained by a "lifting" a base code C_0 via k-local operations to amplify distance. This work also showed that the hypergraphs obtained by considering collections of length-k walks on an expanding graph also satisfy (a slight variant of) splittability, and admit similar algorithms.

2.2 Preliminaries and Notation

2.2.1 Linear Algebra

Recall that for an operator $\mathsf{A} : V \to W$ between two finite-dimensional inner product spaces V and W, the operator norm can be written as

$$\|\mathsf{A}\|_{\mathrm{op}} = \sup_{f,g \neq 0} \frac{\langle \mathsf{A}f, g \rangle}{\|f\| \, \|g\|}.$$

Also, for such an A the adjoint $\mathsf{A}^\dagger : W \to V$ is defined as the (unique) operator satisfying $\langle \mathsf{A}f, g \rangle = \left\langle f, \mathsf{A}^\dagger g \right\rangle$ for all $f \in V, g \in W$. For $\mathsf{A} : V \to W$, we take $\|\mathsf{A}\|_{\mathrm{op}} = \sigma_1(\mathsf{A}) \geq \sigma_2(\mathsf{A}) \geq \cdots \geq \sigma_r(\mathsf{A}) > 0$ to be its singular values in descending order. Note that for $\mathsf{A} : V \to V$, $\sigma_2(\mathsf{A})$ denotes its second largest eigenvalue in absolute value.

2.2.2 High-Dimensional Expanders

A high-dimensional expander (HDX) is a particular kind of downward-closed hypergraph (simplicial complex) satisfying an expansion requirement. We elaborate on these properties and define well known natural walks on HDXs below.

Simplicial Complexes

Definition 2.2.1. *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 also referred to as* faces *of* X*.*

We use the notation $X(i)$ *to denote the collection of all faces* $\mathfrak{s}$ *in* X *with* $|\mathfrak{s}| = i$*. When faces are of cardinality at most* d*, we also use the notation* $X(\leq d)$ *to denote all the faces of* X*. By convention, we take* $X(0) := \{\varnothing\}$*.*

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

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, since we will only be treating these as hypergraphs, we prefer to index faces by their cardinality, to improve readability of related expressions.

An important simplicial complex is the complete complex.

Definition 2.2.2 (Complete Complex $\Delta_d(n)$)**.** *We denote by $\Delta_d(n)$ the complete complex with faces of size at most d i.e., $\Delta_d(n) := \{\mathfrak{s} \subseteq [n] \mid |\mathfrak{s}| \leq d\}$.*

Walks and Measures on Simplicial Complexes

Let C^k denote the space of real valued functions on $X(k)$ i.e.,

$$C^k := \{f \mid f\colon X(k) \to \mathbb{R}\} \cong \mathbb{R}^{X(k)}.$$

We describe natural walks on simplicial complexes considered in [DK17, DDFH18, KO18b], as stochastic operators, which map functions in C^i to C^{i+1} and vice-versa.

To define the stochastic operators associated with the walks, we first need to describe a set of probability measures which serve as the stationary measures for these random walks. For a pure simplicial complex $X(\leq d)$, we define a collection of probability measures $(\Pi_1, \ldots \Pi_d)$, with Π_i giving a distribution on faces in the slice $X(i)$.

Definition 2.2.3 (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* $\mathfrak{s}^{(d-1)}, \ldots, \mathfrak{s}^{(1)}$ *as defined above.*

The following is immediate from the definition above.

Proposition 2.2.4. *Let* $\mathfrak{a} \in X(\ell)$ *be an arbitrary face. For all* $j \geq 0$*, one has*

$$\sum_{\substack{\mathfrak{b} \in X(\ell+j): \\ \mathfrak{b} \supseteq \mathfrak{a}}} \Pi_{\ell+j}(\mathfrak{b}) \;=\; \binom{\ell+j}{j} \cdot \Pi_\ell(\mathfrak{a}).$$

For all k, we define the inner product of functions $f, g \in C^k$, according to associated measure Π_k

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{\mathfrak{s} \sim \Pi_k} [f(\mathfrak{s}) g(\mathfrak{s})] = \sum_{\mathfrak{s} \in X(k)} f(\mathfrak{s}) g(\mathfrak{s}) \cdot \Pi_k(\mathfrak{s}) \,.$$

We now define the up and down operators $\mathsf{U}_i : C^i \to C^{i+1}$ and $\mathsf{D}_{i+1} : C^{i+1} \to C^i$ as

$$\begin{aligned}
[\mathsf{U}_i g](\mathfrak{s}) \;&=\; \mathop{\mathbb{E}}_{\mathfrak{s}' \in X(i),\, \mathfrak{s}' \subseteq \mathfrak{s}} \left[g(\mathfrak{s}') \right] \;=\; \frac{1}{i+1} \cdot \sum_{x \in \mathfrak{s}} g(\mathfrak{s} \backslash \{x\}) \\
[\mathsf{D}_{i+1} g](\mathfrak{s}) \;&=\; \mathop{\mathbb{E}}_{\mathfrak{s}' \sim \Pi_{i+1} | \mathfrak{s}' \supset \mathfrak{s}} \left[g(\mathfrak{s}') \right] \;=\; \frac{1}{i+1} \cdot \sum_{x \notin \mathfrak{s}} g(\mathfrak{s} \sqcup \{x\}) \cdot \frac{\Pi_{i+1}(\mathfrak{s} \sqcup \{x\})}{\Pi_i(\mathfrak{s})}
\end{aligned}$$

An important consequence of the above definition is that U_i and D_{i+1} are adjoints with respect to the inner products of C^i and C^{i+1}.

Fact 2.2.5. $\mathsf{U}_i = \mathsf{D}_{i+1}^\dagger$*, i.e.,* $\langle \mathsf{U}_i f, g \rangle = \langle f, \mathsf{D}_{i+1} g \rangle$ *for every* $f \in C^i$ *and* $g \in C^{i+1}$*.*

Note that the operators can be thought of as defining random walks in a simplicial complex $X(\leq d)$. The operator U_i moves *down* from a face $\mathfrak{s} \in X(i+1)$ to a face $\mathfrak{s}' \in X(i)$, but lifts a function $g \in C^i$ *up* to a function $\mathsf{U} g \in C^{i+1}$. Similarly, the operator D_{i+1} can be thought of as defining a random walk which moves *up* from $\mathfrak{s} \in X(i)$ to $\mathfrak{s}' \in X(i+1)$.

It is easy to verify that these walks respectively map the measure Π_{i+1} to Π_i, and Π_i to Π_{i+1}.

High-Dimensional Expansion

We recall the notion of high-dimensional expansion (defined via local spectral expansion) considered by [DK17]. We first need a few pieces of notation.

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 2.2.6 (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 $\sigma_2(G(X_{\mathfrak{s}})) \leq \gamma$, where $\sigma_2(G(X_{\mathfrak{s}}))$ denotes the second singular value of the (normalized) adjacency matrix of $G(X_{\mathfrak{s}})$.*

2.2.3 Constraint Satisfaction Problems (CSPs)

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

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

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

A constraint $\mathcal{C}_\mathfrak{a}$ is said to be satisfied by an assignment σ if we have $\sigma|_\mathfrak{a} \in \mathcal{C}_\mathfrak{a}$ i.e., the restriction of σ on $\mathfrak{a}$ is contained in $\mathcal{C}_\mathfrak{a}$. We write, $\mathsf{SAT}_\mathfrak{I}(\sigma)$ for the (weighted fraction of the constraints) satisfied by the assignment σ i.e.,

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

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

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

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

2.2.4 *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 $\mathfrak{I} = (H, \mathcal{C}, w)$ 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.

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

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 i.e., for $T \subseteq S \subseteq [n]$, we have $\mu_{S|T} = \mu_T$, where $\mu_{S|T}$ denotes the restriction of the distribution μ_S to the set T. We use these distributions to define a collection of random variables $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ taking values in $[q]$, such that for any set S with $|S| \leq t$, the collection of variables $\{\mathbf{Y}_i\}_{i\in S}$ have a joint distribution μ_S. Note that the entire collection $(\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ *may not* have a joint distribution: this property is only true for sub-collections of size t. We will refer to the collection $(\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ 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 $\beta \in [q]^T$, we can define a $(t-|T|)$-local ensemble $(\mathbf{Y}'_1, \ldots, \mathbf{Y}'_n)$ by "conditioning" the local distributions on the event $\mathbf{Y}_T = \beta$, where Y_T is shorthand for the collection $\{\mathbf{Y}_i\}_{i\in T}$. For any S with $|S| \leq t-|T|$, we define the distribution of $\mathbf{Y}'_S$ as $\mu'_S := \mu_{S\cup T}|\{\mathbf{Y}_T = \beta\}$. Finally, the semidefinite program also ensures that for any such conditioning, the conditional covariance matrix

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

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 $\mu'_{S_1\cup S_2}$. The PSD-ness be easily verified

by noticing that the above matrix can be written as the Gram matrix of the vectors

$$w_{(S,\alpha)} := \frac{1}{\|v_{(T,\beta)}\|} \cdot v_{(T\cup S,\beta\circ\alpha)} - \frac{\|v_{(T\cup S,\beta\circ\alpha)}\|^2}{\|v_{(T,\beta)}\|^3} \cdot v_{(T,\beta)}$$

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

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

2.3 Proof Overview: Approximating MAX 4-XOR

We consider a simple example of a specific k-CSP, which captures most of the key ideas in our proof. Let $\mathfrak{I}$ be an unweighted instance of 4-XOR on n Boolean variables. Let H be a 4-uniform hypergraph on vertex set $[n]$, with a hyperedge corresponding to each constraint i.e., each $\mathfrak{a} = \{i_1, i_2, i_3, i_4\} \in H$ corresponds to a constraint in $\mathfrak{I}$ of the form

$$x_{i_1} + x_{i_2} + x_{i_3} + x_{i_4} = b_\mathfrak{a} \pmod 2,$$

for some $b_\mathfrak{a} \in \{0,1\}$. Let X denote the constraint complex for the instance $\mathfrak{I}$ such that $X(1) = [n]$, $X(4) = H$ and let $\Pi_1, \ldots, \Pi_4$ be the associated distributions (with Π_4 being uniform on H).

Local vs global correlation: the BRS strategy. We first recall the strategy used by [BRS11], which also suggests a natural first step for our proof. Given a 2-CSP instance with an associated graph G, and a t-local PSD ensemble $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ obtained from the SoS relaxation, they consider if the "local correlation" of the ensemble is small across the edges of G (which correspond to constraints) i.e.,

$$\mathop{\mathbb{E}}_{\{i,j\}\sim G} \left[\left\| \left\{ \mathbf{Y}_i \mathbf{Y}_j \right\} - \{\mathbf{Y}_i\} \left\{ \mathbf{Y}_j \right\} \right\|_1 \right] \ \leq \ \varepsilon \,.$$

If the local correlation is indeed small, we easily produce an assignment achieving a value $\mathsf{SDP} - \varepsilon$ in expectation, simply by rounding each variable x_i independently according to the distribution $\{\mathbf{Y}_i\}$. On the other hand, if this is not satisfied, they show (as a special case of their proof) that if G is an expander with second eigenvalue $\lambda \leq c \cdot (\varepsilon^2/q^2)$, then variables also have a high "global correlation", between a typical pair $(i,j) \in [n]^2$. Here, q is the alphabet size and c is a fixed constant. They use this to show that for $(\mathbf{Y}'_1, \ldots, \mathbf{Y}'_n)$ obtained by conditioning on the value of a randomly chosen $\mathbf{Y}_{i_0}$, we have

$$\mathop{\mathbb{E}}_{i} \left[\mathrm{Var} \left[\mathbf{Y}_i \right] \right] - \mathop{\mathbb{E}}_{i_0, \mathbf{Y}_{i_0}} \mathop{\mathbb{E}}_{i} \left[\mathrm{Var} \left[\mathbf{Y}'_i \right] \right] \ \geq \ \Omega(\varepsilon^2/q^2) \,,$$

where the expectations over i and i_0 are both according to the stationary distribution on the vertices of G. Since the variance is bounded between 0 and 1, this essentially shows that the local correlation must be at most ε after conditioning on a set of size $O(q^2/\varepsilon^2)$ (although the actual argument requires a bit more care and needs to condition on a somewhat larger set).

Extension to 4-XOR. As in [BRS11], we check if the t-local PSD ensemble $(\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ obtained from the SDP solution satisfies

$$\mathop{\mathbb{E}}_{\{i_1,i_2,i_3,i_4\}\in H} \left[\left\| \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} - \left\{ \mathbf{Y}_{i_1} \right\} \left\{ \mathbf{Y}_{i_2} \right\} \left\{ \mathbf{Y}_{i_3} \right\} \left\{ \mathbf{Y}_{i_4} \right\} \right\|_1 \right] \ \leq \ \varepsilon \, .$$

As before, independently sampling each x_i from $\{\mathbf{Y}_i\}$ gives an expected value at least $\mathsf{SDP} - \varepsilon$ in this case. If the above inequality is not satisfied, an application of triangle inequality gives

$$\mathop{\mathbb{E}}_{\{i_1,i_2,i_3,i_4\}\in H} \left[\begin{array}{l} \left\| \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} - \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \right\} \left\{ \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} \right\|_1 \ + \\ \left\| \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \right\} - \left\{ \mathbf{Y}_{i_1} \right\} \left\{ \mathbf{Y}_{i_2} \right\} \right\|_1 \ + \ \left\| \left\{ \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} - \left\{ \mathbf{Y}_{i_3} \right\} \left\{ \mathbf{Y}_{i_4} \right\} \right\|_1 \end{array} \right] \ > \ \varepsilon \, .$$

Symmetrizing over all orderings of $\{i_1, i_2, i_3, i_4\}$, we can write the above as

$$\varepsilon_2 + 2 \cdot \varepsilon_1 \ > \ \varepsilon \, ,$$

which gives $\max\{\varepsilon_1, \varepsilon_2\} \geq \varepsilon/3$. Here,

$$\begin{aligned} \varepsilon_1 \ &:= \ \mathop{\mathbb{E}}_{\{i_1,i_2\}\sim\Pi_2} \left[\left\| \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \right\} - \left\{ \mathbf{Y}_{i_1} \right\} \left\{ \mathbf{Y}_{i_2} \right\} \right\|_1 \right] , \quad \text{and} \\ \varepsilon_2 \ &:= \ \mathop{\mathbb{E}}_{\{i_1,i_2,i_3,i_4\}\sim\Pi_4} \left[\left\| \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} - \left\{ \mathbf{Y}_{i_1} \mathbf{Y}_{i_2} \right\} \left\{ \mathbf{Y}_{i_3} \mathbf{Y}_{i_4} \right\} \right\|_1 \right] \\ &= \ \mathop{\mathbb{E}}_{\{i_1,i_2,i_3,i_4\}\sim\Pi_4} \left[\left\| \left\{ \mathbf{Y}_{\{i_1,i_2\}} \mathbf{Y}_{\{i_3,i_4\}} \right\} - \left\{ \mathbf{Y}_{\{i_1,i_2\}} \right\} \left\{ \mathbf{Y}_{\{i_3,i_4\}} \right\} \right\|_1 \right] . \end{aligned}$$

As before, ε_1 measures the local correlation across edges of a weighted graph G_1 with vertex set $X(1) = [n]$ and edge-weights given by Π_2. Also, ε_2 measures the analogous quantity for a graph G_2 with vertex set $X(2)$ (pairs of variables occurring in constraints) and edge-weights given by Π_4.

Recall that the result from [BRS11] can be applied to *any* graph G over variables in a

2-local PSD ensemble, as long as the $\sigma_2(G)$ is small. Since $\{\mathbf{Y}_i\}_{i\in[n]}$ and $\{\mathbf{Y}_\mathfrak{s}\}_{\mathfrak{s}\in X(2)}$ are both $(t/2)$-local PSD ensembles (by Fact 4.7.14), we will apply the result to the graph G_1 on the first ensemble and G_2 on the second ensemble. We consider the potential

$$\Phi(\mathbf{Y}_1,\ldots,\mathbf{Y}_n) \;:=\; \mathop{\mathbb{E}}_{i\sim\Pi_1}\left[\mathrm{Var}\left[\mathbf{Y}_i\right]\right] \;+\; \mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_2}\left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{s}\right]\right].$$

Since local correlation is large along at least one of the graphs G_1 and G_2, using the above arguments (and the non-decreasing nature of variance under conditioning) it is easy to show that in expectation over the choice of $\{i_0,j_0\}\sim\Pi_2$ and $\beta\in[q]^2$ chosen from $\left\{\mathbf{Y}_{\{i_0,j_0\}}\right\}$, the conditional ensemble $(\mathbf{Y}'_1,\ldots,\mathbf{Y}'_n)$ satisfies

$$\Phi(\mathbf{Y}_1,\ldots,\mathbf{Y}_n) - \mathop{\mathbb{E}}_{i_0,j_0,\beta}\left[\Phi(\mathbf{Y}'_1,\ldots,\mathbf{Y}'_n)\right] \;=\; \Omega(\varepsilon^2),$$

provided G_1 *and* G_2 *satisfy* $\sigma_2(G_1),\sigma_2(G_2) \;\le\; c\cdot\varepsilon^2$ for an appropriate constant c.

The bound on the eigenvalue of G_1 follows simply from the fact that it is the skeleton of X, which is a γ-HDX. Obtaining bounds on the eigenvalues of G_2 and similar higher-order graphs, constitutes much of the technical part of this paper. Note that for a random sparse instance of MAX 4-XOR, the graph G_2 will be a matching with high probability (since $\{i_1,i_2\}$ in a constraint will only be connected to $\{i_3,i_4\}$ in the same constraint). However, we show that in case of a γ-HDX, this graph has second eigenvalue $O(\gamma)$. We analyze these graphs in terms of modified high-dimensional random walks, which we call "swap walks".

We remark that our potential and choice of a "seed set" of variables to condition on, is slightly different from [BRS11]. To decrease the potential function above, we need that for each level $X(i)$ ($i=1,2$ in the example above) the seed set must contain sufficiently many independent samples from $X(i)$ sampled according to Π_i. This can be ensured by drawing independent samples from the top level $X(k)$ (though $X(2)$ suffices in the above

example). In contrast, the seed set in [BRS11] consists of random samples from Π_1.

Analyzing Swap Walks. The graph G_2 defined above can be thought of as a random walk on $X(2)$, which starts at a face $\mathfrak{s} \in X(2)$, moves up to a face (constraint) $\mathfrak{s}' \in X(4)$ containing it, and then descends to a face $\mathfrak{t} \in X(2)$ such that $\mathfrak{t} \subset \mathfrak{s}'$ and $\mathfrak{s} \cap \mathfrak{t} = \emptyset$ i.e., the walk "swaps out" the elements in $\mathfrak{s}$ for other elements in $\mathfrak{s}'$. Several walks considered on simplicial complexes allow for the possibility of a non-trivial intersection, and hence have second eigenvalue lower bounded by a constant. On the other hand, swap walks completely avoid any laziness and thus turn out to have eigenvalues which can be made arbitrarily small. To understand the eigenvalues for this walk, we will express it in terms of other canonical walks defined on simplicial complexes.

Recall that the up and down operators can be used to define random walks on simplicial complexes. The up operator $\mathsf{U}_i : C^i \to C^{i+1}$ defines a walk that moves *down* from a face $\mathfrak{s} \in X(i+1)$ to a random face $\mathfrak{t} \in X(i), \mathfrak{t} \subset \mathfrak{s}$ (the operator thus "lifts" a function in C^i to a function in C^{i+1}). Similarly, the down operator $\mathsf{D}_i : C^i \to C^{i-1}$ moves *up* from a face $\mathfrak{s} \in X(i-1)$ to $\mathfrak{t} \in X(i), \mathfrak{t} \supset s$, with probability $\Pi_i(\mathfrak{t})/(i \cdot \Pi_{i-1}(\mathfrak{s}))$. These can be used to define a canonical random walk

$$\mathsf{N}_{2,2}^{(u)} := \mathsf{D}_3 \cdots \mathsf{D}_{u+2}\mathsf{U}_{u+1} \cdots \mathsf{U}_2\,, \qquad \mathsf{N}_{2,2}^{(u)} : C^2 \to C^2\,,$$

which moves from up for u steps $\mathfrak{s} \in X(2)$ to $\mathfrak{s}' \in X(u+2)$, and then descends back to $\mathfrak{t} \in X(2)$. Such walks were analyzed optimally by Dinur and Kaufman [DK17], who proved that $\lambda_2\left(\mathsf{N}_{2,2}^{(u)}\right) = 2/(u+2) \pm O_u(\gamma)$ when X is a γ-HDX. Thus, while this walk gives an expanding graph with vertex set $X(2)$, the second eigenvalue cannot be made arbitrarily small for a fixed u (recall that we are interested in showing that $\sigma_2(G_2) \leq c \cdot \varepsilon^2$). However, note that we are only interested in $\mathsf{N}_{2,2}^{(2)}$ *conditioned on the event* that the two elements from $\mathfrak{s}$ are "swapped out" with new elements in the final set $\mathfrak{t}$ i.e., $\mathfrak{s} \cap \mathfrak{t} = \emptyset$. We

define

$$\mathsf{S}_{2,2}^{(u,j)}(\mathfrak{s},\mathfrak{t}) \;:=\; \begin{cases} \frac{\binom{u+2}{2}}{\binom{u}{j}\cdot\binom{2}{2-j}} \cdot \mathsf{N}_{2,2}^{(u)} & \text{if } |\mathfrak{t}\setminus\mathfrak{s}| = j \\ 0 & \text{otherwise} \end{cases},$$

where the normalization is to ensure stochasticity of the matrix. In this notation, the graph G_2 corresponds to the random-walk matrix $\mathsf{S}_{2,2}^{(2,2)}$. We show that while $\sigma_2(\mathsf{N}_{2,2}^{(2)}) \approx 1/2$, we have that $\sigma_2(\mathsf{S}_{2,2}^{(2,2)}) = O(\gamma)$. We first write the canonical walks in terms of the swap walks. Note that

$$\mathsf{N}_{2,2}^{(2)} \;=\; \frac{1}{6}\cdot\mathsf{I} \;+\; \frac{2}{3}\cdot\mathsf{S}_{2,2}^{(2,1)} \;+\; \frac{1}{6}\cdot\mathsf{S}_{2,2}^{(2,2)},$$

since the "descent" step from $\mathfrak{s}' \in X(4)$ containing $\mathfrak{s} \in X(2)$, produces a $\mathfrak{t} \in X(2)$ which "swaps out" $0, 1$ and 2 elements with probabilities $1/6, 2/3$ and $1/6$ respectively. Similarly,

$$\mathsf{N}_{2,2}^{(1)} \;=\; \frac{1}{3}\cdot\mathsf{I} \;+\; \frac{2}{3}\cdot\mathsf{S}_{2,2}^{(1,1)}.$$

Finally, we use the fact (proved in Section 2.4) that while the canonical walks do depend on the "height" u (i.e., $\mathsf{N}_{2,2}^{(u)} \neq \mathsf{N}_{2,2}^{(u')}$) the swap walks (for a fixed number of swaps j) are independent of the height to which they ascend! In particular, we have

$$\mathsf{S}_{2,2}^{(2,1)} \;=\; \mathsf{S}_{2,2}^{(1,1)}.$$

Using these, we can derive an expression for the swap walk $\mathsf{S}_{2,2}^{(2,2)}$ as

$$\mathsf{S}_{2,2}^{(2,2)} \;=\; \mathsf{I} \;+\; 6\cdot\mathsf{N}_{2,2}^{(2)} - 6\cdot\mathsf{N}_{2,2}^{(1)} \;=\; \mathsf{I} \;+\; 6\cdot(\mathsf{D}_3\mathsf{D}_4\mathsf{U}_3\mathsf{U}_2 - \mathsf{D}_3\mathsf{U}_2)$$

To understand the spectrum of operators such as the ones given by the above expression, we use the beautiful machinery for harmonic analysis over HDXs (and more generally over expanding posets) developed by Dikstein et al. [DDFH18]. They show how to de-

compose the spaces C^k into approximate eigenfunctions for operators of the form DU. Using these decompositions and the properties of expanding posets, we can show that distinct eigenvalues of the above operator are approximately the same (up to $O(\gamma)$ errors) when analyzing the walks on the complete complex. Finally, we use the fact that swap walks in a complete complex correspond to Kneser graphs (for which the eigenvectors and eigenvalues are well-known) to show that $\lambda_2(\mathsf{S}^{(2,2)}_{2,2}) = O(\gamma)$.

Splittable CSPs and high-dimensional threshold rank. We note that the ideas used above can be generalized (at least) in two ways. In the analysis of distance from product distribution for a 4-tuple of random variables forming a contraint, we split it in 2-tuples. In general, we can choose to split tuples in a k-CSP instance along *any* binary tree $\mathcal{T}$ with k leaves, with each parent node corresponding to a swap walk between tuples forming its children. Finally, the analysis from [BRS11] also works if the each of the swap walks in some $\mathcal{T}$ have a bounded number (say r) of eigenvalues above some threshold τ, which provide a notion of high-dimensional threshold rank for hypergraphs. We refer to such an instance as a $(\mathcal{T}, \tau, r)$-splittable.

The arguments sketched above show that high-dimensional expanders are $(\mathcal{T}, O(\gamma), 1)$-splittable for all $\mathcal{T}$. Since the knowledge of $\mathcal{T}$ is only required in our analysis and not in the algorithm, we say that $\mathrm{rank}_\tau(\mathcal{I}) \leq r$ (or that $\mathcal{I}$ is (τ, r)-splittable) if $\mathcal{I}$ is $(\mathcal{T}, \tau, r)$-splittable for any $\mathcal{T}$. We defer the precise statement of results for (τ, r)-splittable instances to Section 2.7.

2.4 Walks

It is important to note that both U_i and D_{i+1} can be thought of as row-stochastic matrices, i.e. we can think of them as the probability matrices describing the movement of a walk

from $X(i+1)$ to $X(i)$; and from $X(i)$ to $X(i+1)$ respectively. More concretely, we will think

$$[\mathsf{D}_{i+1}^{\top} e_{\mathfrak{s}}](\mathfrak{t}) = \mathbb{P}\left[\text{the walk moves } \underline{\text{up}} \text{ from } \mathfrak{s} \in X(i) \text{ to } \mathfrak{t} \in X(i+1)\right]$$

and similarly

$$[\mathsf{U}_{i}^{\top} e_{\mathfrak{t}}](\mathfrak{s}) = \mathbb{P}\left[\text{the walk moves } \underline{\text{down}} \text{ from } \mathfrak{t} \in X(i+1) \text{ to } \mathfrak{s} \in X(i)\right].$$

By referring to the definition of the up and down operators in Section 2.2, it is easy to verify that

$$[\mathsf{D}_{i+1}^{\top} e_{\mathfrak{s}}](\mathfrak{t}) = \mathbf{1}[\mathfrak{t} \supseteq \mathfrak{s}] \cdot \frac{1}{i+1} \frac{\Pi_{i+1}(\mathfrak{t})}{\Pi_i(\mathfrak{s})} \quad \text{and} \quad [\mathsf{U}_{i}^{\top} e_{\mathfrak{t}}](\mathfrak{s}) = \mathbf{1}[\mathfrak{s} \subseteq \mathfrak{t}] \cdot \frac{1}{i+1}.$$

It is easy to see that our notion of random walk respects the probability distributions Π_j, i.e. we have

$$\mathsf{U}_{i}^{\top} \Pi_{i+1} = \Pi_i \quad \text{and} \quad \mathsf{D}_{i+1}^{\top} \Pi_i = \Pi_{i+1},$$

i.e., randomly moving up from a sample of Π_j gives a sample of Π_{j+1} and similarly, moving down from a sample of Π_{j+1} results in a sample of Π_j.

Instead of going up and down by one dimension, one can try going up or down by multiple dimensions since $\left(\mathsf{D}_{i+1} \cdots \mathsf{D}_{i+\ell}\right)$ and $\left(\mathsf{U}_{i+\ell} \cdots \mathsf{U}_{i}\right)$ are still row-stochastic matrices. Further, the corresponding probability vectors still have intuitive explanations in terms of the distributions Π_j. For a face $\mathfrak{s} \in X(k)$, we introduce the notation

$$p_{\mathfrak{s}}^{(u)} = \left(\mathsf{D}_{k+1} \cdots \mathsf{D}_{k+u}\right)^{\top} e_{\mathfrak{s}}$$

where we take $p_{\mathfrak{s}}^{(0)} = e_{\mathfrak{s}}$. This notation will be used to denote the probability distribution of the $\underline{\text{up-walk}}$ starting from $\mathfrak{s} \in X(k)$ and ending in a random face $\mathfrak{t} \in X(k+u)$ satisfying

$\mathfrak{t} \supseteq \mathfrak{s}$.

Note that the following Lemma together with Proposition 2.2.4 implies that $p_{\mathfrak{s}}^{(u)}$ is indeed a probability distribution.

Proposition 2.4.1. *For $\mathfrak{s} \in X(k)$ and $\mathfrak{a} \in X(k+u)$ one has,*

$$p_{\mathfrak{s}}^{(u)}(\mathfrak{a}) = \mathbf{1}[\mathfrak{a} \supseteq \mathfrak{s}] \cdot \frac{1}{\binom{k+u}{u}} \cdot \frac{\Pi_{k+u}(\mathfrak{a})}{\Pi_k(\mathfrak{s})}.$$

Proof. Notice that for $u = 0$, the statement holds trivially. We assume that there exists some $u \geq 0$ that satisfies

$$p_{\mathfrak{s}}^{(u)}(\mathfrak{a}) = \mathbf{1}[\mathfrak{a} \supseteq \mathfrak{s}] \cdot \frac{1}{\binom{k+u}{u}} \cdot \frac{\Pi_{k+u}(\mathfrak{a})}{\Pi_k(\mathfrak{s})}$$

for all $\mathfrak{a} \in X(k+u)$.

For $\mathfrak{b} \in X(k+(u+1))$ one has,

$$p_{\mathfrak{s}}^{(u+1)}(\mathfrak{b}) = [\mathsf{D}_{k+u+1}^{\top} p_{\mathfrak{s}}^{(u)}](\mathfrak{b}) = \frac{1}{k+u+2} \cdot \sum_{x \in \mathfrak{b}} \frac{\Pi_{k+u+1}(\mathfrak{b})}{\Pi_{k+u}(\mathfrak{b}\backslash\{x\})} \cdot p_{\mathfrak{s}}^{(u)}(\mathfrak{b}\backslash\{x\}).$$

Plugging in the induction assumption, this implies

$$\begin{aligned} p_{\mathfrak{s}}^{(u+1)}(\mathfrak{b}) &= \frac{1}{(k+u+1)} \cdot \sum_{x \in \mathfrak{b}} \frac{\Pi_{k+u+1}(\mathfrak{b})}{\Pi_{k+u}(\mathfrak{b}\backslash\{x\})} \cdot \left(\mathbf{1}[(\mathfrak{b}\backslash\{x)\}) \supseteq \mathfrak{s}] \cdot \frac{1}{\binom{k+u}{u}} \cdot \frac{\Pi_{k+u}(\mathfrak{b}\backslash\{x\})}{\Pi_k(\mathfrak{s})} \right), \\ &= \frac{1}{(k+u+1)} \cdot \frac{1}{\binom{k+u}{u}} \cdot \sum_{x \in \mathfrak{b}} \mathbf{1}[\mathfrak{b}\backslash\{x\} \supseteq \mathfrak{s}] \cdot \frac{\Pi_{k+u+1}(\mathfrak{b})}{\Pi_k(\mathfrak{s})}. \end{aligned}$$

First, note that the up-walk only hits the faces that contain $\mathfrak{s}$, otherwise $\mathbf{1}[\mathfrak{b}\backslash\{x\} \supseteq \mathfrak{s}] = 0$.

Suppose therefore $\mathfrak{b} \in X(k+u+1)$ satisfies $\mathfrak{b} \supseteq \mathfrak{s}$. Since there are precisely $(u+1)$

indices whose deletion still preserves the containment of $\mathfrak{s}$, we can write

$$\begin{aligned} p_{\mathfrak{s}}^{(u+1)}(\mathfrak{b}) &= \mathbf{1}[\mathfrak{b} \supseteq \mathfrak{s}] \cdot \frac{u+1}{k+u+1} \cdot \frac{1}{\binom{k+u}{u}} \frac{\Pi_{k+u+1}(\mathfrak{b})}{\Pi_k(\mathfrak{s})}, \\ &= \mathbf{1}[\mathfrak{b} \supseteq \mathfrak{s}] \cdot \frac{1}{\binom{k+u+1}{u+1}} \cdot \frac{\Pi_{k+u+1}(\mathfrak{b})}{\Pi_k(\mathfrak{s})}. \end{aligned}$$

Thus, proving the proposition. ∎

Similarly, we introduce the notation $q_{\mathfrak{a}}^{(u)}$, as

$$q_{\mathfrak{a}}^{(u)}(\mathfrak{s}) = \left(\mathsf{U}_{k+u-1} \cdots \mathsf{U}_k\right)^{\top} e_{\mathfrak{s}},$$

i.e. for the probability distribution of the down-walk starting from $\mathfrak{a} \in X(k+u)$ and ending in a random face of $X(k)$ contained in $\mathfrak{a}$. The following can be verified using Proposition 2.4.1, and the fact that $\left(\mathsf{U}_{k+u-1} \cdots \mathsf{U}_k\right)^{\dagger} = \mathsf{D}_{k+u} \cdots \mathsf{D}_{k+1}$.

Corollary 2.4.2. *Let $X(\leq d)$ be a simplicial complex, and $k, u \geq 0$ be parameters satisfying $k+u \leq d$. For $\mathfrak{a} \in X(k+u)$ and $\mathfrak{s} \in X(k)$, one has*

$$q_{\mathfrak{a}}^{(u)}(\mathfrak{s}) = \frac{1}{\binom{k+u}{u}} \cdot \mathbf{1}[\mathfrak{s} \subseteq \mathfrak{a}].$$

In the remainder of this section, we will try to construct more intricate walks on X from $X(k)$ to $X(l)$.

2.4.1 The Canonical and the Swap Walks on a Simplicial Complex

Definition 2.4.3 (Canonical and Swap u-Walks)**.** *Let $d \geq 0$, $X(\leq d)$ be a simplicial complex, and $k, l, u \geq 0$ be parameters satisfying $l \leq k$, $u \leq l$ and $d \geq k+u$; where the constraints on these parameters are to ensure well-definedness. We will define the following random walks,*

- ***canonical u-walk from $X(k)$ to $X(l)$.*** *Let $\mathsf{N}_{k,l}^{(u)}$ be the (row-stochastic) Markov operator that represents the following random walk: Starting from a face $\mathfrak{s} \in X(k)$,*

  – (random ascent/up-walk) *randomly move up a face $\mathfrak{s}'' \in X(k+u)$ that contains $\mathfrak{s}$, where $\mathfrak{s}''$ is picked with probability*

$$p_{\mathfrak{s}}^{(u)}(\mathfrak{s}'') = [\left(\mathsf{D}_{k+1} \cdots \mathsf{D}_{k+u}\right)^{\top} e_{\mathfrak{s}}](\mathfrak{s}'').$$

  – (random descent/down-walk) *go to a face $\mathfrak{s}' \in X(l)$ picked uniformly among all the l-dimensional faces that are contained in $\mathfrak{s}''$, i.e., the set $\mathfrak{s}'$ is picked with probability*

$$q_{\mathfrak{s}''}(\mathfrak{s}') = \mathbf{1}[\mathfrak{s}' \subseteq \mathfrak{s}''] \cdot \frac{1}{\binom{k+u}{l}} = [\left(\mathsf{U}_{k+u-1} \cdots \mathsf{U}_{l}\right)^{\top} e_{\mathfrak{s}''}](\mathfrak{s}').$$

  The operator $\mathsf{N}_{k,l}^{(u)} \colon C^l \to C^k$ satisfies the following equation,

$$\mathsf{N}_{k,l}^{(u)} = \mathsf{D}_{k+1} \cdots \mathsf{D}_{k+u} \mathsf{U}_{k+u-1} \cdot \mathsf{U}_k \cdots \mathsf{U}_l.$$

  Notice that we have $\mathsf{N}_{k,k}^{(0)} = \mathsf{I}$, and $\mathsf{N}_{k,l}^{(0)} = (\mathsf{U}_{k-1} \dots \mathsf{U}_l)$ for $l < k$.

- ***swapping walk from $X(k)$ to $X(l)$.*** *Let $\mathsf{S}_{k,l}$ be the Markov operator that represents the following random walk: Starting from a face $\mathfrak{s} \in X(k)$,*

  – (random ascent/up-walk) *randomly move up to a face $\mathfrak{s}'' \in X(k+l)$ that contains $\mathfrak{s}$, where as before $\mathfrak{s}''$ is picked with probability*

$$p_{\mathfrak{s}}^{(l)}(\mathfrak{s}'') = [\left(\mathsf{D}_{k+1} \cdots \mathsf{D}_{k+l+1}\right)^{\top} e_{\mathfrak{s}}](\mathfrak{s}'').$$

  – (deterministic descent) *deterministically go to $\mathfrak{s}' = \mathfrak{s}'' \backslash \mathfrak{s} \in X(l)$.*

For our applications, we will need to show that the walk $\mathsf{S}_{k,l}$ has good spectral expan-

sion whenever X is a d-dimensional γ-expander, for γ sufficiently small. To show this, we will relate the swapping walk operator $\mathsf{S}_{k,l}$ on X to the canonical random walk operators $\mathsf{N}_{k,l}^{(u)}$ (q.v. Lemma 2.4.4).

By the machinery of expanding posets (q.v. Section 2.5) it is possible to argue that the spectral expansion of the random walk operator $\mathsf{N}_{k,l}^{(u)}$ on a high dimensional expander will be close to that of the complete complex. This will allow us to conclude using the relation between the swapping walks and the canonical walks (q.v. Lemma 2.4.4) that the spectral expansion of the swapping walk on X, will be comparable with the spectral expansion of the swap walk on the complete complex. More precisely, we will show

Lemma 2.4.4 (Lemma 2.5.34)**.** *For any $d, k, l \geq 0$, and the complete simplicial simplicial complex $X(\leq d)$, one has the following: If $k \geq l \geq 0$ and $d \geq k + l$, we have*

$$\sigma_2(\mathsf{S}_{k,l}) = O_{k,l}\left(\frac{1}{n}\right).$$

Using these two, and the expanding poset machinery, we will conclude

Theorem 2.4.5 (Theorem 2.5.2 simplified)**.** *Let X be a d-dimensional γ expander. If $k \geq l \geq 0$ satisfy $d \geq l + k$ we have,*

$$\sigma_2(\mathsf{S}_{k,l}) = O_{k,l}(\gamma)$$

where $\mathsf{S}_{k,l}$ is the swapping walk on X from $X(k)$ to $X(l)$.

To prove Theorem 2.4.5 we will need to define an intermediate random walk that we will call the j-swapping u-walk from $X(k)$ to $X(l)$:

Definition 2.4.6 (j-swapping u-walk from $X(k)$ to $X(l)$)**.** *Given $d, u, j, k, l \geq 0$ satisfying $l \leq k$, $j \leq u$, $u \leq l$, and $d \geq k + u$. Let $\mathsf{S}_{k,l}^{(u,j)}$ be the Markov operator that represents the following random walk from $X(k)$ to $X(l)$ on a d-dimensional simplicial complex X: Starting from $\mathfrak{s} \in X(k)$*

- (random ascent/up-walk) *randomly move up to a face $\mathfrak{s}'' \in X(k+u)$ that contains $\mathfrak{s}$, where $\mathfrak{s}''$ is picked with probability*

$$p_{\mathfrak{s}}^{(u)}(\mathfrak{s}'') = [(\mathsf{D}_{k+1} \cdots \mathsf{D}_{k+u})^{\top} e_{\mathfrak{s}}](\mathfrak{s}'').$$

- (conditioned descent) *go to a face $\mathfrak{s}' \in X(l)$ sampled uniformly among all the subsets of $\mathfrak{s}'' \in X(k+u)$ that have intersection j with $\mathfrak{s}''\backslash\mathfrak{s}$, i.e. $|\mathfrak{s}' \cap (\mathfrak{s}''\backslash\mathfrak{s})| = j$.*

Notice that $\mathsf{S}_{k,l} = \mathsf{S}_{k,l}^{(l,l)}$ for any k and $\mathsf{I} = \mathsf{S}_{k,k}^{(u,0)}$ for any u.

Remark 2.4.7. *We will prove that the parameter u does not effect the swapping walk $\mathsf{S}_{k,l}^{(u,j)}$ so long as $u \geq j$, i.e. for all $u, u' \geq j$ we have $\mathsf{S}_{k,l}^{(u',j)} = \mathsf{S}_{k,l}^{(u,j)}$. Thus, we will often write $\mathsf{S}_{k,l}^{(j)}$ for $\mathsf{S}_{k,l}^{(j,j)}$.*

2.4.2 Swap Walks are Height Independent

Recall that the swap walk $\mathsf{S}_{k,l}^{(u,j)}$ is the conditional walk defined in terms of $\mathsf{N}_{k,l}^{(u)}$ where $\mathfrak{s} \in X(k)$ is connected to $\mathfrak{t} \in X(l)$ only if $|\mathfrak{t} \setminus \mathfrak{s}| = j$. The parameter u is called the height of the walk, namely the number of times it moves up. Since up and down operators have second singular value bounded away from 1, the second singular value of $\mathsf{N}_{k,l}^{(u)}$ shrinks as u increases. In other words, the operator $\mathsf{N}_{k,l}^{(u)}$ depends on the height u. Surprisingly, the walk $\mathsf{S}_{k,l}^{(u,j)}$ which is defined in terms of $\mathsf{N}_{k,l}^{(u)}$ does not depend on the particular choice of u as long as it is well defined. More precisely, we have the following result.

Lemma 2.4.8. *If X is a d-dimensional simplicial complex, $0 \leq l \leq k$, and $u, u' \in [j, d-k]$, then*

$$\mathsf{S}_{k,l}^{(u,j)} = \mathsf{S}_{k,l}^{(u',j)}.$$

In order to obtain Lemma 2.4.8, we will need a simple proposition:

Proposition 2.4.9. *Let $\mathfrak{s} \in X(k)$, $\mathfrak{s}' \subseteq \mathfrak{s}$ and $|t'| = j$. Suppose $\mathfrak{s}' \sqcup \mathfrak{t}' \in X(l)$. Then, we have*

$$\mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s}, \mathfrak{s}' \sqcup \mathfrak{t}') = \frac{1}{\binom{k}{l-j} \cdot \binom{u}{j}} \cdot \sum_{\substack{\mathfrak{a} \in X(k+u): \\ \mathfrak{a} \supseteq (\mathfrak{s} \sqcup \mathfrak{t}')}} p_{\mathfrak{s}}^{(u)}(\mathfrak{a}).$$

Proof. The only way of picking $\mathfrak{s}' \sqcup \mathfrak{t}'$ at the descent step is picking some $\mathfrak{a} \in X(k+u)$ that contains $\mathfrak{s}' \sqcup \mathfrak{t}'$ in the ascent step. The probability of this happening is precisely,

$$p_1 = \sum_{\substack{\mathfrak{a} \in X(k+u): \\ \mathfrak{a} \supseteq (\mathfrak{s} \sqcup \mathfrak{t}')}} p_{\mathfrak{s}}^{(u)}(\mathfrak{a}).$$

Suppose we are at a set $\mathfrak{a} = \mathfrak{s} \sqcup \mathfrak{t}$, such that $\mathfrak{t} \supseteq \mathfrak{t}'$ and $\mathfrak{s} \cap \mathfrak{t} = \varnothing$. Now, the probability of the descent step ending at $\mathfrak{s}' \sqcup \mathfrak{t}'$ is the probability of a randomly sampled $(l-j)$-elemented subset of $\mathfrak{s}$ being $\mathfrak{s}'$ and the probability of a randomly sampled j-elemented subset of $\mathfrak{t}$ being $\mathfrak{t}'$. It can be verified that this probability is

$$p_2 = \frac{1}{\binom{k}{l-j} \cdot \binom{u}{j}}.$$

By law of total probability we establish that

$$\mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s}, \mathfrak{s}' \sqcup \mathfrak{t}') = p_1 \cdot p_2 = \frac{1}{\binom{k}{l-j} \cdot \binom{u}{j}} \cdot \sum_{\substack{\mathfrak{a} \in X(k+u): \\ \mathfrak{a} \sqcup (\mathfrak{s} \sqcup \mathfrak{t}')}} p_{\mathfrak{s}}^{(u)}(\mathfrak{a}).$$

■

Lemma 2.4.10 (Height Independence)**.** *Let $u \in [j, d-k]$. For any $\mathfrak{s} \in X(k)$, $\mathfrak{s}' \subseteq \mathfrak{s}$ and $\mathfrak{t}' \in X(j)$ satisfying $\mathfrak{s}' \sqcup \mathfrak{t}' \in X(l)$ we have the following,*

$$\mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s}, \mathfrak{s}' \sqcup \mathfrak{t}') = \frac{1}{\binom{k}{l-j}\binom{k+j}{j}} \cdot \frac{\Pi_{k+j}(\mathfrak{s} \sqcup \mathfrak{t}')}{\Pi_k(\mathfrak{s})}.$$

In particular, the choice of $u \in [j, d-k]$ does not affect the swap walk.

Proof. We have,

$$\begin{aligned}\sum_{\mathfrak{a}\in X(k+u):\mathfrak{a}\supseteq\mathfrak{s}\sqcup\mathfrak{t}'} p_{\mathfrak{s}}^{(k+u)}(\mathfrak{a}) &= \frac{1}{\binom{k+u}{u}}\cdot\frac{1}{\Pi_k(\mathfrak{s})}\cdot\sum_{\mathfrak{a}\in X(k+u):\mathfrak{a}\supseteq\mathfrak{s}\sqcup\mathfrak{t}}\Pi_{k+u}(\mathfrak{a}),\\ &= \frac{\binom{k+u}{u-j}}{\binom{k+u}{u}}\cdot\frac{\Pi_{k+j}(\mathfrak{s}\sqcup\mathfrak{t}')}{\Pi_k(\mathfrak{s})}\end{aligned}$$

where the first equality is due to Proposition 2.4.1 and the second is due to Proposition 2.2.4 together with the observation that $\mathfrak{s}\sqcup\mathfrak{t}' \in X(k+j)$.

Thus, by Proposition 2.4.9 we get,

$$\mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s},\mathfrak{t}) = \frac{1}{\binom{u}{j}\cdot\binom{k}{l-j}}\frac{\binom{k+u}{u-j}}{\binom{k+u}{u}}\cdot\frac{\Pi_{k+j}(\mathfrak{s}\sqcup\mathfrak{t}')}{\Pi_k(\mathfrak{s})}.$$

We complete the proof by noting that,

$$\frac{\binom{k+u}{u-j}}{\binom{k+u}{u}} = \frac{\binom{u}{j}}{\binom{k+j}{j}},$$

and thus

$$\mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s},\mathfrak{t}) = \frac{1}{\binom{k}{l-j}\cdot\binom{k+j}{j}}\cdot\frac{\Pi_{k+j}(\mathfrak{s}\sqcup\mathfrak{t}')}{\Pi_k(\mathfrak{s})}$$

which proves the formula. ∎

Since the choice of u does not affect the formula, we obtain Lemma 2.4.8.

2.4.3 Canonical Walks in Terms of the Swap Walks

We show that the canonical walks are given by an average of swap walks with respect to the hypergeometric distribution.

Lemma 2.4.11. *Let $u, l, k, d \geq 0$ be given satisfying $l \leq k$ and $u \leq l$. Then, we have the following formula for the canonical u-walk on any $X(\leq d)$ satisfying $d \geq k + u$*

$$\mathsf{N}_{k,l}^{(u)} = \sum_{j=0}^{u} \frac{\binom{u}{j}\binom{k}{l-j}}{\binom{k+u}{l}} \cdot \mathsf{S}_{k,l}^{(j)}.$$

Proof. Suppose the canonical u-walk starting from $\mathfrak{s} \in X(k)$ picks $\mathfrak{s}'' \in X(k+u)$ in the second step. Write $\mathcal{E}_j(\mathfrak{s}'')$ for the event that the random face $\mathfrak{s}'$ the canonical u-walk picks in the descent step satisfies

$$\left|\mathfrak{s}' \setminus \mathfrak{s}\right| = j.$$

By elementary combinatorics,

$$\mathop{\mathbb{P}}_{\mathfrak{s}' \subseteq \mathfrak{s}''} \left[\mathcal{E}_j(\mathfrak{s}'') \mid \mathfrak{s}''\right] = \frac{\binom{u}{j}\binom{k}{l-j}}{\binom{k+u}{l}}$$

where the draw of the probability is over the subsets $\mathfrak{s}' \in X(l)$ of $\mathfrak{s}''$. Further, let $\mathfrak{t}_j'$ be the random variable that stands for the face picked in the descent step of the j-swapping u-walk from $X(k)$ to $X(l)$.

By the definition of the j-swapping walk from $X(k)$ to $X(l)$, conditioning that the ascent step picks the same $\mathfrak{s}'' \in X(k+u)$ we have

$$\mathbb{P}\left[\mathfrak{t}_j' = \mathfrak{t} \mid \mathfrak{s}''\right] = \mathbb{P}\left[\mathfrak{s}' = \mathfrak{t} \mid \mathfrak{s}'' \text{ and } \mathcal{E}_j(\mathfrak{s}'')\right]. \tag{2.1}$$

Now, by the law of total probability we have

$$\begin{aligned}
\mathsf{N}_{k,l}^{(u)}(\mathfrak{s},\mathfrak{t}) = \mathbb{P}\left[\mathbf{S}' = \mathfrak{t}\right] &= \sum_{j=0}^{u} \sum_{\mathfrak{s}'' \in X(k+u)} \mathbb{P}\left[\mathfrak{s}''\right] \cdot \mathbb{P}\left[\mathcal{E}_j(\mathfrak{s}'') \mid \mathfrak{s}''\right] \cdot \mathbb{P}\left[\mathfrak{s}' = \mathfrak{t} \mid \mathfrak{s}'' \text{ and } \mathcal{E}_j(\mathfrak{s}'')\right], \\
&= \sum_{j=0}^{u} \frac{\binom{u}{j}\binom{k}{l-j}}{\binom{k+u}{l}} \cdot \mathop{\mathbb{E}}_{\mathfrak{s}'' \supseteq \mathfrak{s}} \left[\mathbb{P}\left[\mathfrak{s}' = \mathfrak{t} \mid \mathfrak{s}'' \text{ and } \mathcal{E}_j(\mathfrak{s}'')\right]\right], \\
&= \sum_{j=0}^{u} \frac{\binom{u}{j}\binom{k+u}{l-j}}{\binom{k+u}{l}} \cdot \mathop{\mathbb{E}}_{\mathfrak{s}'' \supseteq \mathfrak{s}} \left[\mathbb{P}\left[\mathfrak{t}'_j = \mathfrak{t} \mid \mathfrak{s}''\right]\right]
\end{aligned}$$

where we used Equation (2.1) to get the last equality. Another application of the law of total probability gives us

$$\mathop{\mathbb{E}}_{\mathfrak{s}'' \supseteq \mathfrak{s}} \left[\mathbb{P}\left[\mathfrak{t}'_j = \mathfrak{t} \mid \mathfrak{s}''\right]\right] = \mathbb{P}\left[\mathfrak{t}'_j = \mathfrak{t}\right].$$

This allows us to write,

$$\begin{aligned}
\mathsf{N}_{k,l}^{(u)}(\mathfrak{s},\mathfrak{t}) &= \sum_{j=0}^{u} \frac{\binom{u}{j}\binom{k}{l-j}}{\binom{k+u}{l}} \cdot \mathbb{P}\left[\mathfrak{t}'_j = \mathfrak{t}\right], \\
&= \sum_{j=0}^{u} \frac{\binom{u}{j}\binom{k}{l-j}}{\binom{k+u}{l}} \cdot \mathsf{S}_{k,l}^{(u,j)}(\mathfrak{s},\mathfrak{t}),
\end{aligned}$$

The statement follows using height independence, i.e. Lemma 2.4.8 ∎

2.4.4 *Inversion: Swap Walks in Terms of Canonical Walks*

We show how the swap walks can be obtained as a signed sum of canonical walks. This result follows from binomial inversion which we recall next.

Fact 2.4.12 (Binomial Inversion, [BS02]). *Let $(a_n)_{n\geq 0}, (b_n)_{n\geq 0}$ be arbitrary sequences. Suppose for all $n \geq 0$ we have,*

$$b_n = \sum_{j=0}^{n} \binom{n}{j} \cdot (-1)^j \cdot a_j.$$

Then, we also have

$$a_n = \sum_{j=0}^{n} \binom{n}{j} \cdot (-1)^j \cdot b_j.$$

Corollary 2.4.13. *Let $k, l, d \geq 0$ be given parameters such that $k + l \leq d$ and $k \geq l$. For any simplicial complex $X(\leq d)$, one has the following formula for the u-swapping walk from $X(k)$ to $X(l)$ in terms of the canonical j-walks:*

$$\binom{k}{l-u} \mathsf{S}_{k,l}^{(u)} = \sum_{j=0}^{u} (-1)^{u-j} \cdot \binom{k+j}{l} \cdot \binom{u}{j} \cdot \mathsf{N}_{k,l}^{(j)}.$$

Proof. Fix faces $\mathfrak{s} \in X(k)$ and $\mathfrak{t} \in X(l)$ and set for all $j \in [0, u]$

$$a_j := \binom{k}{l-j} \cdot (-1)^j \cdot \mathsf{S}_{k,l}^{(j)}(\mathfrak{s}, \mathfrak{t}).$$

Notice that we have by Lemma 2.4.11

$$\binom{k+u}{l} \cdot \mathsf{N}_{k,l}^{(u)}(\mathfrak{s}, \mathfrak{t}) = \sum_{j=0}^{u} \binom{u}{j} \cdot (-1)^j \cdot a_j = \sum_{j=0}^{u} \binom{u}{j} \cdot \binom{k}{l-j} \cdot \cdot \mathsf{S}_{k,l}^{(j)}(\mathfrak{s}, \mathfrak{t}).$$

i.e. if we set

$$b_u = \binom{k+u}{l} \cdot \mathsf{N}_{k,l}^{(u)}(\mathfrak{s}, \mathfrak{t}),$$

we can apply Fact 2.4.12 to obtain

$$\begin{aligned} \binom{k}{l-u} \cdot (-1)^u \cdot \mathsf{S}_{k,l}^{(u)}(\mathfrak{s}, \mathfrak{t}) &= a_u \\ &= \sum_{j=0}^{u} \binom{u}{j} \cdot (-1)^j \cdot b_j \\ &= \sum_{j=0}^{u} \binom{u}{j} \cdot \binom{k+j}{l} \cdot (-1)^j \cdot \mathsf{N}_{k,l}^{(j)}(\mathfrak{s}, \mathfrak{t}). \end{aligned}$$

Dividing both sides of this equation by $(-1)^u$ yields the desired result. ∎

2.5 Spectral Analysis of Swap Walks

Swap walks arise naturally in our k-CSPs approximation scheme on HDXs where the running time and the quality of approximation depend on the expansion of these walks. For this reason, we analyze the spectra of swap walks. We show that swap walks $\mathsf{S}_{k,k}$ of γ-HDXs are indeed expanding for γ sufficiently small. More precisely, the first main result of this section is the following.

Theorem 2.5.1 (Swap Walk Spectral Bound)**.** *Let $X(\leq d)$ be a γ-HDX with $d \geq 2k$. Then the second largest singular value $\sigma_2(\mathsf{S}_{k,k})$ of the swap operator $\mathsf{S}_{k,k}$ is*

$$\sigma_2(\mathsf{S}_{k,k}) \;\leq\; \gamma \cdot \left(2^7 \cdot k^4 \cdot 2^{3k} \cdot k^k\right).$$

Theorem 2.5.1 is enough for the analysis of our k-CSP approximation scheme when k is a power of two. However, to analyze general k-CSPs on HDXs we need to understand the spectra of general swap walks $\mathsf{S}_{k,l}$ where k may differ from l. Therefore, we generalize the spectral analysis of $\mathsf{S}_{k,k}$ above to $\mathsf{S}_{k,l}$ obtaining Theorem 2.5.2, our second main result of this section.

Theorem 2.5.2 (Rectangular Swap Walk Spectral Bound)**.** *Suppose $X(\leq d)$ is a γ-HDX with $d \geq k+l$ and $k \leq l$. Then the largest non-trivial singular value $\sigma_2(\mathsf{S}_{k,l})$ of the swap operator $\mathsf{S}_{k,l}$ is*

$$\sigma_2(\mathsf{S}_{k,l}) \;\leq\; \sqrt{\gamma \cdot \left(2^8 \cdot k^2 \ell^2 \cdot 2^{2k+4l} \cdot k^k\right)}.$$

2.5.1 *Square Swap Walks $\mathsf{S}_{k,k}$*

We prove Theorem 2.5.1 by connecting the spectral structure of $\mathsf{S}_{k,k}$ of general γ-HDXs to the well behaved case of complete simplicial complexes. To distinguish these two cases

we denote by $\mathsf{S}^{\Delta}_{k,k}$ the swap $\mathsf{S}_{k,k}$ of complete complexes [2]. In fact, $\mathsf{S}^{\Delta}_{k,k}$ is the random walk operator of the well known Kneser graph $K(n,k)$ (see Definition 2.5.3).

Definition 2.5.3 (Kneser Graph $K(n,k)$ [GM15])**.** *The Kneser graph $K(n,k)$ is the graph $G = (V,E)$ where $V = \binom{[n]}{k}$ and $E = \{\{\mathfrak{s},\mathfrak{t}\} \mid \mathfrak{s} \cap \mathfrak{t} = \emptyset\}$.*

Then at least for complete complexes we know that $\mathsf{S}^{\Delta}_{k,k}$ is expanding. This is a direct consequence of Fact 2.5.4.

Fact 2.5.4 (Kneser Graph [GM15])**.** *The singular values [3] of the Kneser graph $K(n,k)$ are*

$$\binom{n-k-i}{k-i},$$

for $i = 0,\ldots,k$.

This means that $\sigma_2(\mathsf{S}^{\Delta}_{k,k}) = O_k(1/n)$ as shown in Claim 2.5.5.

Claim 2.5.5. *Let $d \geq 2k$ and $\Delta_d(n)$ be the complete complex. The second largest singular value $\sigma_2(\mathsf{S}^{\Delta}_{k,k})$ of the swap operator $\mathsf{S}^{\Delta}_{k,k}$ on $\Delta_d(n)$ is*

$$\sigma_2(\mathsf{S}^{\Delta}_{k,k}) \;=\; \frac{k}{n-k},$$

provided $n \geq M_k$ where $M_k \in \mathbb{N}$ only depends on k.

Proof. First note that for the complete complex $\Delta_d(n)$, the operator $\mathsf{S}^{\Delta}_{k,k}$ is the walk matrix of the Kneser graph $K(n,k)$. Since the degree of $K(n,k)$ is $\binom{n-k}{k}$, the result follows from Fact 2.5.4. ∎

2. The precise parameters of the complete complex $\Delta_d(n)$ where $\mathsf{S}^{\Delta}_{k,k}$ lives will not be important. We only require that $\mathsf{S}^{\Delta}_{k,k}$ is well defined in the sense that $d \geq 2k$ and $n > d$.

3. The precise eigenvalues are also well known, but singular values are enough in our analysis.

Therefore, if we could claim that $\sigma_2(\mathsf{S}_{k,k})$ of an arbitrary γ-HDX is close to $\sigma_2(\mathsf{S}^{\Delta}_{k,k})$ (provided γ is sufficiently small), we would conclude that general $\mathsf{S}_{k,k}$ walks are also expanding. A priori there is no reason why this claim should hold since a general d-sized γ-HDX may have much fewer hyperedges ($O_d(n)$ versus $\binom{n}{d}$ in the complete $\Delta_d(n)$). Fortunately, it turns out that this claim is indeed true (up to $O_k(\gamma)$ errors).

To prove Theorem 2.5.1 we employ the beautiful expanding poset (EPoset) machinery of Dikstein et al. [DDFH18]. Before we delve into the full technical analysis, it might be instructive to see how Theorem 2.5.1 is obtained from understanding the quadratic form $\langle \mathsf{S}_{k,k} f, f \rangle$ where $f \in C^k$.

First we informally recall the decomposition $C^k = \sum_{i=0}^{k} C^k_i$ from the EPoset machinery where C^k_i can be thought of as the space of approximate eigenfunctions of *degree* i of C^k (the precise definitions are deferred to 2.5.2). In this decomposition, C^k_0 is defined as the space of constant functions of C^k.

We prove the stronger result that the $\mathsf{S}_{k,k}$ operators of any γ-HDX has an an approximate spectrum that only depends on k provided γ is small enough. More precisely, we prove Lemma 2.5.6.

Lemma 2.5.6 (Swap Quadratic Form)**.** *Let $f = \sum_{i=0}^{k} f_i$ with $f_i \in C^k_i$. Suppose $X(\leq d)$ is a γ-HDX with $d \geq 2k$. If $\gamma \leq \varepsilon \left(64k^{k+4}2^{3k+1}\right)^{-1}$, then*

$$\langle \mathsf{S}_{k,k} f, f \rangle \;=\; \sum_{i=0}^{k} \lambda_k(i) \cdot \langle f_i, f_i \rangle \;\pm\; \varepsilon,$$

where $\lambda_k(i)$ depends only on k and i, i.e., $\lambda_k(i)$ is an approximate eigenvalue of $\mathsf{S}_{k,k}$ associated to space C^k_i.

Remark 2.5.7. *From Lemma 2.5.6, it might seem that we are done since there exist approximate eigenvalues $\lambda_k(i)$ that only depend on k and i. However, giving an explicit expression for these approximate eigenvalues is tricky. For this reason, we rely on the expansion of Kneser graphs as*

will be clear later.

Towards showing Lemma 2.5.6, we introduce the notion of *balanced* operators which in particular captures canonical and swap walks and we show that the quadratic form expression of Lemma 2.5.6 is a particular case of a general result for $\langle \mathsf{B}f, f\rangle$ where B is a general *balanced* operator. A *balanced* operator in C^k is any operator that can be obtained as linear combination of *pure balanced* operators, the later being operators that are a formal product of an equal number of up and down operators.

Lemma 2.5.8 (General Quadratic Form)**.** *Let $\varepsilon \in (0,1)$ and let $\mathcal{Y} \subseteq \{\mathsf{Y} \mid \mathsf{Y}\colon C^k \to C^k\}$ be a collection of formal operators that are product of an equal number of up and down walks (i.e., pure balanced operators) not exceeding ℓ walks. Let $\mathsf{B} = \sum_{\mathsf{Y}\in\mathcal{Y}} \alpha^{\mathsf{Y}}\mathsf{Y}$ where $\alpha^{\mathsf{Y}} \in \mathbb{R}$ and let $f = \sum_{i=0}^{k} f_i$ with $f_i \in C_i^k$. If $\gamma \leq \varepsilon \left(16k^{k+2}\ell^2 \sum_{\mathsf{Y}\in\mathcal{Y}} |\alpha^{\mathsf{Y}}|\right)^{-1}$, then*

$$\langle \mathsf{B}f, f\rangle \;=\; \sum_{i=0}^{k}\left(\sum_{\mathsf{Y}\in\mathcal{Y}} \alpha^{\mathsf{Y}}\lambda_k^{\mathsf{Y}}(i)\right)\cdot\langle f_i, f_i\rangle \;\pm\; \varepsilon,$$

where $\lambda_k^{\mathsf{Y}}(i)$ depends only on the operators appearing in the formal expression of Y, i and k, i.e., $\lambda_k^{\mathsf{Y}}(i)$ is the approximate eigenvalue of Y associated to C_i^k.

Remark 2.5.9. *Note that our result generalizes the analysis of [DDFH18] for expanding posets of HDXs which considered the particular case $\mathsf{B} = \mathsf{D}_{k+1}\mathsf{U}_k$. Moreover, their error term analysis treated all the parameters not depending on the number of vertices n as constants. In this work we make the dependence on the parameters explicit since this dependence is important in understanding the limits of our k-CSPs approximation scheme on HDXs. The beautiful EPoset machinery [DDFH18] is instrumental in our analysis.*

Now, we are ready to prove Theorem 2.5.1. For convenience we restate it below.

Theorem 2.5.10 (Swap Walk Spectral Bound (restatement of Theorem 2.5.1))**.** *Let $X(\leq d)$ be a γ-HDX with $d \geq 2k$. For every $\sigma \in (0,1)$, if $\gamma \leq \sigma \cdot \left(64k^{k+4}2^{3k+1}\right)^{-1}$, then the second*

largest singular value $\sigma_2(\mathsf{S}_{k,k})$ of the swap operator $\mathsf{S}_{k,k}$ is

$$\sigma_2(\mathsf{S}_{k,k}) \;\leq\; \sigma.$$

Proof. First we show that for $i \in [k]$ the i-th approximate eigenvalue $\lambda_k(i)$ of the swap operator $\mathsf{S}_{k,k}$ is actually zero. Note that for $i \in [k]$ the space C_i^k is a non-trivial eigenspace (i.e., C_i^k is not the space of constant functions). Let $\mathsf{S}^{\Delta}_{k,k}$ be the swap operator of the complete complex $\Delta_d(n)$. On one hand Claim 2.5.5 gives

$$\sigma_2(\mathsf{S}^{\Delta}_{k,k}) \;=\; \max_{f \in C^k\colon f \perp 1, \|f\|=1} \left|\left\langle \mathsf{S}^{\Delta}_{k,k} f, f \right\rangle\right| \;=\; O_k\left(\frac{1}{n}\right).$$

On the other hand since $\Delta_d(n)$ is a γ^{Δ}-HDX where $\gamma^{\Delta} = O_k(1/n)$, if n is sufficiently large we have $\gamma^{\Delta} \leq \gamma$ and thus Lemma 2.5.8 can be applied to give

$$\sigma_2(\mathsf{S}^{\Delta}_{k,k}) \;\geq\; \max_{f_i \in C_i^k\colon i \in [k], \|f_i\|=1} \left|\left\langle \mathsf{S}^{\Delta}_{k,k} f_i, f_i \right\rangle\right| \;=\; |\lambda_k(i)| \cdot \langle f_i, f_i \rangle \;\pm\; O_k\left(\frac{1}{n}\right).$$

Since n is arbitrary and $\lambda_k(i)$ depends only on k and i, we obtain $\lambda_k(i) = 0$ as claimed. Now applying Lemma 2.5.8 to the swap operator $\mathsf{S}_{k,k}$ of the γ-HDX $X(\leq d)$ yields

$$\sigma_2(\mathsf{S}_{k,k}) \;=\; \max_{f \in C^k\colon f \perp 1, \|f\|=1} \left|\left\langle \mathsf{S}_{k,k} f, f \right\rangle\right| \;\leq\; \max_{i \in [k]} \; |\lambda_k(i)| \;+\; \sigma \;=\; \sigma,$$

concluding the proof. ∎

2.5.2 *Expanding Posets and Balanced Operators*

We state the definitions used in our technical proofs starting with γ-EPoset from [DDFH18].

Definition 2.5.11 (γ-EPoset adapted from [DDFH18])**.** *A complex $X(\leq d)$ with operators*

$\mathsf{U}_0, \ldots, \mathsf{U}_{d-1}, \mathsf{D}_1, \ldots, \mathsf{D}_d$ *is said to be a* γ*-EPoset* [4] *provided*

$$\left\| \mathsf{M}_i^+ - \mathsf{U}_{i-1}\mathsf{D}_i \right\|_{\mathrm{op}} \leq \gamma, \tag{2.2}$$

for every $i = 1, \ldots, d-1$*, where*

$$\mathsf{M}_i^+ := \frac{i+1}{i}\left(\mathsf{D}_{i+1}\mathsf{U}_i - \frac{1}{i+1}\mathsf{I}\right),$$

i.e., M_i^+ *is the non-lazy version of the random walk* $\mathsf{N}_{i,i}^{(1)} = \mathsf{D}_{i+1}\mathsf{U}_i$*.*

Definition 2.5.11 can be directly used as an operational definition of high-dimension expansion as done in [DDFH18]. To us it is important that γ-HDXs are also γ-EPosets as established in Lemma 2.5.12. In fact, these two notions are known to be closely related.

Lemma 2.5.12 (From [DDFH18])**.** *Let* X *be a* d*-sized simplicial complex.*

- *If* X *is a* γ*-HDX, then* X *is a* γ*-EPoset.*
- *If* X *is a* γ*-EPoset, then* X *is a* $3d\gamma$*-HDX.*

Naturally the complete complex $\Delta_d(n)$ is a γ-EPoset since it is a γ-HDX. Moreover, in this particular case γ vanishes as n grows.

Lemma 2.5.13 (From [DDFH18])**.** *The complete complex* $\Delta_d(n)$ *is a* γ*-EPoset with* $\gamma = O_d\left(1/n\right)$*.*

4. We tailor their general EPoset definition to HDXs. In fact, what they call γ-HDX we call γ-EPoset. Moreover, what they call γ-HD expander we call γ-HDX.

The space C^k defined in Section 2.2.2 can be decomposed into subspaces C^k_i of functions of *degree* i for $0 \leq i \leq k$ where

$$C^k_i \;:=\; \{\mathsf{U}^{k-i} h_i \mid h_i \in H_i\},$$

with $H_i \;:=\; \ker(\mathsf{D}_i)$, and

$$C^k_0 \;:=\; \{f \colon X(k) \to \mathbb{R} \mid \; f \text{ is constant}\}.$$

More precisely, we have the following.

Lemma 2.5.14 (From [DDFH18])**.**

$$C^k \;=\; \sum_{i=0}^{k} C^k_i.$$

Lemma 2.5.14 is proven in Appendix A.2 as Lemma A.2.3.

For convenience set $\vec{\delta} \in \mathbb{R}^{d-1}$ such that $\delta_i = 1/(i+1)$ for $i \in [d-1]$. It will also be convenient to work with the following equivalent version of Eq. (2.2)

$$\|\mathsf{D}_{i+1}\mathsf{U}_i - (1-\delta_i)\mathsf{U}_{i-1}\mathsf{D}_i - \delta_i \mathsf{I}\|_{\mathrm{op}} \;\leq\; \frac{i}{i+1}\gamma. \tag{2.3}$$

Towards our goal of understanding quadratic forms of swap operators we study the approximate spectrum of operators of the form $\mathsf{Y} = \mathsf{Y}_\ell \ldots \mathsf{Y}_1$ where each Y_i is either an up or down operator, namely, Y is a generalized random walk of ℓ steps. We regard the expression $\mathsf{Y}_\ell \ldots \mathsf{Y}_1$ defining Y as a formal product.

Definition 2.5.15 (Pure Balanced Operator)**.** *We call* $\mathsf{Y} \colon C^k \to C^k$ *a pure balanced operator if* Y *can be defined as product* $\mathsf{Y}_\ell \ldots \mathsf{Y}_1$ [5] *where each* Y_i *is either an up or down operator. When we*

5. For the analysis it is convenient to order the indices appearing in $\mathsf{Y}_\ell \ldots \mathsf{Y}_1$ in decreasing order from

say that the spectrum of Y *depends on* Y *we mean that it depends on* k *and on the formal expression* $\mathsf{Y}_\ell \dots \mathsf{Y}_1$ *(i.e., pattern of up and down operators).*

Remark 2.5.16. *By definition canonical walks* $\mathsf{N}_{k,k}^{(u)}$ *are pure balanced operators.*

Taking linear combinations of *pure balanced* operators leads to the notion of *balanced* operators.

Definition 2.5.17 (Balanced Operator)**.** *We call* $\mathsf{B}\colon C^k \to C^k$ *a balanced operator provided there exists a set of pure balanced operators* $\mathcal{Y}$ *such that*

$$\mathsf{B} \;=\; \sum_{\mathsf{Y}\in\mathcal{Y}} \alpha^{\mathsf{Y}} \cdot \mathsf{Y},$$

where $\alpha^{\mathsf{Y}} \in \mathbb{R}$.

Remark 2.5.18. *Corollary 2.4.13 establishes that* $\mathsf{S}_{k,k}^{(u)}$ *are balanced operators. In particular,* $\mathsf{S}_{k,k}$ *is a balanced operator.*

It turns out that at a more crude level the behavior of Y is governed by how the number of up operators compares to the number of down operators. For this reason, it is convenient to define $\mathcal{U}(\mathsf{Y}) = \{\mathsf{Y}_i \mid \mathsf{Y}_i \text{ is an up operator}\}$ and $\mathcal{D}(\mathsf{Y}) = \{\mathsf{Y}_i \mid \mathsf{Y}_i \text{ is a down operator}\}$ where Y is a *pure balanced* operator. When Y is clear in the context we use $\mathcal{U} = \mathcal{U}(\mathsf{Y})$ and $\mathcal{D} = \mathcal{D}(\mathsf{Y})$.

Henceforth we assume $h_i \in H_i = \ker(\mathsf{D}_i)$, $f_i \in C_i^k$ and $g \in C^k$. This convention will make the statements of the technical results of Section 2.5.3 cleaner.

2.5.3 *Quadratic Forms over Balanced Operators*

Now we establish all the technical results leading to and including the analysis of quadratic forms over *balanced operators*. By considering this general class of operators our analysis

left to right.

generalizes the analysis given in [DDFH18]. At the same time we refine their error terms analysis by making the dependence on the EPoset parameters explicit. Recall that an explicit dependence on these parameters is important in understanding the limits of our k-CSP approximation scheme.

Lemma 2.5.19 (General Quadratic Form (restatement of Lemma 2.5.8))**.** *Let $\varepsilon \in (0,1)$ and let $\mathcal{Y} \subseteq \{\mathsf{Y} \mid \mathsf{Y} \colon C^k \to C^k\}$ be a collection of formal operators that are product of an equal number of up and down walks (i.e., pure balanced operators) not exceeding ℓ walks. Let $\mathsf{B} = \sum_{\mathsf{Y}\in\mathcal{Y}} \alpha^{\mathsf{Y}}\mathsf{Y}$ where $\alpha^{\mathsf{Y}} \in \mathbb{R}$ and let $f = \sum_{i=0}^k f_i$ with $f_i \in C_i^k$. If $\gamma \leq \varepsilon \left(16k^{k+2}\ell^2 \sum_{\mathsf{Y}\in\mathcal{Y}} |\alpha^{\mathsf{Y}}|\right)^{-1}$, then*

$$\langle \mathsf{B}f, f\rangle \;=\; \sum_{i=0}^{k} \left(\sum_{\mathsf{Y}\in\mathcal{Y}} \alpha^{\mathsf{Y}}\lambda_k^{\mathsf{Y}}(i)\right)\cdot\langle f_i, f_i\rangle \;\pm\; \varepsilon,$$

where $\lambda_k^{\mathsf{Y}}(i)$ depends only on the operators appearing in the formal expression of Y, i and k, i.e., $\lambda_k^{\mathsf{Y}}(i)$ is the approximate eigenvalue of Y associated to C_i^k.

Since swap walks are *balanced operators*, we will deduced the following (as proven later).

Lemma 2.5.20 (Swap Quadratic Form (restatement of Lemma 2.5.6))**.** *Let $f = \sum_{i=0}^k f_i$ with $f_i \in C_i^k$. Suppose $X(\leq d)$ is a γ-HDX with $d \geq 2k$. If $\gamma \leq \varepsilon \left(64k^{k+4}2^{3k+1}\right)^{-1}$, then*

$$\langle \mathsf{S}_{k,k}f, f\rangle \;=\; \sum_{i=0}^{k} \lambda_k(i)\cdot\langle f_i, f_i\rangle \;\pm\; \varepsilon,$$

where $\lambda_k(i)$ depends on only on k an i, i.e., $\lambda_k(i)$ is an approximate eigenvalue of $\mathsf{S}_{k,k}$ associated to space C_i^k.

The next result, Lemma 2.5.21, (implicit in [DDFH18]) will be key in establishing that the spectral structure of γ-EPosets is fully determined by the parameters in $\vec{\delta}$ provided γ is small enough. Note that the Eposet Definition 2.5.11 provides a "calculus" for rearranging

a single pair of up and down DU. The next result treats the more general case of $\mathsf{DU}\cdots\mathsf{U}$.

Lemma 2.5.21 (Structure Lemma)**.** *Suppose $|\mathcal{D}| = 1$. Let $\mathsf{Y}_c \in \mathcal{D}$ be the unique down operator in $\mathsf{Y}_\ell \dots \mathsf{Y}_1$. If $\|\mathsf{A}\|_{\mathrm{op}} \le 1$, then*

$$\langle \mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_1 h_i, g\rangle \;=\; \begin{cases} 0 & \text{if } \ell = 1 \text{ or } c = 1 \\ Q_{c,i}(\vec{\delta}) \cdot \left\langle \mathsf{A}\mathsf{U}^{\ell-2} h_i, g \right\rangle \;\pm\; (c-1)\cdot\gamma\,\|h_i\|\,\|g\| & \text{otherwise,} \end{cases}$$

where $Q_{c,i}$ is a polynomial in the variables $\vec{\delta}$ depending on c, i such that $Q_{c,i}(\vec{\delta}) \le 1$.

Proof. We induct on (ℓ, c). If $\ell = 1$ or $c = 1$, we have $\mathsf{Y}_1 h_i = \mathsf{D}_i h_i = 0$ so the result trivially holds. Otherwise, we have $\mathsf{Y}_c\mathsf{Y}_{c-1} = \mathsf{D}_{j+1}\mathsf{U}_j$ where $j = i + c - 2$. Then

$$\langle \mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_{c+1}(\mathsf{Y}_c\mathsf{Y}_{c-1})\mathsf{Y}_{c-2}\dots\mathsf{Y}_1 h_i, g\rangle\,,$$

becomes

$$\begin{aligned}
&(1-\delta_j)\cdot\langle \mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_{c+1}\mathsf{U}_{j-1}\mathsf{D}_j\mathsf{Y}_{c-2}\dots\mathsf{Y}_1 h_i, g\rangle \;+\; \delta_j\cdot\langle \mathsf{A}\mathsf{Y}_\ell\dots\mathsf{Y}_{c+1}\mathsf{Y}_{c-2}\dots\mathsf{Y}_1 h_i, g\rangle \;\pm\; \gamma\,\|h_i\|\,\|g\| && \text{(Eq. (2.2))}\\
&= (1-\delta_j)\cdot\langle \mathsf{A}\mathsf{Y}_1\dots\mathsf{Y}_{c-1}\mathsf{U}_{j-1}\mathsf{D}_j\mathsf{Y}_{c+2}\dots\mathsf{Y}_\ell h_i, g\rangle \;+\; \delta_j\cdot\left\langle \mathsf{A}\mathsf{U}^{\ell-2}h_i, g\right\rangle \;\pm\; \gamma\,\|h_i\|\,\|g\| &&\\
&= (1-\delta_j)\cdot Q_{c-1,i}(\vec{\delta})\cdot\left\langle \mathsf{A}\mathsf{U}^{\ell-2}h_i, g\right\rangle \;\pm\; (1-\delta_j)\cdot(c-2)\,\gamma\,\|h_i\|\,\|g\| \;+\; \delta_j\cdot\left\langle \mathsf{A}\mathsf{U}^{\ell-2}h_i, g\right\rangle \;\pm\; \gamma\,\|h_i\|\,\|g\| && \text{(I.H.)}\\
&= Q_{c,i}(\vec{\delta})\cdot\left\langle \mathsf{A}\mathsf{U}^{\ell-2}h_i, g\right\rangle \;\pm\; (c-1)\cdot\gamma\,\|h_i\|\,\|g\|\,. &&
\end{aligned}$$

■

With Lemma 2.5.21 we are close to recover the approximate spectrum of $\mathsf{D}_{k+1}\mathsf{U}_k$ from [DDFH18]. However, in our application we will need to analyze more general operators, namely, *pure balanced* and *balanced* operators.

Lemma 2.5.22 (Refinement of [DDFH18]). *If* $\|\mathsf{A}\|_{\mathrm{op}} \leq 1$, *then*

$$\left\langle \mathsf{A}\mathsf{D}_{k+1}\mathsf{U}_k f_i, g \right\rangle \;=\; \lambda_i \cdot \langle \mathsf{A} f_i, g \rangle \;\pm\; (k-i+1)\cdot\gamma \,\|h_i\|\,\|g\|\,,$$

where $\lambda_i = Q_{k-i+2,i}(\vec{\delta})$.

Proof. Recall that $f_i = \mathsf{U}^{k-i} h_i$ where $h_i \in \ker(\mathsf{D}_i)$. Set $\mathsf{Y} = \mathsf{D}_{k+1}\mathsf{U}_k\mathsf{U}^{k-i}$. Lemma 2.5.21 yields

$$\left\langle \mathsf{A}\mathsf{D}_{k+1}\mathsf{U}_k f_i, g \right\rangle \;=\; \lambda_i \cdot \langle \mathsf{A} f_i, g \rangle \;\pm\; (k-i+1)\cdot\gamma \,\|h_i\|\,\|g\|\,,$$

where $\lambda_i = Q_{k-i+2,i}(\vec{\delta})$. ∎

Then powers of the operator $\mathsf{D}_{k+1}\mathsf{U}_k$ behave as expected.

Lemma 2.5.23 (Exponentiation Lemma)**.**

$$\left\langle \left(\mathsf{D}_{k+1}\mathsf{U}_k\right)^s f_i, f_i \right\rangle \;=\; \lambda_i^s \cdot \|f_i\|^2 \;\pm\; s\cdot(k-i+1)\cdot\gamma\,\|h_i\|\,\|f_i\|\,,$$

where λ_i *is given in Lemma 2.5.22.*

Proof. Follows immediately from the foregoing and the fact that $\left\|\mathsf{D}_{k+1}\mathsf{U}_k\right\|_{\mathrm{op}} = 1$. ∎

In case $|\mathcal{D}| > |\mathcal{U}|$, $\mathsf{Y}\colon C^i \to C^j$ is an operator whose kernel approximately contains $\ker(\mathsf{D}_i)$ as the following lemma makes precise.

Lemma 2.5.24 (Refinement of [DDFH18]). *If* $|\mathcal{D}| > |\mathcal{U}|$ *and* $h_i \in \ker(\mathsf{D}_i)$, *then*

$$\langle \mathsf{A}\mathsf{Y}_\ell \ldots \mathsf{Y}_1 h_i, g\rangle \;=\; \pm\,\ell^2\cdot\gamma\,\|h_i\|\,\|g\|\,,$$

provided $\|\mathsf{A}\|_{\mathrm{op}} \leq 1$.

Proof. Let $c \in [\ell]$ be the smallest index for which Y_c is a down operator. Observe that $c < \ell/2$ since $|\mathcal{D}| > |\mathcal{U}|$. We induct on $m = |\mathcal{D}|$. If $c = 1$, then $\langle \mathsf{A}\mathsf{D}_i h_i, g\rangle = 0$. Hence assume $c, m > 1$ implying $\mathsf{Y}_c\mathsf{Y}_{c-1} = \mathsf{D}_{i+c}\mathsf{U}_{i+c-1}$. Applying Lemma 2.5.21 we obtain

$$\begin{aligned}
\langle \mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_1 h_i, g\rangle &= \left\langle (\mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_{c+1})\, \mathsf{D}\mathsf{U}\mathsf{U}^{c-2} h_i, g \right\rangle \\
&= Q_{c,i}(\vec{\delta}) \cdot \left\langle (\mathsf{A}\mathsf{Y}_\ell \dots \mathsf{Y}_{c+1})\, \mathsf{U}^{c-2} h_i, g \right\rangle \pm \frac{\ell}{2} \cdot \gamma \|h_i\| \|g\| \\
&= \pm Q_{c,i}(\vec{\delta}) \cdot (\ell-2)^2 \cdot \gamma \|h_i\| \|g\| \pm \frac{\ell}{2} \cdot \gamma \|h_i\| \|g\| \qquad \text{(Induction)} \\
&= \pm \ell^2 \cdot \gamma \|h_i\| \|g\|,
\end{aligned}$$

where in the last derivation we used $Q_{c,i}(\vec{\delta}) \le 1$. ∎

We turn to an important particular case of $|\mathcal{D}| = |\mathcal{U}|$, namely, the canonical walks. We show that $\mathsf{N}_{k,k}^{(u)}$ is approximately a polynomial in the operator $\mathsf{D}_{k+1}\mathsf{U}_k$. As a warm up consider the case $\mathsf{N}_{k,k}^{(2)} = \mathsf{D}_{k+1}\mathsf{D}_{k+2}\mathsf{U}_{k+1}\mathsf{U}_k$. Using the Eq. (2.3), we get

$$\begin{aligned}
\mathsf{N}_{k,k}^{(2)} &\approx (1-\delta_{k+1}) \cdot \mathsf{D}_{k+1}\mathsf{U}_k\mathsf{D}_{k+1}\mathsf{U}_k + \delta_{k+1} \cdot \mathsf{D}_{k+1}\mathsf{U}_k \\
&= (1-\delta_{k+1}) \cdot \left(\mathsf{D}_{k+1}\mathsf{U}_k\right)^2 + \delta_{k+1} \cdot \mathsf{D}_{k+1}\mathsf{U}_k.
\end{aligned}$$

Inspecting this polynomial more carefully we see that that its coefficients form a probability distribution. This property holds in general as the following Lemma 2.5.25 shows. This gives an alternative (approximate) random walk interpretation of $\mathsf{N}_{k,k}^{(u)}$ as the walk that first selects the power s according to the distribution encoded in the polynomial and then moves according to $\left(\mathsf{D}_{k+1}\mathsf{U}_k\right)^s$.

Lemma 2.5.25 (Canonical Polynomials)**.** *For $k, u \ge 0$ there exists a degree u univariate polynomial $F^N_{u,k,\vec{\delta}}$ depending only on $u, k, \vec{\delta}$ such that*

$$\left\| \mathsf{N}_{k,k}^{(u)} - F^N_{u,k,\vec{\delta}}(\mathsf{D}_{k+1}\mathsf{U}_k) \right\|_{\mathrm{op}} \le (u-1)^2 \cdot \gamma.$$

Moreover, the coefficients of this polynomial form a probability distribution, i.e., $F^{N}_{u,k,\vec{\delta}}(x) = \sum_{i=0}^{u} c_i x^i$ *where* $\sum_{i=0}^{u} c_i = 1$ *and* $c_i \geq 0$ *for* $i = 0, \ldots, u$.

Proof. For $u = 0$, $\mathsf{N}_{k,k}^{(0)} = \mathsf{I}$ and the lemma trivially follows. Similarly, if $u = 1$, $\mathsf{N}_{k,k}^{(1)} = \mathsf{D}_{k+1}\mathsf{U}_k$. Now suppose $u \geq 2$. Set $\mathsf{Y} = \mathsf{N}_{k,k}^{(u)}$, i.e.,

$$\mathsf{Y} \;=\; \mathsf{D}_{k+1} \ldots \left(\mathsf{D}_{k+u}\mathsf{U}_{k+u-1}\right) \ldots \mathsf{U}_k.$$

For convenience let $j = k + u - 1$. Using the Eq. (2.3) we can replace $\mathsf{D}_{j+1}\mathsf{U}_j$ in Y by $(1-\delta_j)\mathsf{U}_{j-1}\mathsf{D}_j + \delta_j \mathsf{I}$ incurring an error of γ (in spectral norm) and yielding

$$\mathsf{Y} \;\approx\; (1-\delta_j) \cdot \mathsf{Y}' \;+\; \delta_j \cdot \mathsf{N}_{k,k}^{(u-1)},$$

where Y' was obtained from Y by moving the rightmost occurence of a down operator (in this case D_{j+1}) one position to right. We continue this process of moving the rightmost occurrence of a down operator until the resulting operator is up to $(u-1)\cdot\gamma$ error

$$\alpha \cdot \mathsf{N}_{k,k}^{(u-1)} \left(\mathsf{D}_{k+1}\mathsf{U}_k\right) \;+\; \beta \cdot \mathsf{N}_{k,k}^{(u-1)},$$

where $\alpha = \prod_{i=k+1}^{j}(1-\delta_i)$ and $\beta = \sum_{i=k+1}^{j} \delta_i \prod_{i=k+1}^{j}(1-\delta_i)$. Since $\delta_i = \delta_i > 0$, α, β are non negative and form a probability distribution. Now the result follows from the induction hypothesis applied to $\mathsf{N}_{k,k}^{(u-1)}$. ∎

Remark 2.5.26. *Having a polynomial expression* $F^{N}_{u,k,\vec{\delta}}(\mathsf{D}_{k+1}\mathsf{U}_k) \approx \mathsf{N}_{k,k}^{(u)}$ *and knowing that* $\mathsf{S}_{k,k}$ *can be written as linear combination of canonical walks, we could deduce that* $\mathsf{S}_{k,k}$ *is also approximately a polynomial in* $\mathsf{D}_{k+1}\mathsf{U}_k$. *Using an error refined version of the Lemma 2.5.23 (showing that exponentiation of* $\mathsf{D}_{k+1}\mathsf{U}_k$ *behaves naturally), we could deduce the approximate spectrum of* $\mathsf{S}_{k,k}$. *We avoid this approach since it analysis introduces unnecessary error terms and we can understand quadratic forms of pure balanced operators directly.*

Remark 2.5.27. *The canonical polynomial $F^N_{u,k,\vec{\delta}}(\mathsf{D}_{k+1}\mathsf{U}_k)$ is used later in the error analysis that relates the norms $\|h_i\|$ and $\|f_i\|$ (Lemma 2.5.30).*

Now we consider Y where $|\mathcal{D}| = |\mathcal{U}|$ in full generality. We show how the quadratic form of Y behaves in terms of the approximate eigenspace decomposition $C^k = \sum_{i=0}^k C_i^k$.

Lemma 2.5.28 (Pure Balanced Walks)**.** *Suppose $\mathsf{Y} = \mathsf{Y}_\ell \dots \mathsf{Y}_1$ is a product of an equal number of up and down operators, i.e., $|\mathcal{D}| = |\mathcal{U}|$. Then for $f_i \in C_i^k$*

$$\langle \mathsf{Y} f_i, f_i \rangle \;=\; \lambda^{\mathsf{Y}}_{k,i} \cdot \langle f_i, f_i \rangle \;\pm\; \gamma \cdot (\ell^2 + \ell(k-i-1)) \, \|h_i\| \, \|f_i\| \,,$$

where $\lambda^{\mathsf{Y}}_{k,i}$ is an approximate eigenvalue depending only on Y, k and i.

Proof. We induct on even ℓ. For $\ell = 0$, the result trivially follows so assume $\ell \geq 2$. Let $c \in [\ell]$ be the smallest index of a down operator. Set $\mathsf{A} = \mathsf{Y}_\ell \dots \mathsf{Y}_{c+1}$ and let $\mathsf{Y}' = \mathsf{Y}_c \dots \mathsf{Y}_1 = \mathsf{D}\mathsf{U} \dots \mathsf{U}$. Observe that

$$\langle \mathsf{A}\mathsf{Y}' f_i, f_i \rangle \;=\; \left\langle \mathsf{A}\mathsf{D}\mathsf{U}^{c-1+k-i} h_i, f_i \right\rangle .$$

Applying Lemma 2.5.21 to the RHS above gives

$$\left\langle \mathsf{A}\mathsf{D}\mathsf{U}^{c-1+k-i} h_i, f_i \right\rangle = Q_{c-1+k-i,i}(\vec{\delta}) \cdot \left\langle \mathsf{A}\mathsf{U}^{c-2} f_i, f_i \right\rangle \;\pm\; (c+k-i-2) \cdot \gamma \, \|h_i\| \, \|f_i\| \,.$$

Applying the induction hypothesis to $\mathsf{Y}'' = A\mathsf{U}^{c-2}$ in the above RHS yields

$$\begin{aligned}
&Q_{c-1+k-i,i}(\vec{\delta}) \cdot \lambda^{\mathsf{Y}''}_{k,i} \, \langle f_i, f_i \rangle \\
&\pm\; Q_{c-1+k-i,i}(\vec{\delta}) \cdot \gamma \cdot ((\ell-1)^2 + (\ell-1)(k-i-1)) \, \|h_i\| \, \|f_i\| \\
&\pm\; (c+k-i-2) \cdot \gamma \, \|h_i\| \, \|f_i\| \\
&=\; \lambda^{\mathsf{Y}}_{k,i} \cdot \langle f_i, f_i \rangle \;\pm\; \gamma \cdot (\ell^2 + \ell(k-i-1)) \, \|h_i\| \, \|f_i\| \,,
\end{aligned}$$

where $\lambda^{\mathsf{Y}}_{k,i} = Q_{c-1+k-i,i}(\vec{\delta}) \cdot \lambda^{\mathsf{Y}''}_{k,i}$ and the last equality follows from $Q_{c-1+k-i,i}(\vec{\delta}) \leq 1$ and $c \leq \ell$. ∎

To understand all errors in the analysis in Lemma 2.5.28 we need to derive the approximate orthogonality of f_i and f_j for $i \neq j$ from [DDFH18] in more detail. We start with the following bound in terms of h_i, h_j.

Lemma 2.5.29 (Refinement of [DDFH18]). *For $i \neq j$,*

$$\left\langle f_i, f_j \right\rangle \;=\; \pm\,(2k-i-j)^2 \cdot \gamma \left\|h_i\right\| \left\|h_j\right\|.$$

Proof. Recall that $f_i = \mathsf{U}^{k-i}h_i$, $f_j = \mathsf{U}^{k-j}h_j$ where $h_i \in \ker(\mathsf{D}_i)$, $h_j \in \ker\left(\mathsf{D}_j\right)$. Without loss of generality suppose $i > j$. We have

$$\left\langle \mathsf{U}^{k-i}h_i, \mathsf{U}^{k-j}h_j \right\rangle \;=\; \left\langle \mathsf{D}^{k-j}\mathsf{U}^{k-i}h_i, h_j \right\rangle.$$

Since $k-j > k-i$, the result follows from Lemma 2.5.24. ∎

To give a bound for Lemma 2.5.29 only in terms of the eigenfunction norms $\|f_i\|$ and not in terms of $\|h_i\|$, we need to understand how the norms of h_i and f_i are related.

Lemma 2.5.30 (Refinement of [DDFH18]). *Let $\eta_{k,i} = (k-i)^2 + 1$ and let*

$$\beta_i = \sqrt{\left|F^N_{k-i,i,\vec{\delta}}(\delta_i) \pm \gamma \cdot \eta_{k,i}\right|},$$

where $F^N_{k-i,k,\vec{\delta}}$ is a canonical polynomial of degree $k-i$ from Lemma 2.5.25. Then

$$\langle f_i, f_i \rangle \;=\; \beta_i^2 \cdot \langle h_i, h_i \rangle.$$

Let $\theta_{k,i} = (i+1)^{k-i}$. Furthermore, if $\gamma \leq 1/(2 \cdot \eta_{k,i} \cdot \theta_{k,i})$, then $\beta_i \geq \frac{1}{2\theta_{k,i}}$.

Proof. Recall that $f_i = \mathsf{U}^{k-i} h_i$ where $h_i \in \ker(\mathsf{D}_i)$. For $i = k$ the result trivially follows so assume $k > i$. First consider the case $k = i + 1$. We have

$$\langle \mathsf{U}_i h_i, \mathsf{U}_i h_i \rangle \;=\; \langle \mathsf{D}_{i+1} \mathsf{U}_i h_i, h_i \rangle \;=\; \delta_i \cdot \langle h_i, h_i \rangle \;\pm\; \gamma \cdot \langle h_i, h_i \rangle . \tag{2.4}$$

For general $k > i$ we have

$$\left\langle \mathsf{U}^{k-i} h_i, \mathsf{U}^{k-i} h_i \right\rangle \;=\; \left\langle \mathsf{D}^{k-i} \mathsf{U}^{k-i} h_i, h_i \right\rangle .$$

Applying Lemma 2.5.25 to $\mathsf{D}^{k-i}\mathsf{U}^{k-i}$ yields

$$\left\langle \mathsf{D}^{k-i} \mathsf{U}^{k-i} h_i, h_i \right\rangle \;=\; \left\langle F^N_{k-i,i,\vec{\delta}}(\mathsf{D}_{i+1}\mathsf{U}_i) h_i, h_i \right\rangle \;\pm\; \gamma \cdot (k-i-1)^2.$$

Combining Eq. (2.4) and Lemma 2.5.23 gives

$$\left\langle F^N_{k-i,i,\vec{\delta}}(\mathsf{D}_{i+1}\mathsf{U}_i) h_i, h_i \right\rangle \;\pm\; \gamma \cdot (k-i-1)^2 \;=\; \left\langle F^N_{k-i,i,\vec{\delta}}(\delta_i) h_i, h_i \right\rangle \;\pm\; \gamma \cdot ((k-i)^2 + 1).$$

Since $F^N_{k-i,i,\vec{\delta}}(x) = \sum_{i=0}^{k-i} c_i x^i$ where the coefficients c_i form a probability distribution, we get

$$F^N_{k-i,i,\vec{\delta}}(\delta_i) \;\geq\; \delta_i^{k-i} \;=\; \left(\frac{1}{i+1}\right)^{k-i}.$$

■

Now, we can state the approximate orthogonality Lemma 2.5.31 in terms of the eigenfunction norms.

Lemma 2.5.31 (Approximate Orthogonality (refinement of [DDFH18]))**.** *Let $\eta_{k,s}, \theta_{k,s}, \beta_s$ for*

$s \in \{i, j\}$ *be given as in Lemma 2.5.30. If* $i \neq j$ *and* $\beta_i, \beta_j > 0$*, then*

$$\left\langle f_i, f_j \right\rangle \;=\; \pm\, \frac{\gamma \cdot (2k - i - j)^2}{\beta_i \beta_j} \left\| f_i \right\| \left\| f_j \right\|.$$

Furthermore, if $\gamma \leq \min\left(1/(2 \cdot \eta_{k,i} \cdot \theta_{k,i}), 1/(2 \cdot \eta_{k,j} \cdot \theta_{k,j})\right)$*, then* $\beta_i, \beta_j > 0$ *and*

$$\left\langle f_i, f_j \right\rangle \;=\; \pm\, \gamma \cdot \theta_{k,i} \cdot \theta_{k,j} \cdot (2k - i - j)^2 \left\| f_i \right\| \left\| f_j \right\|.$$

Proof. Follows directly from Lemma 2.5.30. ∎

We generalize the quadratic form of Lemma 2.5.28 to linear combinations of general *pure balanced* operators Y, namely, to *balanced* operators.

Lemma 2.5.32 (General Quadratic Form (restatement of Lemma 2.5.8))**.** *Let* $\varepsilon \in (0, 1)$ *and let* $\mathcal{Y} \subseteq \{\mathsf{Y} \mid \mathsf{Y} \colon C^k \to C^k\}$ *be a collection of formal operators that are product of an equal number of up and down walks (i.e., pure balanced operators) not exceeding* ℓ *walks. Let* $\mathsf{B} = \sum_{\mathsf{Y} \in \mathcal{Y}} \alpha^{\mathsf{Y}} \mathsf{Y}$ *where* $\alpha^{\mathsf{Y}} \in \mathbb{R}$ *and let* $f = \sum_{i=0}^{k} f_i$ *with* $f_i \in C_i^k$*. If* $\gamma \leq \varepsilon \left(16 k^{k+2} \ell^2 \sum_{\mathsf{Y} \in \mathcal{Y}} |\alpha^{\mathsf{Y}}|\right)^{-1}$*, then*

$$\langle \mathsf{B} f, f \rangle \;=\; \sum_{i=0}^{k} \left(\sum_{\mathsf{Y} \in \mathcal{Y}} \alpha^{\mathsf{Y}} \lambda_k^{\mathsf{Y}}(i) \right) \cdot \langle f_i, f_i \rangle \;\pm\; \varepsilon,$$

where $\lambda_k^{\mathsf{Y}}(i)$ *depends only on the operators appearing in the formal expression of* Y*,* i *and* k*, i.e.,* $\lambda_k^{\mathsf{Y}}(i)$ *is the approximate eigenvalue of* Y *associated to* C_i^k*.*

Proof. Using Lemma 2.5.28 and the assumption on γ gives

$$\begin{aligned}\langle \mathsf{B}f, f\rangle &= \sum_{i=0}^{k}\sum_{Y\in\mathcal{Y}} \alpha^{Y}\lambda_k^{Y}(i)\cdot\langle f_i, f_i\rangle \\ &\quad + \sum_{i\neq j}\sum_{Y\in\mathcal{Y}}\left(\alpha^{Y}\lambda_k^{Y}(i)\cdot\left\langle f_i, f_j\right\rangle \pm \gamma\cdot\alpha^{Y}(\ell^2+\ell(k-i-1))\left\langle h_i, f_j\right\rangle\right) \\ &= \sum_{i=0}^{k}\sum_{Y\in\mathcal{Y}} \alpha^{Y}\lambda_k^{Y}(i)\cdot\langle f_i, f_i\rangle + \sum_{i\neq j}\sum_{Y\in\mathcal{Y}}\alpha^{Y}\lambda_k^{Y}(i)\cdot\left\langle f_i, f_j\right\rangle \pm \frac{\varepsilon}{2}.\end{aligned}$$

Next we use Lemma 2.5.31 to bound the second double summation and conclude the proof. ∎

We instantiate Lemma 2.5.31 for swap walks with their specific parameters. First, we introduce some notation. Using Corollary 2.4.13, we have

$$\mathsf{S}_{k,k} = \sum_{j=0}^{k}(-1)^{k-j}\cdot\binom{k+j}{k}\cdot\binom{k}{j}\cdot\mathsf{N}_{k,k}^{(j)} = \sum_{j=0}^{k}\alpha_j\cdot\mathsf{N}_{k,k}^{(j)},$$

where $\alpha_j = (-1)^{k-j}\cdot\binom{k+j}{k}\cdot\binom{k}{j}$.

Finally, we have all the pieces to prove Lemma 2.5.6 restated below.

Lemma 2.5.33 (Swap Quadratic Form (restatement of Lemma 2.5.6))**.** *Let $f = \sum_{i=0}^{k} f_i$ with $f_i \in C_i^k$. Suppose $X(\leq d)$ is a γ-HDX with $d \geq 2k$. If $\gamma \leq \varepsilon\left(64k^{k+4}2^{3k+1}\right)^{-1}$, then*

$$\left\langle \mathsf{S}_{k,k}f, f\right\rangle = \sum_{i=0}^{k}\lambda_k(i)\cdot\langle f_i, f_i\rangle \pm \varepsilon,$$

where $\lambda_k(i)$ depends on only on k an i, i.e., $\lambda_k(i)$ is an approximate eigenvalue of $\mathsf{S}_{k,k}$ associated to space C_i^k.

Proof. First note that Lemma 2.5.28 establishes the existence of approximate eigenvalues $\lambda_{k,j}(i)$ of $\mathsf{N}_{k,k}^{(j)}$ corresponding to space C_i^k for $i = 0,\ldots,k$ such that $\lambda_{k,j}(i)$ depends only on

k, i and j. To apply Lemma 2.5.8 we need to bound $\sum_{j=0}^{k} |\alpha_j|$. Since

$$\sum_{j=0}^{k} |\alpha_j| \;=\; \sum_{j=0}^{k} \binom{k+j}{k} \cdot \binom{k}{j} \;\leq\; 2^k \cdot \sum_{j=0}^{k} \binom{k+j}{k} \;\leq\; 2^{3k+1},$$

we are done. ∎

2.5.4 *Rectangular Swap Walks $\mathsf{S}_{k,l}$*

We turn to the spectral analysis of rectangular swap walks, i.e., $\mathsf{S}_{k,l}$ where $k \neq l$. Recall that to bound $\sigma_2(\mathsf{S}_{k,k})$ in Section 2.5.1 we proved that the spectrum of $\mathsf{S}_{k,k}$ for a γ-HDX is close to the spectrum of $\mathsf{S}^{\Delta}_{k,k}$ using the analysis of quadratic forms over *balanced* operators from Section 2.5.3. Then we appealed to the fact that $\mathsf{S}^{\Delta}_{k,k}$ is expanding since it is the walk operator of the well known Kneser graph. In this rectangular case, we do not have a classical result establishing that $\mathsf{S}^{\Delta}_{k,l}$ is expanding, but we were able to establish it Lemma 2.5.34.

Lemma 2.5.34. *Let $d \geq k+l$ and $\Delta_d(n)$ be the complete complex. The second largest singular value $\sigma_2(\mathsf{S}^{\Delta}_{k,l})$ of the swap operator $\mathsf{S}^{\Delta}_{k,l}$ on $\Delta_d(n)$ is*

$$\sigma_2(\mathsf{S}^{\Delta}_{k,l}) \;\leq\; \max\left(\frac{k}{n-k}, \frac{l}{n-l}\right),$$

provided $n \geq M_{k,l}$ where $M_{k,l} \in \mathbb{N}$ only depends on k and l.

Towards proving Lemma 2.5.34 we first introduce a generalization of Kneser graphs which we denote *bipartite Kneser* graphs defined as follows.

Definition 2.5.35 (General Bipartite Kneser Graph)**.** *Let $X(\leq d)$ where $d \geq k+l$. We denote by $K^X(n,k,l)$ the bipartite graph on (vertex) partition $(X(k), X(l))$ where $\mathfrak{s} \in X(k)$ is adjacent to $\mathfrak{t} \in X(l)$ if and only if $\mathfrak{s} \cap \mathfrak{t}$ is empty. We also refer to graphs of the form $K^X(n,k,l)$ as bipartite*

Kneser graphs.

It will be convenient to distinguish *bipartite Kneser* graphs coming from general γ-HDX and the complete complex $\Delta_d(n)$.

Definition 2.5.36 (Complete Bipartite Kneser Graph)**.** *Let $X(\leq d)$ where $d \geq k+l$. If X is the complete complex, i.e., $X = \Delta_d(n)$, then we denote $K^X(n,k,l)$ as simply as $K(n,k,l)$ and we refer to it as complete bipartite Kneser.*

We obtain the spectra of *bipartite Kneser* graphs generalizing [6] the classical result of Fact 2.5.4. More precisely, we prove Lemma 2.5.37.

Lemma 2.5.37 (Bipartite Kneser Spectrum)**.** *The non-zero eigenvalues of the (normalized) walk operator of $K(n,k,l)$ are $\pm\lambda_i$ where*

$$\lambda_i \;=\; \frac{\binom{n-k-i}{l-i}\binom{n-l-i}{k-i}}{\binom{n-k}{l}\binom{n-l}{k}},$$

for $i = 0,\ldots,\min(k,l)$.

Now the proof follows a similar strategy to the $\mathsf{S}_{k,k}$, namely, we analyze quadratic forms over $\mathsf{S}_{k,k}$ using the results from Section 2.5.3

Let $X(\leq d)$ where $d \geq k+l$. Let $\mathsf{A}_{k,l}$ be the (normalized) walk operator of $K^X(n,k,l)$, i.e.,

$$\mathsf{A}_{k,l} \;=\; \begin{pmatrix} 0 & \mathsf{S}^{(l)}_{k,l} \\ \left(\mathsf{S}^{(l)}_{k,l}\right)^\dagger & 0 \end{pmatrix}.$$

To determine the spectrum of $\mathsf{A}_{k,l}$ it is enough to consider the spectrum of $\mathsf{B} = \mathsf{S}^{(l)}_{k,l}\left(\mathsf{S}^{(l)}_{k,l}\right)^\dagger$.

6. Note that the singular values of $K(n,k)$ can be deduced from the bipartite case.

Using Corollary 2.4.13, we have

$$\mathsf{B} = \left(\sum_{j=0}^{l}(-1)^{l-j}\binom{k+j}{l}\cdot\binom{l}{j}\cdot\mathsf{N}_{k,l}^{(j)}\right) \left(\sum_{j'=0}^{l}(-1)^{l-j'}\binom{k+j'}{l}\cdot\binom{l}{j'}\cdot\left(\mathsf{N}_{k,l}^{(j')}\right)^{\dagger}\right) = \sum_{j,j'=0}^{l}\alpha_{k,l,j,j'}\mathsf{N}_{k,l}^{(j)}\mathsf{N}_{l,k}^{(j'+k-l)},$$

for some coefficients $\alpha_{k,l,j,j'}$ depending only on k, l, i, j and j'. Since we have not yet used any specific property of HDXs, these coefficients are the same for the complete complex and general HDXs.

Lemma 2.5.38. *Let $X(\leq d)$ be a γ-HDX with $d \geq k+l$. Let $f = \sum_{i=0}^{k} f_i$ with $f_i \in C_i^k$. For $\varepsilon \in (0,1)$, if $\gamma \leq \varepsilon\left(64k^{k+2}\ell^2 2^{2k+4l+2}\right)^{-1}$, then*

$$\langle \mathsf{B}f, f\rangle = \sum_{i=0}^{k}\left(\sum_{j,j'=0}^{l}\alpha_{k,l,j,j'}\lambda_{k,l,j,j'}(i)\right)\cdot\langle f_i, f_i\rangle + \varepsilon,$$

where $\lambda_{k,l,j,j'}(i)$ is the approximate eigenvalues of $\mathsf{N}_{k,l}^{(j)}\mathsf{N}_{l,k}^{(j'+k-l)}$ corresponding to space C_i^k. Furthermore, $\lambda_{k,l,j,j'}(i)$ depends only on k, l, i, j and j'.

Proof. First observe that each $\mathsf{N}_{k,l}^{(j)}\mathsf{N}_{l,k}^{(j'+k-l)}$ maps C^k to itself, so it is a product of the same number of up and down operators. Now to apply Lemma 2.5.8 it only remains to bound $\sum_{j,j'=0}^{l}|\alpha_{k,l,j,j'}|$. Since

$$\sum_{j,j'=0}^{l}|\alpha_{k,l,j,j'}| = \sum_{j,j'=0}^{l}\binom{k+j}{l}\cdot\binom{l}{j}\binom{k+j'}{l}\cdot\binom{l}{j'} \leq 2^{2l}\left(\sum_{j=0}^{l}\binom{k+j}{l}\right)\cdot\left(\sum_{j'=0}^{l}\binom{k+j'}{l}\right) \leq 2^{2k+4l+2},$$

we are done. ∎

Let B and B^{Δ} stand for the B operator for general γ-HDX and the complete complex,

respectively.

Lemma 2.5.39. *Suppose $X(\leq d)$ is a γ-HDX with $d \geq k+l$. For $\varepsilon \in (0,1)$, if we have $\gamma \leq \varepsilon^2 \left(64k^{k+2}\ell^2 2^{2k+4l+2}\right)^{-1}$, then the second largest singular value $\sigma_2(\mathsf{B})$ of B is*

$$\sigma_2(\mathsf{B}) \;\leq\; \varepsilon^2.$$

Furthermore, the second largest non-trivial eigenvalue $\lambda(\mathsf{A}_{k,l})$ of the walk matrix of $K(n,k,l)$ is

$$\lambda(\mathsf{A}_{k,l}) \;\leq\; \varepsilon.$$

Proof. The proof follows the same strategy of Theorem 2.5.1, namely, we first consider B^{Δ} and show that $\sum_{j,j'=0}^{l} \alpha_{k,l,j,j'}\lambda_{k,l,j,j'}(i) = 0$. Using Lemma 2.5.34, we deduce that

$$\left|\sum_{j,j'=0}^{l} \alpha_{k,l,j,j'}\lambda_{k,l,j,j'}(i)\right| \;=\; O_{k,l}\left(\frac{1}{n^2}\right)$$

for $i \in [k]$ where in this range each C_i^k is not the trivial approximate eigenspace (associated with eigenvalue 1). Since $\alpha_{k,l,j,j'}$ and $\lambda_{k,l,j,j'}(i)$ do not depend on n and n is arbitrary, the LHS above is actually zero. Then our choice of γ Lemma 2.5.8 gives

$$\max_{f\in C^k:\, f\perp 1, \|f\|=1} |\langle \mathsf{B}f, f\rangle| \;\leq\; \max_{i\in[k]} \left|\sum_{j,j'=0}^{l} \alpha_{k,l,j,j'}\lambda_{k,l,j,j'}(i)\right| \;+\; \varepsilon^2 \;=\; \varepsilon^2.$$

■

Now the proof of Theorem 2.5.2 follows. For convenience, we restate it.

Theorem 2.5.40 (Rectangular Swap Walk Spectral Bound (restatement of Theorem 2.5.2))**.** *Suppose $X(\leq d)$ is a γ-HDX with $d \geq k+l$ and $k \leq l$. For $\sigma \in (0,1)$, if $\gamma \leq \sigma^2 \cdot \left(64k^{k+2}\ell^2 2^{2k+4l+2}\right)^{-1}$, then the largest non-trivial singular value $\sigma_2(\mathsf{S}_{k,l})$ of the swap oper-*

ator $\mathsf{S}_{k,l}$ *is*

$$\sigma_2(\mathsf{S}_{k,l}) \;\leq\; \sigma.$$

Proof. Follows directly from Lemma 2.5.39. ∎

2.5.5 *Bipartite Kneser Graphs - Complete Complex*

Now we determine the spectrum of the *complete bipartite Kneser* graph $K(n,k,l)$. More precisely, we prove the following.

Lemma 2.5.41 (Bipartite Kneser Spectrum (restatement of Lemma 2.5.37))**.** *The non-zero eigenvalues of the normalized walk operator of* $K(n,k,l)$ *are* $\pm\lambda_i$ *where*

$$\lambda_i \;=\; \frac{\binom{n-k-i}{l-i}\binom{n-l-i}{k-i}}{\binom{n-k}{l}\binom{n-l}{k}},$$

for $i = 0,\ldots,\min(k,l)$.

Henceforth, set $X = \Delta_d(n)$. To prove Lemma 2.5.37 we work with the natural rectangular matrix associated with $K(n,k,l)$, namely, the matrix $W \in \mathbb{R}^{X(k)\times X(l)}$ such that

$$\mathsf{W}(\mathfrak{s},\mathfrak{t}) \;=\; \mathbb{1}_{[\mathfrak{s}\cap\mathfrak{t}=\emptyset]}$$

for every $\mathfrak{s} \in X(k)$ and $\mathfrak{t} \in X(l)$.

Observe that the entries of $\mathsf{W}\mathsf{W}^\top$ and $\mathsf{W}^\top\mathsf{W}$ only depend on the size of the intersection of the sets indexing the row and columns. Hence, these matrices belong to the Johnson scheme [GM15] $J(n,k)$ and $J(n,l)$, respectively. Moreover, the left and right singular vectors of W are eigenvectors of these schemes.

We adopt the eigenvectors used in Filmus' work [Fil16], i.e., natural basis vectors coming from some irreducible representation of S_n (see [Sag13]). First we introduce some

notation. Let $\mu = (n-i, i)$ be a partition of n and let τ_μ be a standard tableau of shape μ. Suppose the first row τ_μ contains $a_1 < \cdots < a_{n-i}$ whereas the second contains $b_1 < \cdots < b_i$. To τ_μ we associate the function $\varphi_{\tau_\mu} \in \mathbb{R}^{\binom{[n]}{k}}$ as follows

$$\varphi_{\tau_\mu} \;=\; (\mathbb{1}_{a_1} - \mathbb{1}_{b_1}) \dots (\mathbb{1}_{a_i} - \mathbb{1}_{b_i}),$$

where $\mathbb{1}_a \in \mathbb{R}^{\binom{n}{k}}$ is the containment indicator of element a, i.e., $\mathbb{1}_a(\mathfrak{s}) = 1$ if and only if $a \in \mathfrak{s}$. Filmus proved that

$$\left\{ \varphi_{\tau_\mu} \mid 0 \le i \le k, \mu \vdash (n-i, i), \tau_\mu \text{ standard} \right\}$$

is an eigenbasis of $\mathcal{J}(n,k)$. We abuse the notation by considering φ_{τ_μ} as both a function in $\mathbb{R}^{\binom{n}{k}}$ and $\mathbb{R}^{\binom{n}{l}}$ as long as these functions are well defined.

Claim 2.5.42. *If $\mu = (n-i, i)$ and $k, l \ge i$, then*

$$\mathsf{W}\varphi_{\tau_\mu} \;=\; (-1)^i \cdot \binom{n-k-i}{l-i} \cdot \varphi_{\tau_\mu}.$$

Proof. We follow a similar strategy of Filmus. For convenience suppose $\varphi_{\tau_\mu} = (\mathbb{1}_1 - \mathbb{1}_2) \dots (\mathbb{1}_{2i-1} - \mathbb{1}_{2i})$. For $i = 0$ the claim follows immediately so assume $i \ge 1$. Consider $\left(\mathsf{W}\varphi_{\tau_\mu}\right)(\mathfrak{s})$ where $\mathfrak{s} \in \binom{[n]}{k}$. Note that

$$\left(\mathsf{W}\varphi_{\tau_\mu}\right)(\mathfrak{s}) \;=\; \sum_{\mathfrak{t} \in Y \,:\, \mathfrak{s} \cap \mathfrak{t} = \emptyset} \varphi_{\tau_\mu}(\mathfrak{t}).$$

If $2j-1, 2j \in \mathfrak{s}$ for some $j \in [i]$, then $2j-1, 2j \notin \mathfrak{t}$ so $\varphi_{\tau_\mu}(\mathfrak{s}) = 0 = \left(W\varphi_{\tau_\mu}\right)(\mathfrak{s})$. If $2j-1, 2j \notin \mathfrak{s}$ for some $j \in [i]$, for each $\mathfrak{t}$ adjacent to $\mathfrak{s}$ there four cases: $2j-1, 2j \in \mathfrak{t}$, $2j-1, 2j \notin \mathfrak{t}$, $2j-1 \in \mathfrak{t}$ and $2j \notin \mathfrak{t}$ or vice-versa. The first two cases yield $\varphi_{\tau_\mu}(\mathfrak{t}) = 0$ while the last two cases cancel each other in the summation and again $\varphi_{\tau_\mu}(\mathfrak{s}) = 0 =$

$\left(W\varphi_{\tau_\mu}\right)(\mathfrak{s})$. Now suppose that $\mathfrak{s}$ contains exactly one element of each pair $2j-1, 2j$. For any adjacent $\mathfrak{t}$ to yield $\varphi_{\tau_\mu}(\mathfrak{t}) \neq 0$, $\mathfrak{t}$ must contain $[2i] \setminus \mathfrak{s}$. Since there are $\binom{n-k-i}{l-i}$ such possibilities for $\mathfrak{t}$ we obtain

$$\mathsf{W}\varphi_{\tau_\mu} \;=\; (-1)^i \cdot \binom{n-k-i}{l-i} \cdot \varphi_{\tau_\mu},$$

where the sign $(-1)^i$ follows from the product of the signs of each the i pairs and the fact that $\mathfrak{s}$ and $\mathfrak{t}$ partition the elements in each pair. ∎

Since we are working with singular vectors, we need to be careful with their normalization when deriving the singular values. We stress that the norm of φ_{τ_μ} depends on the space where φ_{τ_μ} lies.

Claim 2.5.43. *If $\mu = (n-i, i)$ and $\varphi_{\tau_\mu} \in \mathbb{R}^{\binom{n}{k}}$, then*

$$\left\|\varphi_{\tau_\mu}\right\|_2 \;=\; \sqrt{2^i\binom{n-2i}{k-i}}.$$

Proof. Since φ_{τ_μ} assumes values in $\{-1, 0, 1\}$ so its enough to count the number of sets $\mathfrak{s} \in \binom{[n]}{k}$ such that $\varphi_{\tau_\mu}(\mathfrak{s}) \neq 0$. To have $\varphi_{\tau_\mu}(\mathfrak{s}) \neq 0$, $\mathfrak{s}$ must contain exactly one element in each pair and the remaining $k-i$ elements of $\mathfrak{s}$ can be chosen arbitrarily among the elements avoiding the $2i$ elements appearing in the indicators defining φ_{τ_μ}. ∎

Now the singular values of W follow.

Corollary 2.5.44 (Singular Values)**.** *The singular values of W are*

$$\sigma_i \;=\; \binom{n-k-i}{l-i} \cdot \frac{\left\|\varphi^k_{\tau_\mu}\right\|_2}{\left\|\varphi^l_{\tau_\mu}\right\|_2},$$

for $i = 0, \ldots, \min(k, l)$.

Note that for $k = l$ we recover the well know result of Fact 2.5.4.

Finally we compute the eigenvalues of the bipartite graph $K(n,k,l)$. Let $\mathsf{A}_{n,k,l}$ be its normalized adjacency matrix, i.e.,

$$\mathsf{A}_{n,k,l} = \begin{pmatrix} 0 & \frac{1}{\binom{n-k}{l}}\mathsf{W} \\ \frac{1}{\binom{n-l}{k}}\mathsf{W}^\top & 0 \end{pmatrix}.$$

Lemma 2.5.45 (Bipartite Kneser Spectrum (restatement of Lemma 2.5.37))**.** *The non-zero eigenvalues of the normalized walk operator of $K(n,k,l)$ are $\pm\lambda_i$ where*

$$\lambda_i = \frac{\binom{n-k-i}{l-i}\binom{n-l-i}{k-i}}{\binom{n-k}{l}\binom{n-l}{k}},$$

for $i = 0,\dots,\min(k,l)$.

Proof. Since the spectrum of a bipartite graph is symmetric around zero, it is enough to compute the eigenvalues of $A^2_{n,k,l}$. Set $\alpha = 1/\binom{n-k}{l}\binom{n-l}{k}$. Moreover, we consider $\alpha \cdot \mathsf{W}\mathsf{W}^\top$ since $\alpha \cdot \mathsf{W}^\top\mathsf{W}$ has the same non-zero eigenvalues. The non-zero eigenvalues of $\alpha \cdot \mathsf{W}\mathsf{W}^\top$ are

$$\lambda_i = \frac{\binom{n-k-i}{l-i}\binom{n-l-i}{k-i}}{\binom{n-k}{l}\binom{n-l}{k}},$$

for $i = 0,\dots,\min(k,l)$. ∎

2.6 Approximating Max-k-CSP

In the following, we will show that k-CSP instances $\mathfrak{I}$ whose constraint complex $X_{\mathfrak{I}}(\leq k)$ is a suitable expander admit an efficient approximation algorithm. We will assume throughout that $X_{\mathfrak{I}}(1) = [n]$, and drop the subscript $\mathfrak{I}$.

This was shown for 2-CSPs in [BRS11]. In extending this result to k-CSPs we will

rely on a central Lemma of their paper. Before, we explain our algorithm we give a basic outline of our idea:

We will work with the SDP relaxation for the k-CSP problem given by L-levels of SoS hierarchy, as defined in Section 2.2.4 (for L to be specified later). This will give us an L-local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$, which attains some value $\mathsf{SDP}(\mathfrak{I}) \geq \mathsf{OPT}(\mathfrak{I})$. Since $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$, is a local PSD ensemble, and not necessarily a probability distribution, we cannot sample from it directly. Nevertheless, since $\left\{\mathbf{Y}_j\right\}$ will be actual probability distributions for all $j \in [n]$, one can independently sample $\sigma_j \sim \left\{\mathbf{Y}_j\right\}$ and use $\sigma = (\sigma_1, \ldots, \sigma_n)$ as the assignment for the k-CSP instance $\mathfrak{I}$.

Unfortunately, while we know that the local distributions $\{\mathbf{Y}_\mathfrak{a}\}_{\mathfrak{a} \in X(k)}$ induced by $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ will satisfy the constraints of $\mathfrak{I}$ with good probability, i.e.,

$$\mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \mathop{\mathbb{E}}_{\{\mathbf{Y}_\mathfrak{a}\}} \left[\mathbf{1}[\underbrace{\mathbf{Y}_a \text{ satisfies the constraint on } \mathfrak{a}}_{\iff \mathbf{Y}_\mathfrak{a} \in \mathcal{C}_\mathfrak{a}}] \right] = \mathsf{SDP}(\mathfrak{I}) \geq \mathsf{OPT}(\mathfrak{I}),$$

this might not be the case for the assignment σ sampled as before. It might be that the random variables $\mathbf{Y}_{a_1}, \ldots, \mathbf{Y}_{a_k}$ are highly correlated for $\mathfrak{a} \in X(k)$, i.e.,

$$\mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \left\| \{\mathbf{Y}_\mathfrak{a}\} - \{\mathbf{Y}_{a_1}\} \cdots \{\mathbf{Y}_{a_k}\} \right\|_1$$

is large. One strategy employed by [BRS11] to ensure that the quantity above is small, is making the local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ be consistent with a randomly sampled partial assignment for a small subset of variables (q.v. Section 2.2.4). We will show that this strategy is succesful if $X(\leq k)$ is a γ-HDX (for γ sufficiently small). Our final algorithm will be the following,

Algorithm 2.6.1 (Propagation Rounding Algorithm)**.**

Input *An L-local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ and some distribution Π on $X(\leq \ell)$.*

Output *A random assignment $\sigma : [n] \to [q]$.*

1. *Choose $m \in \{1, \ldots, L/\ell\}$ uniformly at random.*
2. *Independently sample m ℓ-faces, $\mathfrak{s}_j \sim \Pi$ 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 $\sigma_S : S \to [q]$ according to the local distribution, $\{\mathbf{Y}_S\}$.*
5. *Set $\mathbf{Y}' = \{\mathbf{Y}_1, \ldots \mathbf{Y}_n | \mathbf{Y}_S = \sigma_S\}$, i.e. the local ensemble $\mathbf{Y}$ conditioned on agreeing with σ_S.*
6. *For all $j \in [n]$, sample independently $\sigma_j \sim \{\mathbf{Y}'_j\}$.*
7. *Output $\sigma = (\sigma_1, \ldots, \sigma_n)$.*

In our setting, we will apply Algorithm 4.7.16 with the distribution Π_k and the L-local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$. Notice that in expectation, the marginals of $\mathbf{Y}'$ on faces $\mathfrak{a} \in X(k)$ – which are actual distributions – will agree with the marginals of $\mathbf{Y}$, i.e. $\mathbb{E}_{S,\eta_S} \mathbb{E}\, \mathbf{Y}'_{\mathfrak{a}} = \mathbb{E}\, \mathbf{Y}_{\mathfrak{a}}$. In particular, the approximation quality of Algorithm 4.7.16 will depend on the average correlation of $\mathbf{Y}'_{a_1}, \ldots, \mathbf{Y}'_{a_k}$ on the constraints $\mathfrak{a} \in X(k)$, where $\mathbf{Y}'$ is the local PSD ensemble obtained at the end of the first phase of Algorithm 4.7.16.

In the case where $k = 2$, the following is known

Theorem 2.6.2 (Theorem 5.6 from [BRS11])**.** *Suppose a undirected graph $G = ([n], E, \Pi_2)$ and an L-local PSD ensemble $Y = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ are given. There exists absolute constants $c \geq 0$ and $C \geq 0$ satisfying the following: If $L \geq c \cdot \frac{q}{\varepsilon^4}$, $\mathrm{Supp}(\mathbf{Y}_i) \leq q$ for all $i \in V$, and $\lambda_2(G) \leq C \cdot \varepsilon^2/q^2$ then we have*

$$\mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2} \left\| \{\mathbf{Y}'_i, \mathbf{Y}'_j\} - \{\mathbf{Y}'_i\}\{\mathbf{Y}'_j\} \right\|_1 \leq \varepsilon,$$

where $\mathbf{Y}'$ *is as defined in Algorithm 4.7.16 on the input of* $\{\mathbf{Y}_1, \dots, \mathbf{Y}_n\}$ *and* Π_1.

To approximate k-CSPs well, we will show the following generalization of Theorem 2.6.2 for k-CSP instances $\mathfrak{I}$, whose constraint complex $X(\leq k)$ is γ-HDX, for γ sufficiently small.

Theorem 2.6.3. *Suppose a simplicial complex* $X(\leq k)$ *with* $X(1) = [n]$ *and an L-local PSD ensemble* $\mathbf{Y} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_n\}$ *are given.*

There exists some universal constants $c' \geq 0$ *and* $C' \geq 0$ *satisfying the following: If* $L \geq c' \cdot (q^k \cdot k^5/\varepsilon^4)$, $\mathrm{Supp}(\mathbf{Y}_j) \leq q$ *for all* $j \in [n]$, *and* X *is a* γ*-HDX for* $\gamma \leq C' \cdot \varepsilon^4/(k^{8+k} \cdot 2^{6k} \cdot q^{2k})$. *Then, we have*

$$\mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \left\| \{\mathbf{Y}'_{\mathfrak{a}}\} - \left\{\mathbf{Y}'_{a_1}\right\} \cdots \left\{\mathbf{Y}'_{a_k}\right\} \right\|_1 \leq \varepsilon, \tag{2.5}$$

where $\mathbf{Y}'$ *is as defined in Algorithm 4.7.16 on the input of* $\{\mathbf{Y}_1, \dots, \mathbf{Y}_n\}$ *and* Π_k.

Indeed, using Theorem 2.6.3, it will be straightforward to prove the following,

Corollary 2.6.4. *Suppose* $\mathfrak{I}$ *is a* q*-ary* k*-CSP instance whose constraint complex* $X(\leq k)$ *is a* γ*-HDX.*

There exists absolute constants $C' \geq 0$ *and* $c' \geq 0$, *satisfying the following: If* $\gamma \leq C' \cdot \varepsilon^4/(k^{8+k} \cdot 2^{6k} \cdot q^{2k})$, *there is an algorithm that runs in time* $n^{O(k^5 \cdot q^{2k} \cdot \varepsilon^{-4})}$ *based on* $(\frac{c' \cdot k^5 \cdot q^k}{\varepsilon^4})$*-levels of SoS-hierarchy and Algorithm 4.7.16 that outputs a random assignment* $\sigma : [n] \to [q]$ *that in expectation ensures* $\mathsf{SAT}_{\mathfrak{I}}(\sigma) = \mathsf{OPT}(\mathfrak{I}) - \varepsilon$.

Proof of Corollary 2.6.4. The algorithm will just run Algorithm 4.7.16 on the local PSD-ensemble $\{\mathbf{Y}_1, \dots, \mathbf{Y}_n\}$ given by the SDP relaxation of $\mathfrak{I}$ strengthened by $L = c' \cdot \frac{k^5 \cdot q^{2k}}{\varepsilon^4}$-levels of SoS-hierarchy and Π_k – where $c' \geq 0$ is the constant from Theorem 2.6.3. $\mathbf{Y}$ satisfies,

$$\mathsf{SDP}(\mathfrak{I}) = \mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \left[\mathop{\mathbb{E}}_{\{\mathbf{Y}_{\mathfrak{a}}\}} [\mathbf{1}[\mathbf{Y}_{\mathfrak{a}} \in \mathcal{C}_{\mathfrak{a}}]] \right] \geq \mathsf{OPT}(\mathfrak{I}). \tag{2.6}$$

Let S, σ_S, and $\mathbf{Y}'$ be defined as in Algorithm 4.7.16 on the input of $\mathbf{Y}$ and Π_k. Since the conditioning done on $\{\mathbf{Y}'\}$ is consistent with the local distribution, by law of total expectation and Eq. (3.14) one has

$$\mathop{\mathbb{E}}_{S} \mathop{\mathbb{E}}_{\sigma_S \sim \{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \mathop{\mathbb{E}}_{\{\mathbf{Y}'_\mathfrak{a}\}} \left[\mathbf{1}[\mathbf{Y}'_\mathfrak{a} \in \mathcal{C}_\mathfrak{a}]\right] = \mathsf{SDP}(\mathfrak{I}) \geq \mathsf{OPT}(\mathfrak{I}). \tag{2.7}$$

By Theorem 2.6.3 we know that

$$\mathop{\mathbb{E}}_{S} \mathop{\mathbb{E}}_{\sigma_S \sim \{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \left\| \{\mathbf{Y}'_\mathfrak{a}\} - \{\mathbf{Y}'_{a_1}\} \cdots \{\mathbf{Y}'_{a_k}\} \right\|_1 \leq \varepsilon \tag{2.8}$$

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

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{SAT}_\mathfrak{I}(\sigma)] = \mathop{\mathbb{E}}_{S} \mathop{\mathbb{E}}_{\sigma_S \sim \{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \mathop{\mathbb{E}}_{(\sigma_1,\ldots,\sigma_n) \sim \{\mathbf{Y}'_1\} \cdots \{\mathbf{Y}'_n\}} [\mathbf{1}[\sigma|_\mathfrak{a} \in \mathcal{C}_\mathfrak{a}]].$$

By using Eq. (3.16), we can obtain

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{SAT}_\mathfrak{I}(\sigma)] \geq \mathop{\mathbb{E}}_{S} \left[\mathop{\mathbb{E}}_{\sigma_S \sim \{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\{\mathbf{Y}_\mathfrak{a}\}} \mathbf{1}[\mathbf{Y}'_\mathfrak{a} \text{ satisfies the constraint on } \mathfrak{a}] \right] - \varepsilon.$$

Using Eq. (3.15), we can conclude

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{SAT}_\mathfrak{I}(\sigma)] \geq \mathsf{SDP}(\mathfrak{I}) - \varepsilon = \mathsf{OPT}(\mathfrak{I}) - \varepsilon.$$

■

Our proof of Theorem 2.6.3 will hinge on the fact that we can upper-bound the expected correlation of a face of large cardinality ℓ, in terms of expected correlation over faces of smaller cardinality and expected correlations along the edges of a swap graph.

The swap graph G_{ℓ_1,ℓ_2} here is defined as a weighted graph

$$G_{\ell_1,\ell_2} = \Big(X(\ell_1) \sqcup X(\ell_2), E(\ell_1,\ell_2), w_{\ell_1,\ell_2}\Big),$$

where

$$E(\ell_1,\ell_2) = \{\{\mathfrak{a},\mathfrak{b}\} : \mathfrak{a} \in X(\ell_1), \mathfrak{b} \in X(\ell_2), \text{ and } \mathfrak{a}\sqcup\mathfrak{b} \in X(\ell_1+\ell_2)\}.$$

We will assume $\ell_1 \geq \ell_2$, and if $\ell_1 = \ell_2$ we are going to identify the two copies of every vertex. We will endow $E(\ell_1,\ell_2)$ with the weight function,

$$w_{\ell_1,\ell_2}(\mathfrak{a},\mathfrak{b}) = \frac{\Pi_{\ell_1+\ell_2}(\mathfrak{a}\sqcup\mathfrak{b})}{\binom{\ell_1+\ell_2}{\ell_1}},$$

which can easily be verified to be a probability distribution on $E(\ell_1,\ell_2)$ Notice that in the case where $\ell_1 \neq \ell_2$ the random walk matrix of G_{ℓ_1,ℓ_2} is given by

$$\mathsf{A}_{\ell_1,\ell_2} = \begin{pmatrix} 0 & \mathsf{S}_{\ell_1,\ell_2} \\ \mathsf{S}^{\dagger}_{\ell_1,\ell_2} & 0 \end{pmatrix},$$

and if $\ell_1 = \ell_2$ we have $\mathsf{A}_{\ell_1,\ell_1} = \mathsf{S}_{\ell_1,\ell_1}$. The stationary distribution of $\mathsf{A}_{\ell_1,\ell_2}$ is Π_{ℓ_1,ℓ_2} defined by,

$$\Pi_{\ell_1,\ell_2}(\mathfrak{b}) = \mathbf{1}[\mathfrak{b} \in X(\ell_1)] \cdot \frac{1}{2} \cdot \Pi_{\ell_1}(\mathfrak{b}) + \mathbf{1}[\mathfrak{b} \in X(\ell_2)] \cdot \frac{1}{2} \cdot \Pi_{\ell_2}(\mathfrak{b}). \tag{2.9}$$

When we write an expectation of $f(\bullet,\bullet)$ over the edges in $E(\ell_1,\ell_2)$ with respect to w_{ℓ_1,ℓ_2}, it is important to note,

$$\mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}[f(\mathfrak{s},\mathfrak{t})] = \sum_{\{\mathfrak{s},\mathfrak{t}\}\in E(\ell_1,\ell_2)} \frac{1}{\binom{\ell_1+\ell_2}{\ell_1}} \cdot f(\mathfrak{s},\mathfrak{t}) \cdot \Pi_{\ell_1+\ell_2}(\mathfrak{s}\sqcup\mathfrak{t}) = \frac{1}{\binom{\ell}{\ell_1}} \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k}\left[\sum_{\{\mathfrak{s},\mathfrak{t}\}\sim\mathfrak{a}} f(\mathfrak{s},\mathfrak{t})\right], \tag{2.10}$$

where sum within the expectation in the RHS runs over the $\binom{\ell_1+\ell_2}{\ell_1}$ possible ways of

splitting $\mathfrak{a}$ into $\mathfrak{s} \sqcup \mathfrak{t}$ such that $\mathfrak{s} \in X(\ell_1)$ and $\mathfrak{t} \in X(\ell_2)$. When we are speaking about the spectral expansion of G_{ℓ_1,ℓ_2}, we will be speaking with regards to $\lambda_2(G_{\ell_1,\ell_2})$ and not with regards to $\sigma_2(G_{\ell_1,\ell_2})$.

Remark 2.6.5. *By simple linear algebra, we have*

$$\lambda_2(G_{\ell_1,\ell_2}) := \lambda_2(\mathsf{A}_{\ell_1,\ell_2}) \leq \sigma_2(\mathsf{S}_{\ell_1,\ell_2}),$$

where we employ the notation $\lambda_2(\mathsf{M})$ to denote the second largest eigenvalue (signed) of the matrix M.

With this, we will show

Lemma 2.6.6 (Glorified Triangle Inequality)**.** *For a simplicial complex $X(\leq k)$, $\ell_1 \geq \ell_2 \geq 0$, $\ell = \ell_1 + \ell_2$, $\ell \leq k$, and an ℓ-local ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$, one has*

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathfrak{a}\in\Pi_\ell}\left[\left\|\{\mathbf{Y}_\mathfrak{a}\} - \prod_{i=1}^{\ell}\{\mathbf{Y}_{a_i}\}\right\|_1\right] \leq \mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}[\|\{\mathbf{Y}_\mathfrak{s},\mathbf{Y}_\mathfrak{t}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1] \\ + \mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_{\ell_1}}\left[\left\|\{\mathbf{Y}_\mathfrak{s}\} - \prod_{i=1}^{\ell_1}\{\mathbf{Y}_{s_i}\}\right\|_1\right] + \mathop{\mathbb{E}}_{\mathfrak{t}\sim\Pi_{\ell_2}}\left[\left\|\{\mathbf{Y}_\mathfrak{t}\} - \prod_{i=1}^{\ell_2}\{\mathbf{Y}_{t_i}\}\right\|_1\right] \quad (2.11)\end{aligned}$$

One useful observation, is that by using Lemma 2.6.6 repeatedly, we can reduce the problem of bounding $\mathbb{E}_{\mathfrak{a}\in\Pi_\ell}\left\|\{\mathbf{Y}_\mathfrak{a}\} - \prod_{i=1}^{\ell}\{\mathbf{Y}_{a_i}\}\right\|_1$ to a problem of bounding

$$\mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}\|\{\mathbf{Y}_\mathfrak{s},\mathbf{Y}_\mathfrak{t}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1,$$

for $\ell_1 + \ell_2 \leq k$. Though it is not a direct implication, it is heavily suggested by Fact 4.7.14 and Theorem 2.6.2, that if G_{ℓ_1,ℓ_2} is a good spectral expander, after an application of Algorithm 4.7.16 with our chosen parameters, we should be able to bound these expressions. Using a key lemma used from [BRS11], we will prove that this is indeed the case. The

only thing we need to make sure after this point, is that the second eigenvalue $\lambda_2(G_{\ell_1,\ell_2})$ of the swap graphs G_{ℓ_1,ℓ_2} we will be using are small enough for our purposes. Indeed, our choice of γ in Theorem 2.6.3 and Corollary 2.6.4 is to make sure that the bound we get on $\lambda_2(G_{\ell_1,\ell_2})$ from Theorem 2.5.2 (together with Remark 2.6.5) is good enough for our purposes.

2.6.1 *Breaking Correlations for Expanding CSPs: Proof of Theorem 2.6.3*

Throughout this section, we will use the somewhat non-standard definition of variance introduced in [BRS11],

$$\mathrm{Var}\left[\mathbf{Y}_{\mathfrak{a}}\right] = \sum_{\sigma\in[q]^{\mathfrak{a}}} \mathrm{Var}\left[\mathbf{1}[\mathbf{Y}_{\mathfrak{a}} = \sigma]\right].$$

We will use the following central lemma from [BRS11] in our proof of Theorem 2.6.3:

Lemma 2.6.7 (Lemma 5.4 from [BRS11])**.** *Let $G = (V, E, \Pi_2)$ be a weighted graph, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a local PSD ensemble, where we have $\mathrm{Supp}(\mathbf{Y}_i) \leq q$ for every $i \in V$, and $q \geq 0$. Suppose $\varepsilon \geq 0$ is a lower bound on the expected statistical difference between independent and correlated sampling along the edges,i.e.,*

$$\varepsilon \leq \mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2}\left[\left\|\{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\}\right\|_1\right].$$

There exists absolute constants $c_0 \geq 0$ and $c_1 \geq 0$ that satisfy the following: If $\lambda_2(G) \leq c_0 \cdot \frac{\varepsilon^2}{q^2}$. Then, conditioning on a random vertex decreases the variances,

$$\mathop{\mathbb{E}}_{i\sim\Pi_1}\mathop{\mathbb{E}}_{j\sim\Pi_2}\mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}}\left[\mathrm{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j\right]\right] \leq \mathop{\mathbb{E}}_{i\sim\Pi_1}\left[\mathrm{Var}\left[\mathbf{Y}_i\right]\right] - c_1\cdot\frac{\varepsilon^2}{q^2}.$$

For our applications, we will be instantiating Lemma 2.6.7 with G_{ℓ_1,ℓ_2} as G; and with the local PSD ensemble $\{\mathbf{Y}_{\mathfrak{a}}\}_{\mathfrak{a}\in X}$ that is obtained from $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ (q.v. Fact 4.7.14). For convenience, we will write the concrete instance of the Lemma that we will use,

Corollary 2.6.8. *Let $\ell_1 \geq \ell_2 \geq 0$ satisfying $\ell_1 + \ell_2 \leq k$ be given parameters, and let G_{ℓ_1,ℓ_2} be the swap graph defined for a γ-HDX $X(\leq k)$. Let $\{\mathbf{Y}_\mathfrak{a}\}_{\mathfrak{a}\in X}$ be a local PSD ensemble; satisfying $\mathrm{Supp}(Y_\mathfrak{a}) \leq q^k$ for every $\mathfrak{a} \in X(\ell_1) \cup X(\ell_2)$ for some $q \geq 0$. Suppose $\varepsilon \geq 0$ satisfies,*

$$\frac{\varepsilon}{4k} \leq \mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\in w_{\ell_1,\ell_2}} \left[\left\|\{\mathbf{Y}_{\mathfrak{s}\sqcup\mathfrak{t}}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\right\|_1\right].$$

There exists absolute constants $c_0 \geq 0$ and $c_2 \geq 0$ that satisfy the following: If $\lambda_2(G) \leq c_0 \cdot (\varepsilon/(4k \cdot q^k))^2$. Then, conditioning on a random face $\mathfrak{a} \sim \Pi_{\ell_1,\ell_2}$ decreases the variances, i.e.

$$\begin{aligned} 2 \cdot \mathop{\mathbb{E}}_{\mathfrak{a},\mathfrak{b}\sim\Pi^2_{\ell_1,\ell_2}} \left[\mathop{\mathbb{E}}_{\{\mathbf{Y}_\mathfrak{a}\}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{b} \mid \mathbf{Y}_\mathfrak{a}\right]\right]\right] &= \mathop{\mathbb{E}}_{\mathfrak{a}\in\Pi_{\ell_1,\ell_2}} \left[\mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_{\ell_1}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{s} \mid \mathbf{Y}_\mathfrak{a}\right]\right] + \mathop{\mathbb{E}}_{\mathfrak{t}\sim\Pi_{\ell_2}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{t} \mid \mathbf{Y}_\mathfrak{a}\right]\right]\right], \\ &\leq \mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_{\ell_1}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{s}\right]\right] + \mathop{\mathbb{E}}_{\mathfrak{t}\sim\Pi_{\ell_2}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{t}\right]\right] - c_2 \cdot \frac{\varepsilon^2}{16 \cdot k^2 \cdot q^{2k}}. \end{aligned}$$

Here, it can be verified that the expansion criterion presupposed by Lemma 2.6.7 is satisfied by Corollary 2.6.8 by Theorem 2.5.2. The constant c_2 satisfies $c_2 = 2 \cdot c_1$.

Proof of Theorem 2.6.3. We will follow the same proof strategy in [BRS11], and extend their arguments for k-CSPs.

Write Π_k^m for the distribution of the random set that is obtained in steps (2)-(3) of Algorithm 4.7.16 with $\Pi = \Pi_k$, i.e. $S \sim \Pi_k^m$ is sampled by

1. independently sampling m k-faces $\mathfrak{s}_j \sim \Pi_k$ for $j = 1, \ldots, m$.
2. outputting $S = \bigcup_{j=1}^m \mathfrak{s}_j$.

First, for $m \in [L/k]$ we will define

$$\varepsilon_m = \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \left[\left\|\{\mathbf{Y}_\mathfrak{a} \mid \mathbf{Y}_S\} - \prod_{j=1}^k \{\mathbf{Y}_{a_j} \mid \mathbf{Y}_S\}\right\|_1\right],$$

which will measure the average correlation along $X(k)$ after conditioning on m k-faces. Notice that our goal is ensuring,

$$\mathop{\mathbb{E}}_{m\sim[L/k]} \varepsilon_m \leq \varepsilon$$

where m is sampled uniformly at random.

To help us with this goal, we will define a potential function

$$\Phi_m = \mathop{\mathbb{E}}_{i\sim[k]} \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_i} \mathrm{Var}\left[\mathbf{Y}_{\mathfrak{a}} \mid \mathbf{Y}_S\right]. \tag{2.12}$$

where i is sampled uniformly at random. Observe that Φ_m always satisfies $0 \leq \Phi_m \leq 1$. Using this, we will try to bound the fraction of indices $m \in [L/k]$ such that ε_m is large, i.e., say $\varepsilon_m \geq \varepsilon/2$. To this end assume $\varepsilon_m \geq \varepsilon/2$, i.e. we have

$$\mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}} \mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \left[\left\| \{\mathbf{Y}_{\mathfrak{a}} \mid \mathbf{Y}_S\} - \prod_{i=1}^{k} \{\mathbf{Y}_{a_1} | \mathbf{Y}_S\} \right\|_1\right] \geq \frac{\varepsilon}{2}. \tag{2.13}$$

We will use Lemma 2.6.6 in the following way: Let $\mathcal{T}$ be any binary tree with k leaves. We will label each of the vertices $v \in \mathcal{T}$ with the number of leaves of the subtree rooted at v. Notice that this ensures that

1. the root vertex of $\mathcal{T}$ has the label k,
2. for any vertex $v \in \mathcal{T}$ with label ℓ, the label ℓ_1 of the left child of v and the label ℓ_2 of the right child of v add up to k, i.e. $\ell_1 + \ell_2 = k$,
3. every vertex $v \in \mathcal{T}$ with the label 1 is a leaf.

We write $J(\mathcal{T})$ for the set of labels ℓ of the internal nodes of $\mathcal{T}$, note $|J(\mathcal{T})| \leq k$. We will use the notation ℓ_1 (resp. ℓ_2) to refer to the label of the left (resp. right) of a vertex $v \in \mathcal{T}$ with the label ℓ.

By applying Lemma 2.6.6, we obtain that for any local PSD ensemble $\mathbf{Z}$ one has

$$\mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k}\left[\left\|\{\mathbf{Z}_{\mathfrak{a}}\}-\prod_{i=1}^{k}\{\mathbf{Z}_{\mathfrak{a}_i}\}\right\|_1\right]\leq\sum_{\ell\in J(\mathcal{T})}\mathop{\mathbb{E}}_{\{t_1,t_2\}\in w_{\ell_1,\ell_2}}\left[\left\|\{\mathbf{Z}_{t_1\sqcup t_2}\}-\{\mathbf{Z}_{t_1}\}\{\mathbf{Z}_{t_2}\}\right\|_1\right].$$

Now, by plugging this in Eq. (4.8), with $\mathbf{Z}_{\mathfrak{a}}=\{\mathbf{Y}_{\mathfrak{a}}\mid\mathbf{Y}_S\}$, we obtain

$$\mathop{\mathbb{E}}_{S\sim\Pi_k^m}\mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}}\left[\sum_{\ell\in J(\mathcal{T})}\mathop{\mathbb{E}}_{\{t_1,t_2\}\sim w_{\ell_1,\ell_2}}\left\|\{\mathbf{Y}_{t_1\sqcup t_2}\mid\mathbf{Y}_S\}-\{\mathbf{Y}_{t_1}\mid\mathbf{Y}_S\}\{\mathbf{Y}_{t_2}\mid\mathbf{Y}_S\}\right\|_1\right]\geq\frac{\varepsilon}{2}. \quad (2.14)$$

In particular, in the sum over $J(\mathcal{T})$ there should be some large term corresponding to some $\ell\in J(\mathcal{T})$. i.e. we have,

$$\mathop{\mathbb{E}}_{S\sim\Pi_k^m}\mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}}\left[\mathop{\mathbb{E}}_{\{t_1,t_2\}\in w_{\ell_1,\ell_2}}\left\|\{\mathbf{Y}_{t_1\sqcup t_2}\mid\mathbf{Y}_S\}-\{\mathbf{Y}_{t_1}\mid\mathbf{Y}_S\}\{\mathbf{Y}_{t_2}\mid\mathbf{Y}_S\}\right\|_1\right]\geq\frac{\varepsilon}{2\cdot|J(\mathcal{T})|}\geq\frac{\varepsilon}{2k}.$$

Now, we have

$$\mathop{\mathbb{P}}_{\substack{S\sim\Pi_k^m\\{\mathbf{Y}_S\}}}\left[\mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\in w_{\ell_1,\ell_2}}\left\|\{\mathbf{Y}_{t_1\sqcup t_2}\mid\mathbf{Y}_S\}-\{\mathbf{Y}_{t_1}\mid\mathbf{Y}_S\}\{\mathbf{Y}_{t_2}\mid\mathbf{Y}_S\}\right\|_1\geq\frac{\varepsilon}{4k}\right]\geq\frac{\varepsilon}{4k}.$$

This together with Corollary 2.6.8 implies,

$$\mathop{\mathbb{P}}_{\substack{S\sim\Pi_k^m\\{\mathbf{Y}_S\}}}\left[\mathop{\mathbb{E}}_{\mathfrak{a}\in\Pi_{\ell_1,\ell_2}}\left[\begin{array}{c}\mathbb{E}_{t_1\sim\Pi_{\ell_1}}[\mathrm{Var}[\mathbf{Y}_{t_1}\mid\mathbf{Y}_S]-\mathrm{Var}[\mathbf{Y}_{t_1}\mid\mathbf{Y}_S,\mathbf{Y}_{\mathfrak{a}}]]\\+\quad\mathbb{E}_{t_2\in\Pi_{\ell_2}}[\mathrm{Var}[\mathbf{Y}_{t_2}|\mid\mathbf{Y}_S]-\mathrm{Var}[\mathbf{Y}_{t_2}\mid\mathbf{Y}_S,\mathbf{Y}_{\mathfrak{a}}]]\end{array}\right]\geq c_2\cdot\frac{\varepsilon^2}{16\cdot k^2\cdot q^{2k}}\right]$$
$$\geq\frac{\varepsilon}{4k}, \quad (2.15)$$

provided that $\lambda_2(G_{\ell_1,\ell_2})\leq c_0(\varepsilon/(4k\cdot q^k))^2$.

Now, observe that a sample $\mathfrak{a}\sim\Pi_{\ell_1,\ell_2}$ can be obtained from a sample $\mathfrak{s}_{m+1}\sim\Pi_k$ in the following way,

1. with probability $\frac{1}{2}$ each, pick $j = 1$ or $j = 2$.

2. delete all but ℓ_j elements from $\mathfrak{s}_{m+1}$.

It is important to note that for the sample $\mathfrak{a} \sim \Pi_{\ell_1,\ell_2}$ obtained this way, we have $\mathfrak{s}_{m+1} \supseteq \mathfrak{a}$. An application of Jensen's inequality shows that the variance is non-increasing under conditioning, i.e. for random variables $\mathbf{Z}$ and $\mathbf{W}$ we have,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{Z}}\left[\mathrm{Var}\left[\mathbf{W} \mid \mathbf{Z}\right]\right] &= \mathop{\mathbb{E}}_{\mathbf{Z}}\left[\mathop{\mathbb{E}}_{\mathbf{W}}\left[\mathbf{W}^2 \mid \mathbf{Z}\right]\right] - \mathop{\mathbb{E}}_{\mathbf{Z}}\left[\left(\mathop{\mathbb{E}}_{\mathbf{W}}\left[\mathbf{W} \mid \mathbf{Z}\right]^2\right)\right], \\
&\leq \mathbb{E}\left[\mathbf{W}^2\right] - \left(\mathop{\mathbb{E}}_{\mathbf{Z}}\left[\mathbb{E}\left[\mathbf{W} \mid \mathbf{Z}\right]\right]\right)^2, \\
&= \mathrm{Var}\left[\mathbf{W}\right].
\end{aligned}$$

This means conditioning on $\mathfrak{s}_{m+1}$, the drop in variance can only be more, i.e., Eq. (2.15) implies

$$\mathop{\mathbb{P}}_{\substack{S\sim\Pi_k^m \\ \{\mathbf{Y}_S\}}}\left[\mathop{\mathbb{E}}_{\mathfrak{s}_{m+1}\in\Pi_k}\left[\begin{array}{c}\mathbb{E}_{t_1\sim\Pi_{\ell_1}}[\mathrm{Var}[\mathbf{Y}_{t_1} \mid \mathbf{Y}_S] - \mathrm{Var}[\mathbf{Y}_{t_1} \mid \mathbf{Y}_S, \mathbf{Y}_{\mathfrak{s}_{m+1}}]] \\ + \quad \mathbb{E}_{t_2\in\Pi_{\ell_2}}[\mathrm{Var}[\mathbf{Y}_{t_2}| \mid \mathbf{Y}_S] - \mathrm{Var}[\mathbf{Y}_{t_2} \mid \mathbf{Y}_S, \mathbf{Y}_{\mathfrak{s}_{m+1}}]]\end{array}\right] \geq c_2 \cdot \frac{\varepsilon^2}{16 \cdot k^2 \cdot q^{2k}}\right] \geq \frac{\varepsilon}{4k}.$$

By relabeling ℓ_1 as ℓ_2 if needed, we can obtain the following inequality from the above

$$\mathop{\mathbb{P}}_{\substack{S\sim\Pi_k^m \\ \{\mathbf{Y}_S\}}}\left[\mathop{\mathbb{E}}_{\mathfrak{s}_{m+1}\in\Pi_k}\left[\mathop{\mathbb{E}}_{t_1\sim\Pi_{\ell_1}}\left[\mathrm{Var}\left[\mathbf{Y}_{t_1} \mid \mathbf{Y}_S\right] - \mathrm{Var}\left[\mathbf{Y}_{t_1} \mid \mathbf{Y}_S, \mathbf{Y}_{\mathfrak{s}_{m+1}}\right]\right]\right] \geq c_2 \cdot \frac{\varepsilon^2}{32 \cdot k^2 \cdot q^{2k}}\right] \geq \frac{\varepsilon}{4k}. \tag{2.16}$$

This implies

$$\Phi_m - \Phi_{m+1} \geq \frac{1}{k} \cdot \frac{\varepsilon}{4k} \cdot \left(c_2 \cdot \frac{\varepsilon^2}{32 \cdot k^2 \cdot q^{2k}}\right) = c_2 \cdot \frac{\varepsilon^3}{128 \cdot k^4 \cdot q^{2k}},$$

where the $\frac{1}{k}$ term in the RHS corresponds to $\ell_1 \in [k]$ being chosen in Eq. (4.7), the $\frac{\varepsilon}{4k}$ term in the RHS corresponds to the probability of the variances in $X(\ell_1)$ drop by $\left(c_2 \cdot \frac{\varepsilon^2}{32 \cdot k^2 \cdot q^k}\right)$.

Since, the variance is non-increasing under conditioning

$$1 \geq \Phi_1 \geq \cdots \geq \Phi_m \geq 0.$$

this means there can be at most $128k^4 \cdot q^{2k}/(c_2 \cdot \varepsilon^3)$ indices $m \in [L/k]$ such that $\varepsilon_m \geq \varepsilon/2$. In particular, since the total number of indices is (L/k) we have,

$$\mathop{\mathbb{E}}_{m \sim [L/k]} \varepsilon_m \leq \frac{\varepsilon}{2} + \frac{k}{L} \cdot \frac{128 \cdot k^4 \cdot q^{2k}}{c_2 \cdot \varepsilon^3}.$$

This means that there exists an absolute constant $c' \geq 0$ such that

$$L \geq c' \cdot \frac{k^5 \cdot q^{2k}}{\varepsilon^4} \quad \text{ensures} \quad \mathop{\mathbb{E}}_{m \in [L/k]} [\varepsilon_m] \leq \varepsilon.$$

To finish our proof, we note that to justify our applications of Corollary 2.6.8 it suffices to ensure

$$\lambda_2(G_{\ell_1,\ell_2}) \leq c_0 \cdot \left(\frac{\varepsilon}{4k \cdot q^k}\right)^2 = c_0 \cdot \frac{\varepsilon^2}{16 \cdot k^2 \cdot q^{2k}}$$

for all ℓ_1, ℓ_2 occurring in $\mathcal{T}$ as a label. It can be verified that our choice of γ together with Theorem 2.5.2 (and Remark 2.6.5) satisfies this, where the constant $C' \geq 0$ will account for c_0, c', and the constants hidden within the O-notation in Theorem 2.5.2. ∎

2.6.2 *The Glorified Triangle Inequality: Proof of Lemma 2.6.6*

In this Section, we will prove Lemma 2.6.6.

Proposition 2.6.9. *Let* $\mathbf{Y}, \mathbf{Z}, \mathbf{U}, \mathbf{W}$ *be random variables where* $\mathbf{Y}$ *and* $\mathbf{Z}$*; and* $\mathbf{U}$ *and* $\mathbf{W}$ *are on the same support. Then,*

$$\|\{\mathbf{Y}\}\{\mathbf{U}\} - \{\mathbf{Z}\}\{\mathbf{W}\}\|_1 \leq \|\{\mathbf{Y}\} - \{\mathbf{Z}\}\|_1 + \|\{\mathbf{U}\} - \{\mathbf{W}\}\|_1.$$

Proof. Tensoring with the same probability distribution does not change the total variation distance, i.e.

$$\|\{\mathbf{Y}\} - \{\mathbf{Z}\}\|_1 = \|\{\mathbf{Y}\}\{\mathbf{U}\} - \{\mathbf{Z}\}\{\mathbf{U}\}\|_1 \quad \text{and} \quad \|\{\mathbf{U}\} - \{\mathbf{W}\}\|_1 = \|\{\mathbf{Z}\}\{\mathbf{U}\} - \{\mathbf{Z}\}\{\mathbf{W}\}\|_1 .$$

Now, a simple application of the triangle inequality proves the Proposition. ∎

A straightforward implication of Proposition 2.6.9 is the following, which will allow us to bound the correlation along a face $\mathfrak{a} \in X(k)$, using the correlation along sub-faces $\mathfrak{s}, \mathfrak{t} \subseteq \mathfrak{a}$.

Corollary 2.6.10. *Let $\mathfrak{a} \in X(\ell)$ and $\mathfrak{s} \in X(\ell_1), \mathfrak{t} \in X(\ell_2)$ be given such that $\mathfrak{a} = \mathfrak{s} \sqcup \mathfrak{t}$. Then for any k-local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ we have*

$$\begin{aligned}\left\|\{\mathbf{Y}_\mathfrak{a}\} - \{\mathbf{Y}_{a_1}\} \cdots \{\mathbf{Y}_{a_\ell}\}\right\|_1 \leq \|\{\mathbf{Y}_\mathfrak{a}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1 \\ + \left\|\{\mathbf{Y}_\mathfrak{s}\} - \{\mathbf{Y}_{s_1}\} \cdots \{\mathbf{Y}_{s_{\ell_1}}\}\right\|_1 + \left\|\{\mathbf{Y}_\mathfrak{t}\} - \{\mathbf{Y}_{t_1}\} \cdots \{\mathbf{Y}_{t_{\ell_2}}\}\right\|_1\end{aligned}$$

With this, we can go ahead and prove Lemma 2.6.6

Proof of Lemma 2.6.6. Let $\mathfrak{a} \in X(\ell)$ be a fixed face. By Corollary 2.6.10 and averaging over all the $\binom{\ell=\ell_1+\ell_2}{\ell_1}$ ways of splitting $\mathfrak{a}$ into $\{\mathfrak{s}, \mathfrak{t}\}$ such that $\mathfrak{s} \in X(\ell_1)$ and $\mathfrak{t} \in X(\ell_2)$ we have

$$\begin{aligned}&\left\|\{\mathbf{Y}_\mathfrak{a}\} - \prod_{i=1}^{\ell=\ell_1+\ell_2} \{\mathbf{Y}_{a_i}\}\right\|_1 \\ &\leq \frac{1}{\binom{\ell_1+\ell_2}{\ell_1}} \sum_{\{\mathfrak{s},\mathfrak{t}\}} \left(\|\{\mathbf{Y}_\mathfrak{a}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1 + \left\|\{\mathbf{Y}_\mathfrak{s}\} - \prod_{i=1}^{\ell_1} \{Y_{s_i}\}\right\|_1 + \left\|\{\mathbf{Y}_\mathfrak{t}\} - \prod_{i=1}^{\ell_2} \{\mathbf{Y}_{t_i}\}\right\| \right).\end{aligned}$$

Now, by taking an average over all the edges $\mathfrak{a} \in X(\ell)$ (with respect to the measure Π_ℓ)

we obtain,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_\ell}\left[\left\|\{\mathbf{Y}_\mathfrak{a}\}-\prod_{i=1}^{\ell}\{\mathbf{Y}_{a_i}\}\right\|_1\right]\\
&\qquad\le \frac{1}{\binom{\ell}{\ell_1}}\cdot\mathop{\mathbb{E}}_{\mathfrak{a}\in\Pi_\ell}\left[\sum_{\{\mathfrak{s},\mathfrak{t}\}}\left(\|\{\mathbf{Y}_\mathfrak{a}\}-\{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1+\left\|\{\mathbf{Y}_\mathfrak{s}\}-\prod_{i=1}^{k_1}\{\mathbf{Y}_{s_i}\}\right\|_1+\left\|\{\mathbf{Y}_\mathfrak{t}\}-\prod_{i=1}^{\ell_2}\{\mathbf{Y}_{t_i}\}\right\|_1\right)\right]
\end{aligned}$$

where the indices $\{\mathfrak{s},\mathfrak{t}\}$ run over the all the ways of splitting $\mathfrak{a}$ into $\mathfrak{s}$ and $\mathfrak{t}$ as before. We can now see that the RHS can be thought as an average over the (weighted) edges in $E(\ell_1,\ell_2)$ (q.v. Eq. (2.10)), i.e.,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_\ell}\left[\left\|\{\mathbf{Y}_\mathfrak{a}\}-\prod_{i=1}^{\ell}\{\mathbf{Y}_{a_i}\}\right\|_1\right]\\
&\qquad\le \mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}\left[\|\{\mathbf{Y}_\mathfrak{a}\}-\{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1+\left\|\{\mathbf{Y}_\mathfrak{s}\}-\prod_{i=1}^{\ell_1}\{\mathbf{Y}_{s_i}\}\right\|_1+\left\|\{\mathbf{Y}_\mathfrak{t}\}-\prod_{i=1}^{\ell_2}\{Y_{t_i}\}\right\|_1\right]
\end{aligned}$$

Now, note that since Π_{ℓ_1,ℓ_2} (q.v. Eq. (2.9)) is the stationary distribution of the walk defined on G_{ℓ_1,ℓ_2}, i.e.,

$$2\Pi_{\ell_1,\ell_2}(\mathfrak{a})=\sum_{\mathfrak{b}:\{\mathfrak{a},\mathfrak{b}\}\in E(\ell_1,\ell_2)} w_{\ell_1,\ell_2}(\mathfrak{a},\mathfrak{b}),$$

the lemma follows. This is because, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathfrak{a}\in X(\ell)}\left[\left\|\{\mathbf{Y}_\mathfrak{a}\}-\prod_{i=1}^{\ell}\{\mathbf{Y}_{a_i}\}\right\|_1\right]\\
&\quad\le \mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}[\|\{\mathbf{Y}_\mathfrak{a}\}-\{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1]+\mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim w_{\ell_1,\ell_2}}\left[\left\|\{\mathbf{Y}_\mathfrak{s}\}-\prod_{i=1}^{\ell_1}\{\mathbf{Y}_{s_i}\}\right\|_1+\left\|\{\mathbf{Y}_\mathfrak{t}\}-\prod_{i=1}^{\ell_2}\{Y_{t_i}\}\right\|_1\right]\\
=\mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\sim E(\ell_1,\ell_2)}&[\|\{\mathbf{Y}_\mathfrak{a}\}-\{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1]+\mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_{\ell_1}}\left[\left\|\{\mathbf{Y}_\mathfrak{s}\}-\prod_{i=1}^{\ell_1}\{\mathbf{Y}_{s_i}\}\right\|_1\right]+\mathop{\mathbb{E}}_{\mathfrak{t}\sim\Pi_{\ell_2}}\left[\left\|\{\mathbf{Y}_\mathfrak{t}\}-\prod_{i=1}^{\ell_2}\{\mathbf{Y}_{t_i}\}\right\|_1\right]
\end{aligned}$$

■

2.7 High-Dimensional Threshold Rank

In [BRS11], Theorem 2.6.2 was proven for a more general class of graphs than expander graphs – namely, the class of low threshold rank graphs.

Definition 2.7.1 (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*

$$\mathrm{rank}_{\geq\tau}(\mathsf{A}) := |\{i|\lambda_i \geq \tau\}|\,.$$

There [BRS11], the authors asked for the correct notion of threshold rank for k-CSPs. In this section, we give a candidate definition of low threshold rank motivated by our techniques.

To break k-wise correlations it is sufficient to assume that the involved swap graphs in the foregoing discussion are low threshold rank since this is enough to apply a version of Lemma 2.6.7, already described in the work of [BRS11].

Moreover, we have some flexibility as to which swap graphs to consider as long as they satisfy some splitting conditions. To define a swap graph it is enough to have a distributions on the hyperedges of a (constraint) hypergraph. Hence, the notion of swap graph is independent of high-dimensional expansion. HDXs are just an interesting family of objects for which the swap graphs are good expanders.

To capture the many ways of splitting the statistical distance over hyperedges into the statistical distance over the edges of swap graphs, we first define the following notion. We say that a binary tree $\mathcal{T}$ is a k-splitting tree if it has exactly k leaves. Thus, labeling every vertex with the number of leaves on the subtree rooted at that vertex ensures,

- 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 satisfy the property that its label is the sum of the labels of its two children.

Note that, we will think of each non-leaf node with left and right children labeled as a and b as representing the swap graph from $X(a)$ to $X(b)$ for some simplicial complex $X(\leq k)$. Let $\mathrm{Swap}(\mathcal{T}, X)$ be the set of all such swap graphs over X finding representation in the splitting tree $\mathcal{T}$. Indeed the tree $\mathcal{T}$ used in the proof of Theorem 2.6.3 is just one special instance of a k-splitting tree.

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 of $\mathcal{A}$ as

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

where $\mathrm{rank}_{\geq\tau}(\mathsf{A})$ is denotes usual threshold rank of A as in Definition 3.9.14.

Now, we are ready to define the notion of a k-CSP instance being $(\mathcal{T}, \tau, r)$-splittable as follows.

Definition 2.7.2 ($(\mathcal{T}, \tau, r)$-splittability)**.** *A k-CSP instance $\mathfrak{I}$ with the constraint complex $X(\leq k)$ is said to be $(\mathcal{T}, \tau, r)$-splittable if $\mathcal{T}$ is a k-splitting tree and*

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

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

Now, using this definition we can show that whenever $\mathrm{rank}_\tau(\mathfrak{I})$ is bounded for the appropriate choice of τ, after conditioning on a random partial assignment as in Algorithm 4.7.16 we will have small correlation over the faces $\mathfrak{a} \in X(k)$, i.e.,

Theorem 2.7.3. *Suppose a simplicial complex $X(\leq k)$ with $X(1) = [n]$ and an L-local PSD ensemble $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ are given. There exists some universal constants $c_4 \geq 0$ and $C'' \geq 0$ satisfying the following: If $L \geq C'' \cdot (q^{4k} \cdot k^7 \cdot r/\varepsilon^5)$, $\mathrm{Supp}(\mathbf{Y}_j) \leq q$ for all $j \in [n]$, and $\mathfrak{I}$ is $(c_4 \cdot (\varepsilon/(4k \cdot q^k))^2, r)$-splittable. Then, we have*

$$\mathop{\mathbb{E}}_{\mathfrak{a} \in X(k)} \left[\left\| \{\mathbf{Y}'_{\mathfrak{a}}\} - \left\{\mathbf{Y}'_{a_1}\right\} \cdots \left\{\mathbf{Y}'_{a_k}\right\} \right\|_1 \right] \leq \varepsilon, \tag{2.17}$$

where $\mathbf{Y}'$ is as defined in Algorithm 4.7.16 on the input of $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ and Π_k.

It is important to note that the specific knowledge of the k-splitting tree $\mathcal{T}$ that makes $\mathfrak{I}$ $(\mathcal{T}, \tau, r)$-splittable is only needed for the proof of Theorem 3.9.19. The conclusion of Theorem 3.9.19 can be used without the knowledge of the specific k-splitting tree $\mathcal{T}$. The attentive reader might have noticed is that in the proof of Theorem 2.6.3, the choice of $\mathcal{T}$ is not important, as all the splitting tree are guaranteed to have be expanders provided that X is a γ-HDX. The proof of Theorem 3.9.19, in this light can be thought of an extension of the proof of Theorem 2.6.3 to the case where not necessarily every tree is good, and where we can bound the threshold rank instead of the spectral expansion.

This, will readily imply an algorithm

Corollary 2.7.4. *Suppose $\mathfrak{I}$ is a q-ary k-CSP instance whose constraint complex is $X(\leq k)$. There exists an absolute constant $C'' \geq 0$ and $c_4 \geq 0$ that satisfies the following: If $\mathfrak{I}$ is $(c_4 \cdot (\varepsilon/(4k \cdot q^k))^2, r)$-splittable, then there is an algorithm that runs in time $n^{O\left(\frac{q^{4k} \cdot k^7 \cdot r}{\varepsilon^5}\right)}$ and that is based on $(\frac{C'' \cdot k^5 \cdot q^k \cdot r}{\varepsilon^4})$-levels of SoS-hierarchy and Algorithm 4.7.16 that outputs a random assignment $\sigma : [n] \to [q]$ that in expectation ensures $\mathsf{SAT}_{\mathfrak{I}}(\sigma) = \mathsf{OPT}(\mathfrak{I}) - \varepsilon$.*

Since the proof of Corollary 3.9.21 given Theorem 3.9.19, will be almost identical to the proof of Corollary 2.6.4, given Theorem 2.6.3, we will omit the proof of this.

2.7.1 *Breaking Correlations for Splittable CSPs: Proof of Theorem 3.9.19*

We will need the more general version of Lemma 2.6.7, already proven in [BRS11].

Lemma 2.7.5 (Lemma 5.4 from [BRS11]). [7] *Let $G = (V, E, \Pi_2)$ be a weighted graph, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a local PSD ensemble, where we have $\mathrm{Supp}(\mathbf{Y}_i) \leq q$ for every $i \in V$, and $q \geq 0$. If $\varepsilon \geq 0$ is a lower bound on the expected statistical difference between independent and correlated sampling along the edges,i.e.,*

$$\varepsilon \leq \mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2} \left[\left\| \{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\} \right\|_1 \right].$$

There exists absolute constants $c_3 \geq 0$ and $c_4 \geq 0$ that satisfy the following: Then, conditioning on a random vertex decreases the variances,

$$\mathop{\mathbb{E}}_{i\sim\Pi_1} \mathop{\mathbb{E}}_{j\sim\Pi_1} \mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}} \left[\mathrm{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j \right] \right] \leq \mathop{\mathbb{E}}_{i\sim\Pi_1} \left[\mathrm{Var}\left[\mathbf{Y}_i \right] \right] - c_3 \cdot \frac{\varepsilon^4}{q^4 \cdot \mathrm{rank}_{\geq c_4\varepsilon^2/q^2}(G)}.$$

Since we will use this lemma, only with the swap graphs G_{ℓ_1,ℓ_2} and (L/k)-local PSD ensemble $\{\mathbf{Y}_\mathfrak{a}\}_{\mathfrak{a}\in X}$ obtained from the L-local PSD ensemble $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$, for convenience we will write the corollary we will use more explicitly

Corollary 2.7.6. *Let $\ell_1 \geq \ell_2 \geq 0$ satisfying $\ell_1 + \ell_2 \leq k$ be given parameters, and let G_{ℓ_1,ℓ_2} be the swap graph defined for a γ-HDX $X(\leq k)$. Let $\{\mathbf{Y}_\mathfrak{a}\}_{\mathfrak{a}\in X}$ be a local PSD ensemble; and suppose we have $\mathrm{Supp}(Y_\mathfrak{a}) \leq q^k$ for every $\mathfrak{a} \in X(\ell_1) \cup X(\ell_2)$ for some $q \geq 0$. Suppose $\varepsilon > 0$ satisfies,*

$$\frac{\varepsilon}{4k} \leq \mathop{\mathbb{E}}_{\{\mathfrak{s},\mathfrak{t}\}\in E(\ell_1,\ell_2)} \left[\|\{\mathbf{Y}_{\mathfrak{s}\cup\mathfrak{t}}\} - \{\mathbf{Y}_\mathfrak{s}\}\{\mathbf{Y}_\mathfrak{t}\}\|_1 \right].$$

7. We give a derivation of this lemma in Appendix A.1.

There exists absolute constants $c_3 \geq 0$ and $c_5 \geq 0$ that satisfy the following:
If $\mathrm{rank}_{\geq c_4 \cdot (\varepsilon/(4k \cdot q^k))^2}(G_{\ell_1,\ell_2}) \leq r$*, then conditioning on a random face* $\mathfrak{a} \sim \Pi_{\ell_1,\ell_2}$ *decreases the variances, i.e.*

$$\begin{aligned} 2 \cdot \mathop{\mathbb{E}}_{\mathfrak{a},\mathfrak{b} \sim \Pi_{\ell_1,\ell_2}} \left[\mathop{\mathbb{E}}_{\{\mathbf{Y}_\mathfrak{a}\}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{b} \mid \mathbf{Y}_\mathfrak{a} \right] \right] \right] &= \mathop{\mathbb{E}}_{\mathfrak{a} \in \Pi_{\ell_1,\ell_2}} \left[\mathop{\mathbb{E}}_{\mathfrak{s} \sim \Pi_{\ell_1}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{s} \mid \mathbf{Y}_\mathfrak{a} \right] \right] + \mathop{\mathbb{E}}_{\mathfrak{t} \sim \Pi_{\ell_2}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{t} \mid \mathbf{Y}_\mathfrak{a} \right] \right] \right], \\ &\leq \mathop{\mathbb{E}}_{\mathfrak{s} \sim \Pi_{\ell_1}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{s} \right] \right] + \mathop{\mathbb{E}}_{\mathfrak{t} \sim \Pi_{\ell_2}} \left[\mathrm{Var}\left[\mathbf{Y}_\mathfrak{t} \right] \right] - c_5 \cdot \frac{\varepsilon^4}{256 \cdot k^4 \cdot q^{4k} \cdot r}. \end{aligned}$$

Here the constant c_5 satisfies $c_5 = 2 \cdot c_3$.

Proof. As the proof will mostly follow Theorem 2.6.3, we will only highlight the relevant differences and carry out the relevant computations.

Let $\tau = c_4 \cdot (\varepsilon/(4k \cdot q^k))^2$, and let $\mathcal{T}$ be the k-splitting tree certifying that $\mathfrak{I}$ is $(\mathcal{T}, \tau, r)$ splittable, i.e., the tree $\mathcal{T}$ satisfies $\mathrm{rank}_\tau(\mathrm{Swap}(\mathcal{T}, X)) \leq r$. This means that all the swap graphs G_{ℓ_1,ℓ_2} finding representation in $\mathcal{T}$ satisfy $\mathrm{rank}_\tau(G_{\ell_1,\ell_2}) \leq r$.

Similarly, as in the proof of we will try to argue that the fraction of indices $m \in [L/k]$ such that ε_m that is large, say $\varepsilon_m \geq \varepsilon/2$, is small by arguing about the potential Φ_m with both quantities ε_m and Φ_m as defined as in the Proof of Theorem 2.6.3. We assume similarly, that $\varepsilon_m \geq \varepsilon/2$ for some $m \in [L/k]$.

Analogously to Section 2.7.1 in the proof of Theorem 2.6.3, from Corollary 2.7.6 we obtain

$$\mathop{\mathbb{E}}_{S \sim \Pi_k^m} \mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}} \left[\sum_{\ell \in J(\mathcal{T})} \mathop{\mathbb{E}}_{\{\mathfrak{t}_1,\mathfrak{t}_2\} \in E(\ell_1,\ell_2)} \left[\left\| \{\mathbf{Y}_{\mathfrak{t}_1 \sqcup \mathfrak{t}_2} \mid \mathbf{Y}_S\} - \{\mathbf{Y}_{\mathfrak{t}_1} \mid \mathbf{Y}_S\} \{\mathbf{Y}_{\mathfrak{t}_2} \mid \mathbf{Y}_S\} \right\|_1 \right] \right] \geq \frac{\varepsilon}{2}.$$

Notice that the assumption that Section 2.7.1 makes on the threshold rank is satisfied by the assumption $\mathrm{rank}_\tau(\mathfrak{I}) \leq r$ and where the set $J(\mathcal{T})$ contains all labels ℓ of internal nodes $v \in \mathcal{T}$, and we write ℓ_1 (resp. ℓ_2) for the label of the left (resp. right) child of the vertex with the label ℓ. Similarly, to the proof of Theorem 2.6.3, there exists some $(\ell_1, \ell_2) \in J(\mathcal{T})$

that satisfies

$$\mathop{\mathbb{E}}_{S\sim\Pi_k^m}\mathop{\mathbb{E}}_{\{\mathbf{Y}_S\}}\mathop{\mathbb{E}}_{\{t_1,t_2\}\sim w_{\ell_1,\ell_2}}\left\|\{\mathbf{Y}_{t_1\sqcup t_2}\mid\mathbf{Y}_S\}-\{\mathbf{Y}_{t_1}\mid\mathbf{Y}_S\}\{\mathbf{Y}_{t_2}\mid\mathbf{Y}_S\}\right\|_1\geq\frac{\varepsilon}{2k}.$$

Now, analogously to Eq. (2.15), using $\ell_1\leq k$ using we have

$$\mathop{\mathbb{P}}_{\substack{S\sim\Pi_k^m\\{\mathbf{Y}_S\}}}\left[\mathop{\mathbb{E}}_{\mathfrak{a}\in\Pi_{\ell_1,\ell_2}}\left[\begin{array}{c}\mathbb{E}_{t_1\in X(\ell_1)}[\mathrm{Var}[\mathbf{Y}_{t_1}\mid\mathbf{Y}_S]-\mathrm{Var}[\mathbf{Y}_{t_1}\mid\mathbf{Y}_S,\mathbf{Y}_\mathfrak{a}]]\\+\quad\mathbb{E}_{t_2\in X(\ell_2)}[\mathrm{Var}[\mathbf{Y}_{t_2}|\mid\mathbf{Y}_S]-\mathrm{Var}[\mathbf{Y}_{t_2}\mid\mathbf{Y}_S,\mathbf{Y}_\mathfrak{a}]]\end{array}\right]\geq c_5\cdot\frac{\varepsilon^4}{256\cdot k^4\cdot q^{4k}\cdot r}\right]\geq\frac{\varepsilon}{4k},\tag{2.18}$$

Using the same arguments in the proof of Theorem 2.6.3, we can get that

$$\Phi_m-\Phi_{m+1}\geq\frac{1}{k}\cdot\frac{\varepsilon}{4k}\cdot\frac{c_5\cdot\varepsilon^4}{512\cdot k^6\cdot q^{4k}\cdot r}=c_5\cdot\frac{\varepsilon^5}{2048\cdot k^4\cdot q^{4k}\cdot r}.$$

Again, this would mean that there can be at most $2048\cdot k^6\cdot q^{4k}\cdot r/(\varepsilon^5\cdot c_5)$ indices m such that $\varepsilon_{m/2}\geq\varepsilon/2$. In particular,

$$\mathop{\mathbb{E}}_{m\in[L/k]}[\varepsilon_m]\leq\frac{\varepsilon}{2}+\frac{k}{L}\cdot\frac{2048\cdot k^6\cdot q^{4k}\cdot r}{\varepsilon^5\cdot c_5}.$$

i.e. there exists a universal constant $C''\geq 0$, such that

$$L\geq C''\cdot\frac{k^7\cdot q^{4k}\cdot r}{\varepsilon^5}\text{ ensures }\mathop{\mathbb{E}}_{m\sim[L/k]}\varepsilon_m\leq\varepsilon.$$

■

2.8 Quantum k-local Hamiltonian

Our k-CSP results extend to the quantum setting generalizing the approximation scheme for 2-local Hamiltonians on bounded degree low threshold rank graphs from Brandão and Harrow [BH13] (BH). Before we can make the previous statement more precise we

will need to introduce some notation. A well studied quantum analogue of classical k-CSPs are the so-called quantum k-local Hamiltonians [AAV13].

Definition 2.8.1 (k-local Hamiltonian)**.** *We say that* $\mathsf{H} = \mathbb{E}_{\mathfrak{s}\sim\Pi_k}\mathsf{H}_{\mathfrak{s}}$ *is an instance of the k-local Hamiltonian problem over q-qudits on ground set* $[n]$ *if there is a distribution* Π_k *on subsets of size k of* $[n]$ *such that for every* $\mathfrak{s} \in \mathrm{Supp}(\Pi_k)$ *there is an Hermitian operator* $\mathsf{H}_{\mathfrak{s}}$ *on* $\mathbb{C}^{q^n}$ *with* $\|\mathsf{H}_{\mathfrak{s}}\|_{\mathrm{op}} \leq 1$ *and acting (possibly) non-trivially on the q-qudits of* $\mathfrak{s}$ *and trivially on* $[n] \setminus \mathfrak{s}$*.*

Given an instance $\mathsf{H} = \mathbb{E}_{\mathfrak{s}\sim\Pi_k}\mathsf{H}_{\mathfrak{s}}$ of the k-local Hamiltonian problem on ground set $[n]$, the goal is to provide a good (additive) approximation to the *ground state energy* $e_0(\mathsf{H})$, i.e., the smallest eigenvalue of H. Equivalently, the goal is to approximate

$$e_0(\mathsf{H}) = \min_{\rho\in D\left(\mathbb{C}^{q^n}\right)} \mathrm{Tr}(\mathsf{H}\rho),$$

where $D\left(\mathbb{C}^{q^n}\right)$ is the set of density operators, PSD operators of trace one, on $\mathbb{C}^{q^n}$. The eigenspace of H associated to $e_0(\mathsf{H})$ is called the *ground space* of H.

Remark 2.8.2. *The locality k of a k-local Hamiltonian has a similar role as the arity of k-CPSs whereas the qudit dimension q has the role of alphabet size. Observe that for a k-CSP the goal is to maximize the fraction of satisfied constrains while for a k-local Hamiltonian the goal is to minimize the energy (constraint violations).*

We will need an informationally complete measurement Λ modeled as a channel

$$\Lambda\colon D\left(\mathbb{C}^q\right) \to D\left(\mathbb{C}^{q^8}\right),$$

and defined as

$$\Lambda(\rho) \coloneqq \sum_{y\in\mathcal{Y}} \mathrm{Tr}(\mathsf{M}_y\rho)\cdot e_y e_y^\dagger,$$

where $\{\mathsf{M}_y\}_{y\in\mathcal{Y}}$ is a POVM [8] and $\{e_y\}_{y\in\mathcal{Y}}$ is an orthonormal basis (see Lemma 2.8.8 below for the properties of Λ). Recall that an informationally complete measurement is an injective channel, i.e., the probability outcomes $p(y) = \mathrm{Tr}(\mathsf{M}_y\rho)$ fully determine ρ. By definition given this probability distribution $\{p(y)\}_{y\in\mathcal{Y}}$ we can uniquely determine ρ. We use the notation $\rho = \Lambda^{-1}\left(\{p(y)\}_{y\in\mathcal{Y}}\right)$ for the recovered state from probability outcomes $\{p(y)\}_{y\in\mathcal{Y}}$.

BH using the informationally complete measurement Λ reduced the quantum 2-local Hamiltonian problem to a classical problem involving PSD ensembles of indicator random variables of outcomes $\mathcal{Y}$ of Λ. In this reduction, they had to ensure that the local distributions encoded by these indicators random variables are indeed consistent with probability distributions of outcomes arising from actual local density matrices. Note that the channel Λ is only injective, an arbitrary probability distribution on $\mathcal{Y}$ may not correspond to a valid quantum state. For this reason, they introduced a new SDP hierarchy to find this special kind of PSD ensemble, which we refer to as *quantum PSD ensemble*, minimizing the value of the given input k-local Hamiltonian instance.

Using our k-CSP approximation scheme for low threshold rank hypergraphs, we show that product state approximations close to the ground space of k-local Hamiltonians on bounded degree low threshold rank hypergraphs can be computed efficiently in polynomial time by Algorithm 2.8.3. Our result is a generalization of the $k = 2$ case of Brandão and Harrow [BH13] for 2-local Hamiltonians on bounded degree low threshold rank graphs. Their algorithm is based on the 2-CSP result from [BRS11].

8. A POVM is a collection of operators $\{\mathsf{M}_y\}_{y\in\mathcal{Y}}$ such that $\sum_{y\in\mathcal{Y}} M_y = \mathsf{I}$ and $(\forall y \in \mathcal{Y})(\mathsf{M}_y \succeq 0)$.

Algorithm 2.8.3 (Quantum Propagation Rounding Algorithm)**.**

Input *L-local quantum PSD ensemble* [a] $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ *and distribution* Π *on* $X(\leq \ell)$.

Output *A random state* $\rho = \rho_1 \otimes \ldots \otimes \rho_n$ *where each* $\rho_i \in D(\mathbb{C}^q)$.

1. *Choose* $m \in \{1, \ldots, L/\ell\}$ *at random.*
2. *Independently sample* m ℓ*-faces,* $\mathfrak{s}_j \sim \Pi$ *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* $\sigma_S : S \to [q]$ *according to the local distribution,* $\{\mathbf{Y}_S\}$.
5. *Set* $\mathbf{Y}' = \{\mathbf{Y}_1, \ldots \mathbf{Y}_n | \mathbf{Y}_S = \sigma_S\}$*, i.e. the local ensemble* $\mathbf{Y}$ *conditioned on agreeing with* σ_S.
6. *For all* $j \in [n]$*, set* $\rho_j = \Lambda^{-1}(\{\mathbf{Y}'_j\})$.
7. *Output* $\rho = \rho_1 \otimes \ldots \otimes \rho_n$.

a. We define the quantum ensemble as the PSD ensemble produced by the SDP hierarchy of [BH13]

The precise result is given in Theorem 2.8.4.

Theorem 2.8.4. *Suppose* $\mathfrak{I} = (\mathsf{H} = \mathbb{E}_{\mathfrak{s} \sim \Pi_k} \mathsf{H}_{\mathfrak{s}})$ *is a* q*-qudit* k*-local Hamiltonian instance whose constraint complex* [9] *is* $X(\leq k)$ *and has bounded normalized degree, i.e.,* $\Pi_1 \leq \delta$*. Let* $\tau = c_4 \cdot (\varepsilon^2/(16k^2q^{8k}))^2$*, for* $\varepsilon > 0$*. There exists an absolute constant* C' *that satisfies the following:*

Set $L = (\frac{C' \cdot k^5 \cdot q^{8k} \cdot \mathrm{rank}_\tau(\mathfrak{I})}{\varepsilon^4})$*. Then there is an algorithm based on* L*-levels of SoS-hierarchy and Algorithm 2.8.3 that outputs a random product state* $\rho = \rho_1 \otimes \ldots \otimes \rho_n$ *that in expectation ensures*

$$\mathrm{Tr}(\mathsf{H}\rho) \leq e_0(\mathsf{H}) + (18q)^{k/2} \cdot \varepsilon + L \cdot k \cdot \delta,$$

where $e_0(\mathsf{H})$ *is the ground state energy of* H.

9. We define the constraint complex of a k-local Hamiltonian in the same way we define it for k-CSPs, namely, by taking the downward closure of the support of Π_k.

Remark 2.8.5. *Similarly to the classical case, Theorem 2.8.4 serves as a no-go barrier (in its parameter regime) to the quantum local-Hamiltonian version of the quantum PCP Conjecture [AAV13]. In particular, k-local Hamiltonians on bounded degree γ-HDXs for γ sufficiently small can be efficiently approximated in polynomial time.*

Now we sketch a proof of Theorem 2.8.4. We provide a sketch rather than a full proof since Theorem 2.8.4 easily follows from the BH analysis once the main result used by them, Theorem 5.6 from [BRS11], is appropriately generalized to "break" k-wise correlations as accomplished by our Theorem 3.9.19 (restated below for convenience). Furthermore, a full proof would require introducing more objects and concepts only needed in this simple derivation (the reader is referred to [BH13] for the quantum terminology and the omitted details).

Theorem 2.8.6 (Adaptation of Theorem 3.9.19)**.** *Suppose a simplicial complex $X(\leq k)$ with $X(1) = [n]$ and an L-local PSD ensemble $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ are given. There exists some universal constants $c_4 \geq 0$ and $C'' \geq 0$ satisfying the following: If $L \geq C'' \cdot (q^{4k} \cdot k^7 \cdot r/\varepsilon^5)$, $\mathrm{Supp}(\mathbf{Y}_j) \leq q$ for all $j \in [n]$, and $\mathfrak{I}$ is $(c_4 \cdot (\varepsilon/(4k \cdot q^{8k}))^4, r)$-splittable. Then, we have*

$$\mathop{\mathbb{E}}_{\mathfrak{a} \in X(k)} \left[\left\| \{\mathbf{Y}'_{\mathfrak{a}}\} - \left\{\mathbf{Y}'_{a_1}\right\} \cdots \left\{\mathbf{Y}'_{a_k}\right\} \right\|_1 \right] \leq \varepsilon, \tag{2.19}$$

where $\mathbf{Y}'$ is as defined in Algorithm 2.8.3 on the input of $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ and Π_k.

Once in possession of the *quantum PSD ensemble* the problem becomes essentially classical. The key result in the BH approach is Theorem 5.6 from [BRS11] that brings (in expectation under conditioning on a random small seed set of qudits) the local distributions, over the edges of the constraint graph of a 2-local Hamiltonian, close to product distributions [10]. Now, using the fact that they have an informationally complete measurement

10. For this to hold we need the underlying constraint graph to be low threshold rank and the SoS degree to be sufficiently large

Λ they can "lift" the conditioned marginal distribution on each qudit $\{\mathbf{Y}'_j\}$ to an actual quantum state as $\rho_j = \Lambda^{-1}(\{\mathbf{Y}'_j\})$ (see Algorithm 2.8.3). In this lifting process, they pay an **average** distortion cost of $18q \cdot \varepsilon$ (for using the marginal over the qudits). For k-local Hamiltonians, the distortion of k q-qudits is given by Lemma 2.8.7 (stated next without proof).

Lemma 2.8.7. *Let $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ be random variables in an L-local quantum PSD ensemble with $L \geq k$. Suppose that*

$$\varepsilon := \left\| \{\mathbf{Z}_1, \ldots, \mathbf{Z}_k\} - \prod_{i=1}^{k} \{\mathbf{Z}_i\} \right\|_1 .$$

Then

$$\left\| \left(\Lambda^{\otimes k}\right)^{-1} (\{\mathbf{Z}_1, \ldots, \mathbf{Z}_k\}) - \left(\Lambda^{\otimes k}\right)^{-1} \left(\prod_{i=1}^{k} \{\mathbf{Z}_i\} \right) \right\|_1 \leq (18q)^{k/2} \cdot \varepsilon.$$

Note that Lemma 2.8.7 is a direct consequence of Lemma 2.8.8 from [BH13].

Lemma 2.8.8 (Informationally complete measurements (Lemma 16 [BH13]))**.** *For every positive integer q there exists a measurement Λ with $\leq q^8$ outcomes such that for every positive integer k and every traceless operator ξ, we have*

$$\|\xi\|_1 \leq (18q)^{k/2} \left\| \Lambda^{\otimes k}(\xi) \right\|_1 .$$

BH also pay a full cost for each local term in the Hamiltonian that involves a seed qudit since its state was not reconstructed using the full distribution of a qudit given by the quantum ensemble but rather reconstructed from a single outcome $y \in \mathcal{Y}$ of Λ. Naively, this means that the final state of this qudit may be far from the intended state given by SDP relaxation. In our case, we assume that the normalized degree satisfies $\Pi_1 \leq \delta$. Therefore, the total error from constraints involving seed qudits is at most

$$L \cdot k \cdot \delta.$$

Putting the above pieces together we conclude the proof (sketch) of Theorem 2.8.4.

CHAPTER 3

LIST DECODING OF DIRECT SUM CODES

3.1 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 $\mathcal{C}' = \mathrm{dsum}^g_{X(k)}(\mathcal{C})$, 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 3.1.1 (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

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.

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 3.1.2 (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 $(1-\varepsilon)$.

Remark 3.1.3. *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 3.1.4 (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 $\gamma \leq \varepsilon^{O(1)}$.*

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.

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 3.1.1, Corollary 3.1.2, and Theorem 3.1.4 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 3.1.1 to explicit objects derived from Ramanujan complexes [LSV05b, LSV05a] and applying Theorem 3.1.4 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 3.7.2 of Section 3.7 and in Corollary 3.9.5 of Section 3.9, 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 [RY20] 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 4.3 after introducing some objects and notation in Section 4.2. In Section 4.3, 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 3.1.1) captures and strengthens some of their parameters in Corol-

lary 3.1.2. 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 [RY20] 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.

3.2 Preliminaries

3.2.1 Simplicial Complexes

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

Definition 3.2.1. *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*

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.

$X(0) := \{\varnothing\}$.

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

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

Definition 3.2.2 (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 3.9.1).

3.2.2 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 $\Delta(z,z') := \left|\{i \mid z_i \neq z'_i\}\right|/n$. Any set $\mathcal{C} \subset \Sigma^n$ gives rise to a q-ary code. The distance of $\mathcal{C}$ is defined as $\Delta(\mathcal{C}) := \min_{z \neq z'} \Delta(z,z')$ where $z, z' \in \mathcal{C}$. We say that $\mathcal{C}$ is a linear code [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 3.2.3 (Bias). *The* bias *of a word* [5] $z \in \mathbb{F}_2^n$ *is* $\mathrm{bias}(z) := \left|\mathbb{E}_{i\in[n]}(-1)^{z_i}\right|$. *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 3.2.4 (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 $\mathrm{dsum}^g_{X(k)}(z) = y$ 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 $\mathfrak{s}$.*

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

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

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*.

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

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

Definition 3.2.5 (Robust Lifting). *We say that* $\mathrm{dsum}^{g}_{X(k)}$ *is* (δ_0, δ)*-robust if for every* $z, z' \in \Sigma_1^{X(1)}$ *we have*

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

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 3.2.6 (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 3.2.7 (Parity Sampler). *Let* $g \colon \mathbb{F}_2^k \to \mathbb{F}_2$*. We say that* $\mathrm{dsum}^{g}_{X(k)}$ *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* $\mathrm{bias}(\mathrm{dsum}(z)) \leq \beta$*.*

3.2.3 Constraint Satisfaction Problems (CSPs)

A k-CSP instance $\mathcal{J}(H, \mathcal{P}, w)$ 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 $\mathsf{SAT}_{\mathcal{J}}(\sigma)$ 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:

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

3.2.4 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 3.2.8. *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

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

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 $\mathbb{P}\left[\mathbf{Z}_i = a\right] = \widetilde{\mathbb{E}}[\mathbf{Z}_{i,a}] > 0$. This can be done by considering the new operator

$$\widetilde{\mathbb{E}}_{|Z_i = a} \colon \mathcal{R}^{\leq 2t-2} \to \mathbb{R}$$

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

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 $X(1)$.
- $\mathbf{Y} \coloneqq \{\mathbf{Y}_\mathfrak{s} \coloneqq (\mathrm{dsum}(\mathbf{Z}))_\mathfrak{s} \mid \mathfrak{s} \in X(k)\}$ is a local ensemble on $X(k)$.

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

3.3 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 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 $\mathrm{dsum}^g_{X(k)}$: (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 $\mathcal{C}_k = \mathrm{dsum}(\mathcal{C}_1)$ 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 $\left|\langle y', y \rangle\right| \geq 2 \cdot \varepsilon$. 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

9. The precise SOS program used is given in Section 3.6.2.

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

to the robustness properties of the lifting and the fact that we actually have "coupled" pairs $(z, y = \mathrm{dsum}(z))$ and $(z', y' = \mathrm{dsum}(z'))$ 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 $y = \mathrm{dsum}(z)$. Furthermore, if $\mathcal{C}_1$ admits an efficient unique decoder, we can show that this step in list decoding $\tilde{y}$ can be done efficiently.

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

rized in Fig. 3.1 along with the required assumptions on the lifting.

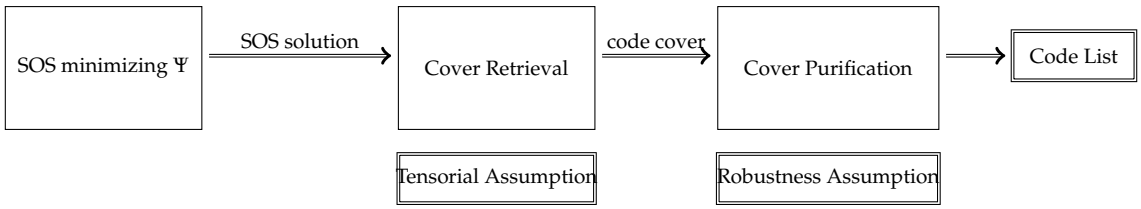

Figure 3.1: List decoding framework with the assumptions required in each stage.

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

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

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 3.4, 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 3.5 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 3.6. Next, we instantiate the framework with the direct sum lifting on HDXs in Section 3.7. As an interlude between the first and second instantiation, Section 3.8 describes how the first concrete instantiation of Section 3.7 captures the direct product lifting on HDXs via a reduction to the direct sum lifting. Finally, in Section 3.9, we show how to instantiate the framework with the direct sum lifting on the collection of length $k-1$ walks of an expander graph.

3.4 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 $1/2$.

3.4.1 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 3.4.1 (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.

3.4.2 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 3.4.2 (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 3.4.3 (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 3.4.4 (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 3.4.5 (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 $X(i)$.

3.4.3 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 3.4.6 (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

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

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

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

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

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

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

so we have

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

■

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

Definition 3.4.7 (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 $\mu = \mathbb{E}_{v \sim \Pi_V} f(v)$,*

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

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 3.4.8 (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 3.4.6, 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 3.4.9 (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*

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

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

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

By Claim 3.4.6, the magnitude of the expectation of $(-1)^{z_i}$ over the edges of size k in the complete complex $X|_\mathfrak{t}$ is close to $\left|\mathbb{E}_{i\sim X|_\mathfrak{t}(1)}(-1)^{z_i}\right|$, which is just the bias of $z|_\mathfrak{t}$. Then

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

■

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

Lemma 3.4.10 (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 3.4.8 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 3.4.3, it suffices to have

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

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 3.4.9, $X(k)$ 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. ∎

3.4.4 *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 3.4.11 (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 3.4.10, 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 3.4.5, 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 3.4.10 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 3.4.11, 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 3.4.1.

Lemma 3.4.12 (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 3.4.1 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 $r_k \geq r_1 \cdot \beta^{O(1)}$.

∎

3.5 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.

3.5.1 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 3.5.1, 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 3.5.1. *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 3.5.2. *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 $\mathcal{I}(X(k), \tilde{y})$ allows us to recover $z^\star \in \mathcal{C}_1$ provided $\tilde{y}$ is sufficiently close to $\mathcal{C}_k$.

Corollary 3.5.3. *Suppose $\mathcal{C}_1$, $X(k)$, $z^\star$, $y^\star$ and $\tilde{y}$ are as in the assumptions of Lemma 3.5.1. If $z \in \mathbb{F}_2^n$ is such that*

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

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

Proof. By the assumption on z, we have

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

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 3.5.1. ∎

3.5.2 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 $\mathfrak{J}(X(k), \tilde{y})$ and obtain $z \in \mathbb{F}_2^n$.

Theorem 3.5.4 ([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 $X_{\mathfrak{J}}$ 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 $(\mathsf{OPT}_{\mathfrak{J}} - \beta)$ 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 3.5.3 to achieve efficient unique decodability of $\mathcal{C}_k = \mathrm{dsum}_{X(k)}(\mathcal{C}_1)$.

Corollary 3.5.5. *Let $X(\leq d)$ be a d-dimensional γ-HDX satisfying the premises of Lemma 3.4.10 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 3.4.10, 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 3.5.4 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 3.5.3 to find z^* and uniquely decode $\tilde{y}$ as $y^* = \mathrm{dsum}_{X(k)}(z^*)$. ∎

Expander Walks

In Section 3.9, 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 3.9.21), which occurs when $X(k)$ is a set of walks on on a suitably strong expander (Corollary 3.9.18). In particular, we have the following.

Theorem 3.5.6. *Let $G = (V, E)$ be a graph with $\sigma_2(G) = \lambda$ and k be a given parameter. Let $\mathfrak{I}$*

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

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 $n^{O(q^{4k}\cdot k^7/\beta^5)}$.

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 3.5.7. *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 3.4.1, 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 3.5.6 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 3.5.3 to find z^* and uniquely decode $\tilde{y}$ as $y^* = \mathrm{dsum}_{X(k)}(z^*)$. ∎

Remark 3.5.8. *In both Corollary 3.5.5 and Corollary 3.5.7, 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* $\Delta(\mathcal{C}_k)/2$*.*

3.6 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 3.6.1 and use it to define the SOS program for list decoding in Section 3.6.2. In Section 3.6.3, we establish key properties of Ψ capturing its importance as a list decoding tool. We recall the Propagation Rounding algorithm in Section 3.6.4 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 3.6.5. 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 3.6.6. This abstract framework will be instantiated using the direct sum lifting: on HDXs in Section 3.7 and on expander walks in Section 3.9.

3.6.1 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 3.6.1. A similar

list decoding technique was also (independently) used by Karmalkar et al. [KKK19] and Raghavendra–Yau [RY20], but in the context of learning.

Definition 3.6.1 (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*

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

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* $\mathbf{Y}$.

3.6.2 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 $\mathcal{C}_k = \mathrm{dsum}(\mathcal{C}_1)$. 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 3.1: List decoding SOS formulation for $\tilde{y}$.

3.6.3 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 3.6.2 (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 $\mathfrak{s} \cup \mathfrak{t} = \{i_1, \ldots, i_t\}$. By definition $\mathbf{Y}_\mathfrak{s}\mathbf{Y}_\mathfrak{t} = \mathrm{dsum}(\mathbf{Z})_\mathfrak{s} \cdot \mathrm{dsum}(\mathbf{Z})_\mathfrak{t}$, i.e., $\mathbf{Y}_\mathfrak{s}\mathbf{Y}_\mathfrak{t}$ is a function f on input $\mathbf{Z}_{i_1}, \ldots, \mathbf{Z}_{i_t} \in \{\pm 1\}$. Let

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

be the Fourier decomposition of f. Then

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

Since $\widetilde{\mathbb{E}}\left[\mathbf{Y}_\mathfrak{s}\mathbf{Y}_\mathfrak{t}\right]$ is a linear function of $\widetilde{\mathbb{E}}$, we obtain $\left(\widetilde{\mathbb{E}}\left[\mathbf{Y}_\mathfrak{s}\mathbf{Y}_\mathfrak{t}\right]\right)^2$ 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.

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].

Lemma 3.6.3 (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

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

■

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

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

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

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

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

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

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

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

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 $\mathbb{E}_{\delta_z}$.

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 $\Psi \leq 1$. ∎

3.6.4 *Propagation Rounding*

A central algorithm in our list decoding framework is the Propagation Rounding Algorithm 4.7.16. 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 4.7.16 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 4.7.16 follows.

Algorithm 3.6.5 (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 4.7.16 is completely determined by the tuple (m, S, σ) which we will refer to as a slice of the PSD ensemble.

Definition 3.6.6 (Slice)**.** *We call a tuple (m, S, σ) obtainable by Algorithm 4.7.16 a slice and let Ω denote the set of all slices obtainable by Algorithm 4.7.16.*

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 4.7.16. We also define a pseudo-expectation operator for each slice.

Definition 3.6.7 (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* $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n | \mathbf{Z}_s = \sigma\}$.

3.6.5 Tensorial Structures

In general, a local PSD ensemble $\mathbf{Z}' = \{\mathbf{Z}'_1, \ldots, \mathbf{Z}'_n\}$ output by the Propagation Rounding Algorithm 4.7.16 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 3.6.8 (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 the local PSD ensemble $\mathbf{Z}'$ returned by Propagation Rounding Algorithm 4.7.16 with SOS degree parameter L satisfies*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{\mathfrak{a} \sim \Pi_k} \left\| \{\mathbf{Z}'_{\mathfrak{a}}\} - \left\{\mathbf{Z}'_{a_1}\right\} \cdots \left\{\mathbf{Z}'_{a_k}\right\} \right\|_1 \leq \mu. \tag{3.1}$$

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 3.6.9 (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 Propa-*

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].

gation Rounding Algorithm 4.7.16 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 3.7.1, 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 3.9.1 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 3.7.4 establishes that HDXs of appropriate expansion satisfy this assumption, and Lemma 3.9.20 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 3.6.10 (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 3.6.11 (Lifted Variables Close to Pairwise Independent)**.** *Let $\mu \in (0,1]$. We define the event P_μ as*

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

These events satisfy a simple concentration property.

Claim 3.6.12 (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 4.7.16. Let $\mu \in (0,1]$. If $X(k)$ is $(\mu^4/4, L)$-two-step tensorial, then*

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[K_\mu^c\right] \leq \frac{\mu^2}{2}, \tag{3.2}$$

and

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[P_\mu^c\right] \leq \frac{\mu^2}{2}. \tag{3.3}$$

Proof. We only prove Eq. (3.2) since the proof of Eq. (3.3) is similar. Define the random variable $\mathbf{R} := \mathbb{E}_{\mathfrak{a}\sim\Pi_k} \left\| \{\mathbf{Z}'_\mathfrak{a}\} - \left\{\mathbf{Z}'_{a_1}\right\} \cdots \left\{\mathbf{Z}'_{a_k}\right\} \right\|_1$ 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. ∎

3.6.6 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 3.6.13 (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 3.6.14. *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 3.6.15 (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*

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

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 3.6.16 stated below whose building blocks and analysis we develop in this section.

Algorithm 3.6.16 (List Decoding Algorithm)**.**

Input *A word* $\tilde{y} \in \{\pm 1\}^{X(k)}$ *which is* $(1/2 - \sqrt{\varepsilon})$*-close to* $\mathcal{C}_k = \mathrm{dsum}(\mathcal{C}_1)$*.*

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 3.6.29 on* $\mathbf{Z}$
3. *Let* $\mathcal{L}'$ *be the output of the Cover Purification Algorithm 3.6.36 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. 3.1 of Section 4.3, 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 3.6.6. Next, the formal definition of cover is presented in Section 3.6.6 and we have all the elements to present the Cover Retrieval Algorithm 3.6.29 with its correctness in Section 3.6.6. Then, we use the *robustness* properties of the lifting to purify the cover in Section 3.6.6. Finally, in Section 3.6.6, we assemble the building blocks and prove the main technical result, Theorem 3.6.17, whose proof follows easily once the properties of the building blocks are in place.

Note that Theorem 3.6.17 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 3.7 and to the direct sum lifting on expander walks in Section 3.9.2.

Theorem 3.6.17 (List Decoding Theorem)**.** *Suppose that* dsum *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 3.6.16 returns the coupled code list* $\mathcal{L}(\widetilde{y}, \mathcal{C}_1, \mathcal{C}_k)$*. Furthermore, the running time is*

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

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

Remark 3.6.18. *Regarding Theorem 3.6.17, 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 3.6.17, 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 3.6.6 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.

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 3.6.20), then we can retrieve a $z \in \{\pm 1\}^{X(1)}$ such that $\mathrm{dsum}(z)$ 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 3.6.6.

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

Definition 3.6.19 (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 3.6.20 shows how to round the pseudo-expectation in a slice.

Lemma 3.6.20 (From fractional to integral in a slice)**.** *Let $(m, S, \sigma) \in \Omega$ be a slice. Suppose*

$$\mathop{\mathbb{E}}_{\mathfrak{a}\sim\Pi_k} \left\|\{\mathbf{Z}_{\mathfrak{a}}|\mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{a_1}|\mathbf{Z}_S = \sigma\}\cdots\{\mathbf{Z}_{a_k}|\mathbf{Z}_S = \sigma\}\right\|_1 \leq \mu, \tag{3.4}$$

and

$$\mathop{\mathbb{E}}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2} \|\{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{\mathfrak{s}}|\mathbf{Z}_S = \sigma\}\{\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\}\|_1 \leq \mu. \tag{3.5}$$

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

$$\mathbb{P}_{z\sim\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}} \left[\left| \mathbb{E}_{\mathfrak{s}\sim\Pi_k} y_{\mathfrak{s}} \cdot \mathrm{dsum}(z)_{\mathfrak{s}} \right| \ge \sqrt{1-\beta} \cdot \kappa \right] \ge \frac{\beta \cdot \kappa^2}{4}. \tag{3.6}$$

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

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

From our assumptions Eq. (3.4) and Eq. (3.5), it follows that $\mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k^2} \mu_{\mathfrak{s},\mathfrak{t}} \le 3 \cdot \mu$. Using the fact that $|y_{\mathfrak{s}} y_{\mathfrak{t}}| = 1$ and Hölder's inequality, we get

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

Alternatively,

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

Define the random variable $\mathbf{R} := \left(\mathbb{E}_{\mathfrak{s}\sim\Pi_k} [y_{\mathfrak{s}} \cdot \mathrm{dsum}(z)_{\mathfrak{s}}] \right)^2$. Using Fact C.1.1 with approximation parameter $\beta/2$, we get

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

from which Eq. (3.6) 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 3.6.21 (y-Correlated Event)**.** *Let $\kappa \in (0,1]$ and $y \in \{\pm 1\}^{X(k)}$. 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 4.7.18.

Definition 3.6.22 (y-Recoverable Event)**.** *Let $\kappa, \mu \in (0,1]$ and $y \in \{\pm 1\}^{X(k)}$. We define the event $R_{\kappa,\mu}(y)$ as*

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

Lemma 3.6.20 motivates the following "recoverability" condition.

Definition 3.6.23 (Recoverable Word)**.** *Let $\kappa, \mu \in (0,1]$ and $y \in \{\pm 1\}^{X(k)}$. 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 3.6.24 (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 $y = \mathrm{dsum}(z)$. 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 3.6.12, we get

$$\mathop{\mathbb{P}}_{(m,S,\sigma)\sim\Omega}\left[C_\kappa(y)^c\right] \;\geq\; 1-\mu^2. \tag{3.7}$$

Since $\widetilde{\mathbb{E}}$ is a valid solution to the List Decoding Program, Lemma 3.6.3 implies the lower bound

$$\Psi\left(\{\mathbf{Y}_\mathfrak{s}\}_{\mathfrak{s}\in X(k)}\right) \;\geq\; 16\cdot\varepsilon^2. \tag{3.8}$$

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

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

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 3.6.4 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

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

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

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 3.6.24. 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

$$|\mathbb{E}_{\mathfrak{s} \sim \Pi_k} y_{\mathfrak{s}} \cdot y'_{\mathfrak{s}}| > C \cdot \varepsilon, \tag{3.9}$$

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 $y = \mathrm{dsum}(z)$. The second property is that the lifting is robust: even the weak agreement given by Eq. (3.9)

translates into a much stronger agreement in the ground set between z and $z' \in \mathcal{C}_1$ where $y' = \mathrm{dsum}(z')$. 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 3.6.25 (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* dsum *provided $y = \mathrm{dsum}(z)$.*

Remark 3.6.26. *If the function* dsum *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 3.6.27 (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* dsum *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 3.6.28 (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*

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

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.

Cover Retrieval

When the code list $\mathcal{L}(\tilde{y}, \mathcal{C}_k)$ becomes recoverable in the SOS sense as per Lemma 3.6.24, 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 3.6.20), so we need to repeat this process a few times to boost our success probability[16]. This is accomplished by Algorithm 3.6.29.

Algorithm 3.6.29 (Cover Retrieval Algorithm)**.**

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

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 3.6.29 follows easily given the properties established so far.

Lemma 3.6.30 (Cover lemma)**.** *Let $\beta \in (0,1)$. Suppose that* dsum *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 $(1/2 -$*

16. In fact, this process can be derandomized using standard techniques in our instantiations. See Lemma C.3.1 for detais.

$\sqrt{\varepsilon}$)-close to $\mathcal{C}_k$. *If* $\theta \le \beta \cdot \varepsilon/2^4$, *then w.v.h.p.*[17] *the Cover Retrieval algorithm 3.6.29 returns a* δ*-bias cover* $\mathcal{M}$ *of the code list* $\mathcal{L}(\widetilde{y}, \mathcal{C}_k)$ *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 $\mathbf{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_n\}$ be an η-optimum solution to the List Decoding Program where $\eta \le \theta^2\cdot\varepsilon^4$ 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 3.6.24 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 3.6.20 yields

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

where $\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}$ (c.f. Definition 3.6.19) 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 $z^{(1)}, \dots, z^{(T)}$ and thus also the coupled list

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

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

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

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

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

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

concluding the proof. ∎

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 3.6.29 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 $(z, y = \text{dsum}(z))$ and $(z', y' = \text{dsum}(z'))$ (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 3.6.31 (Coupled unique decoding from distance)**.** *Suppose that* dsum *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 3.6.32.

Claim 3.6.32 (Coupled unique decoding from bias I)**.** *Suppose* dsum *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 $\mathrm{dsum}(z'') = \mathrm{dsum}(z) \cdot \mathrm{dsum}(z')$. Since the lifting takes bias $1/2 + \varepsilon_0$ to $2 \cdot \varepsilon$, we have

$$\mathrm{bias}(\mathrm{dsum}(z) \cdot \mathrm{dsum}(z')) = \mathrm{bias}(\mathrm{dsum}(z'')) \leq 2 \cdot \varepsilon,$$

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 3.6.33.

Claim 3.6.33 (Coupled unique decoding from bias II)**.** *Suppose* dsum *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.,* $\mathrm{dsum}(-z) = -\mathrm{dsum}(z)$*. 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

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

or

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

Then either $\Delta(y, \mathrm{dsum}(z')) \leq 1/2 - \varepsilon$ or $\Delta(y, \mathrm{dsum}(-z')) \leq 1/2 - \varepsilon$. Using Claim 3.6.31

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

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

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

The correctness of the cover purification process is established next.

Lemma 3.6.35 (Purification lemma)**.** *Suppose* dsum *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

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

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 3.6.32 or Claim 3.6.33 yields that either z' or $-z'$ can be uniquely decoded in $\mathcal{C}_1$. Then

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

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 3.6.36.

Algorithm 3.6.36 (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* $y = \mathrm{dsum}(z)$
6. *Set* $\mathcal{L}' = \mathcal{L}' \cup \{(z, y)\}$
7. *Output* $\mathcal{L}'$.

Correctness of the List Decoding Algorithm

The building blocks developed so far are assembled to form the final list decoding algorithm (Algorithm 3.6.16), which is restated below for convenience.

Algorithm 3.6.37 (List Decoding Algorithm)**.**

Input *A word* $\tilde{y} \in \{\pm 1\}^{X(k)}$ $(1/2 - \sqrt{\varepsilon})$*-close to* $\mathcal{C}_k = \mathrm{dsum}(\mathcal{C}_1)$

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 3.6.29 on* $\mathbf{Z}$
3. *Let* $\mathcal{L}'$ *be the output of the Cover Purification Algorithm 3.6.36 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 3.6.38 (List Decoding Theorem (Restatement of Theorem 3.6.17))**.** *Suppose that* dsum *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 3.6.16 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 3.6.30 establishes that the Cover Retrieval Algorithm 3.6.29 returns w.v.h.p. a 2ε-bias cover. Then, Lemma 3.6.35 states that providing this 2ε-bias cover as input to the Cover Purification Algorithm 3.6.36 yields a coupled list containing the code list $\mathcal{L}(\widetilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. Finally, the last step in Algorithm 3.6.16 ensures the output is precisely $\mathcal{L}(\widetilde{y}, \mathcal{C}_1, \mathcal{C}_k)$. ∎

3.7 Instantiation I: Direct Sum on HDXs

We instantiate the list decoding framework to the direct sum lifting on HDXs obtaining Theorem 3.7.1, 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 3.7.1.

Theorem 3.7.1 (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 3.6.15) 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 3.7.1 on HDXs extracted from Ramanujan complexes (as in Lemma 3.4.5), we obtain Corollary 3.7.2.

Corollary 3.7.2. *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 3.6.15) 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 3.7.1, 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 3.4.11 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

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

■

3.7.1 HDXs are Two-Step Tensorial

Theorem 3.7.3 proven in [AJT19] establishes that HDXs of appropriate expansion parameter are tensorial objects for constant $L = O_{k,q,\mu}(1)$.

Theorem 3.7.3 (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 3.7.4 (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 3.7.3 (this is the only place where the assumption on γ is used), so we only need to show

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

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

the potential

$$\Phi_m = \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\sigma\sim\{\mathbf{Z}_S\}} \mathop{\mathbb{E}}_{\mathfrak{s}\sim\Pi_k} \mathrm{Var}\left[\mathbf{Z}_{\mathfrak{s}} \mid \mathbf{Z}_S = \sigma\right], \tag{3.10}$$

and consider the error term

$$\mu_m \coloneqq \mathop{\mathbb{E}}_{S\sim\Pi_k^m} \mathop{\mathbb{E}}_{\sigma\sim\{\mathbf{Z}_S\}} D(S,\sigma), \tag{3.11}$$

where $D(S,\sigma) \coloneqq \mathbb{E}_{\mathfrak{s},\mathfrak{t}\sim\Pi_k}[\|\{\mathbf{Z}_{\mathfrak{s}}\mathbf{Z}_{\mathfrak{t}} \mid \mathbf{Z}_S = \sigma\} - \{\mathbf{Z}_{\mathfrak{s}}|\mathbf{Z}_S = \sigma\}\{\mathbf{Z}_{\mathfrak{t}}|\mathbf{Z}_S = \sigma\}\|_1]$. 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 $\mathbb{E}_{m\in[L/k]}[\mu_m] \leq \mu$. ∎

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

3.7.2 Instantiation to Linear Base Codes

First, we instantiate the list decoding framework to the seemingly simpler case of binary *linear* base codes in Lemma 3.7.5. As we show later, with a simple observation we can essentially use the proof of Lemma 3.7.5 to obtain Theorem 3.7.1 for general codes.

Lemma 3.7.5 (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 3.6.15) 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.*

Proof. We show that under our assumption on the γ-HDX $X(\leq d)$ we obtain sufficient

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.

robustness and *tensorial* parameters to apply Theorem 3.6.17. 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 3.6.17, 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 3.4.10 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 3.7.3 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 3.6.17 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

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

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

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

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

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 3.6.17 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. ∎

3.7.3 Instantiation to General Base Codes

We can extend Lemma 3.7.5 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 3.7.6. *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 3.7.1.

Theorem 3.7.7 (Direct Sum Lifting on HDX (Restatement of Theorem 3.7.1))**.** *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 3.6.15) 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 3.7.5, it is enough to consider when $\mathcal{C}_1$ is not linear. Note that in the proof of Lemma 3.7.5 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 3.7.6 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 3.7.1 is now the same as the proof of Lemma 3.7.5. ∎

3.8 List Decoding Direct Product Codes

3.8.1 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 3.8.1 (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 $\mathrm{dprod}_{X(\ell)}(z) = (x_\mathfrak{t})_{\mathfrak{t} \in X(\ell)}$, where $x_\mathfrak{t} = (z_i)_{i \in \mathfrak{t}}$. 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 3.4.8 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.

3.8.2 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 3.4.5) comes in. Recall that to obtain codewords in the direct product code $\mathrm{dprod}_{X(\ell)}(\mathcal{C}_1)$ 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 3.8.2 (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 3.8.3 (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 C.1.2 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 3.8.2 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 3.8.3 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 3.8.4 (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 3.8.5 (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 3.8.4 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 3.8.3, 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 3.8.4 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 3.8.6 (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 the direct sum code $\mathrm{dsum}_{X(k)}(\mathcal{C}_1)$ 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 3.7.1 (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 3.8.5, 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 3.6.14) 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 3.7.1, where the rate calculation requires a slight modification for dealing with the increased alphabet size and φ is the projection from Claim 3.7.6. ∎

Using Corollary 3.8.6 with HDXs obtained from Ramanujan complexes as in Corollary 3.7.2, 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 $\gamma = O(\exp(-1/\varepsilon))$.

3.9 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 3.9.1.

Definition 3.9.1 (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.,*

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

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 4.7.16 remains the same, but we need to establish the tensorial properties of $W_G(k)$ which is done in Section 3.9.1.

The main result of this section follows.

Theorem 3.9.2 (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 3.6.15) 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 3.9.2 to the explicit family of Ramanujan expanders of Lubotzky et al. from Theorem 3.9.3.

Theorem 3.9.3 (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 3.9.4 (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 3.9.5.

Corollary 3.9.5. *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 $\mathcal{C}_k^{(i)}$ of the form

$\mathcal{C}_k^{(i)} = \mathrm{dsum}_{X(k)}(\varphi(\mathcal{C}_1^{(i)}))$ *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 3.6.15) 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 3.9.4 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 3.9.3 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 3.9.2, we obtain the desired claims. ∎

3.9.1 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 3.9.1 in a manner tailored to the present case $X(k) = W_G(k)$. Then, in Section 3.9.1, 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 3.9.1, 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.

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 3.2.4 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 4.7.16. It is easy to show that the tensorial condition of Eq. (3.12) (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 3.9.1. For now, we take this observation for granted and use it to show how the Swap operators emerge in obtaining the inequality

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w_1}\right\} \cdots \left\{\mathbf{Z}'_{w_k}\right\} \right\|_1 \leq \mu \tag{3.12}$$

present in the definition of tensoriality.

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

given walk.

Definition 3.9.6 (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 3.9.7 (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 3.9.8 (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 3.9.1. 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 3.9.9 (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 3.9.7 on the second and third terms and sim-

plifying, we get

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

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. ∎

Swap Operators Arising from Expander Walks

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

Definition 3.9.10 (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*

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

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 3.9.11. *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 3.9.12. *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 $\left\langle f, \mathsf{S}^{\circ}_{k_1,k_2} g \right\rangle = \left\langle \mathfrak{S}_{k_1,k_2,f}, g \right\rangle$. On

one hand we have

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

On the other hand we have

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

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

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 3.9.13. *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 3.9.1). Solving these kinds of k-CSPs might be of independent interest. For the coding application, the threshold rank (Definition 3.9.14) will be one, i.e., we will be working with expander graphs.

Definition 3.9.14 (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*

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

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

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

where $\mathrm{rank}_{\geq\tau}(\mathsf{A})$ denotes the usual threshold rank of A as in Definition 3.9.14.

Definition 3.9.15 ($(\mathcal{T}, \tau, r)$-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 3.9.16. *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 3.9.17. *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 D.1.10 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 3.9.18. *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).*

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 3.9.19.

Theorem 3.9.19 (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*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim W_G(k)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w_1}\right\} \cdots \left\{\mathbf{Z}'_{w_k}\right\} \right\|_1 \leq \mu, \tag{3.13}$$

where $\mathbf{Z}'$ is as defined in Algorithm 4.7.16 on the input of $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ and Π_k.

Using Theorem 3.9.19, 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 3.9.20 (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 3.7.4 for HDXs, so we omit it. ∎

Interlude: Approximating k-CSP on Walk Constraints

Now, we digress to show how using Theorem 3.9.19 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 3.9.21. *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 4.7.16 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 4.7.16 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 3.9.19. $\mathbf{Z}$ satisfies

$$\mathsf{SDP}(\mathfrak{I}) = \mathop{\mathbb{E}}_{w \sim \Pi_k} \left[\mathop{\mathbb{E}}_{\{\mathbf{Z}_w\}} [\mathbf{1}[\mathbf{Z}_w \in \mathcal{P}_w]] \right] \geq \mathsf{OPT}(\mathfrak{I}). \tag{3.14}$$

Since the conditioning done on $\{\mathbf{Z}'\}$ is consistent with the local distribution, by law of total expectation and Eq. (3.14) we have

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim \Pi_k} \mathbf{1}[\mathbf{Z}'_w \in \mathcal{P}_w] = \mathsf{SDP}(\mathfrak{I}) \geq \mathsf{OPT}(\mathfrak{I}). \tag{3.15}$$

By Theorem 3.9.19 we know that

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \sim \Pi_k} \left\| \{\mathbf{Z}'_w\} - \{\mathbf{Z}'_{w_1}\} \cdots \{\mathbf{Z}'_{w_k}\} \right\|_1 \leq \mu. \tag{3.16}$$

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

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

By using Eq. (3.16), we can obtain

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

Using Eq. (3.15), we conclude

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

■

3.9.2 *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 3.9.22 (Walks on Expanders are Parity Samplers [TS17] (Restatement of Theorem 3.4.1))**.** *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 3.9.23 (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 $\gamma \leq \varepsilon^C$, 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 3.6.15) 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 3.7.5. 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 3.6.17. 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 3.6.17, 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 3.9.22 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

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

which will ensure that

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

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

To get a (μ, L)-tensorial collection of walks, Lemma 3.9.20 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 3.6.17 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 3.6.17 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. ∎

3.9.3 *Instantiation to General Base Codes*

The proof of Theorem 3.9.2 follows from Lemma 3.9.23 in the same way that Theorem 3.7.7 follows from Lemma 3.7.5 in the case of HDXs.

CHAPTER 4

DECODING EXPLICIT ε-BALANCED CODES NEAR THE GILBERT–VARSHAMOV BOUND

4.1 Introduction

Binary error correcting codes have pervasive applications [Gur10, GRS19] and yet we are far from understanding some of their basic properties [Gur09]. For instance, until very recently no explicit binary code achieving distance $1/2 - \varepsilon/2$ with rate near $\Omega(\varepsilon^2)$ was known, even though the existence of such codes was (non-constructively) established long ago [Gil52, Var57] in what is now referred as the Gilbert–Varshamov (GV) bound. On the impossibility side, a rate upper bound of $O(\varepsilon^2 \log(1/\varepsilon))$ is known for binary codes of distance $1/2 - \varepsilon/2$ (e.g., [Del75, MRRW77, NS09]).

In a breakthrough result [TS17], Ta-Shma gave an explicit construction of binary codes achieving nearly optimal distance versus rate trade-off, namely, binary codes of distance $1/2 - \varepsilon/2$ with rate $\Omega(\varepsilon^{2+\beta})$ where β vanishes as ε vanishes [1]. Actually, Ta-Shma obtained ε-balanced binary linear codes, that is, linear binary codes with the additional property that non-zero codewords have Hamming weight bounded not only below by $1/2 - \varepsilon/2$ but also above by $1/2 + \varepsilon/2$, and this is a fundamental property in the study of pseudo-randomness [NN90, AGHP92].

While the codes constructed by Ta-Shma are explicit, they were not known to admit efficient decoding algorithms, while such results are known for codes with smaller rates. In particular, an explicit binary code due to Guruswami and Rudra [GR06] is known to be even list decodable at an error radius $1/2 - \varepsilon$ with rate $\Omega(\varepsilon^3)$. We consider the following question:

1. In fact, Ta-Shma obtained $\beta = \beta(\varepsilon) = \Theta(((\log\log 1/\varepsilon)/\log 1/\varepsilon)^{1/3})$ and thus $\lim_{\varepsilon \to 0} \beta(\varepsilon) = 0$.

Do explicit binary codes near the GV bound admit an efficient decoding algorithm?

Here, we answer this question in the affirmative by providing an efficient [2] unique decoding algorithm for (essentially) Ta-Shma's code construction, which we refer as Ta-Shma codes. More precisely, by building on Ta-Shma's construction and using our unique decoding algorithm we have the following result.

Theorem 4.1.1 (Unique Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$ where $\beta = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) a unique decoding algorithm with running time $N^{O_{\varepsilon,\beta}(1)}$.

Furthermore, if instead we take $\beta > 0$ to be an arbitrary constant, the running time becomes $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$ (fixed polynomial time).

We can also perform "gentle" list decoding in the following sense (note that this partially implies Theorem 5.1.1).

Theorem 4.1.2 (Gentle List Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$ where $\beta = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) a list decoding algorithm that decodes within radius $1/2 - 2^{-\Theta((\log_2(1/\varepsilon))^{1/6})}$ in time $N^{O_{\varepsilon,\beta}(1)}$.

2. By "efficient", we mean polynomial time. Given the fundamental nature of the problem of decoding nearly optimal binary codes, it is an interesting open problem to make these techniques viable in practice.

We observe that the exponent in the running time $N^{O_{\varepsilon,\beta}(1)}$ appearing in Theorem 5.1.1 and Theorem 5.1.2 depends on ε. This dependence is no worse than $O(\log\log(1/\varepsilon))$, and if $\beta > 0$ is taken to be an arbitrarily constant (independent of ε), the running time becomes $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$. Avoiding this dependence in the exponent when $\beta = \beta(\varepsilon)$ is an interesting open problem. Furthermore, obtaining a list decoding radius of $1/2 - \varepsilon/2$ in Theorem 5.1.2 with the same rate (or even $\Omega(\varepsilon^2)$) is another very interesting open problem and related to a central open question in the adversarial error regime [Gur09].

Direct sum codes. Our work can be viewed within the broader context of developing algorithms for the decoding of direct sum codes. Given a (say linear) code $\mathcal{C} \subseteq \mathbb{F}_2^n$ and a collection of tuples $W \subseteq [n]^t$, the code $\mathrm{dsum}_W(\mathcal{C})$ with block length $|W|$ is defined as

$$\mathrm{dsum}_W(\mathcal{C}) = \left\{ (z_{w_1} + z_{w_2} + \cdots + z_{w_t})_{w \in W} \mid z \in \mathcal{C} \right\}.$$

The direct sum operation has been used for several applications in coding and complexity theory [ABN+92, IW97, GI01, IKW09, DS14, DDG+15, Cha16, DK17, Aro02]. It is easy to see that if $\mathcal{C}$ is ε_0-balanced for a constant ε_0, then for any $\varepsilon > 0$, choosing W to be a random collection of tuples of size $O(n/\varepsilon^2)$ results in $\mathrm{dsum}_W(\mathcal{C})$ being an ε-balanced code. The challenge in trying to construct good codes using this approach is to find explicit constructions of (sparse) collections W which are "pseudorandom" enough to yield a similar distance amplification as above. On the other hand, the challenge in decoding such codes is to identify notions of "structure" in such collections W, which can be exploited by decoding algorithms.

In Ta-Shma's construction [TS17], such a pseudorandom collection W was constructed by considering an expanding graph G over the vertex set $[n]$, and generating t-tuples using sufficiently long walks of length $t-1$ over the so-called s-wide replacement product of G with another (small) expanding graph H. Roughly speaking, this graph product is a

generalization of the celebrated zig-zag product [RVW00] but with s different steps of the zig-zag product instead of a single one. Ta-Shma's construction can also be viewed as a clever way of selecting a *sub-collection* of all walks in G, which refines an earlier construction suggested by Rozenman and Wigderson [Bog12] (and also analyzed by Ta-Shma) using *all* walks of length $t-1$.

Identifying structures to facilitate decoding. For the closely related direct product construction (where the entry corresponding to $w \in W$ is the entire t-tuple $(z_{w_1}, \ldots, z_{w_t})$) which amplifies distance but increases the alphabet size, it was proved by Alon et al. [ABN+92] that the resulting code admits a unique decoding algorithm if the incidence graph corresponding to the collection W is a good sampler. Very recently, it was proved by Dinur et al. [DHK+19] that such a direct product construction admits list decoding if the incidence graph is a "double sampler". The results of [DHK+19] also apply to direct sum, but the use of double samplers pushes the rate away from near optimality.

For the case of direct sum codes, the decoding task can be phrased as a maximum t-XOR problem with the additional constraint that the solution must lie in $\mathcal{C}$. More precisely, given $\tilde{y} \in \mathbb{F}_2^W$ within the unique decoding radius of $\mathrm{dsum}_W(\mathcal{C})$, we consider the following optimization problem

$$\operatorname*{argmin}_{z \in \mathcal{C}} \ \Delta(\tilde{y}, \mathrm{dsum}_W(z)),$$

where $\Delta(\cdot,\cdot)$ is the (normalized) Hamming distance. While maximum t-XOR is in general hard to solve to even any non-trivial degree of approximation [Hås97], previous work by the authors [AJQ+20] identified a structural condition on W called "splittability" under which the above constraint satisfaction problem can be solved (approximately) resulting in efficient unique and list decoding algorithms. However, by itself the splittability condition is too crude to be applicable to codes such as the ones in Ta-Shma's

construction. The requirements it places on the expansion of G are too strong and the framework in [AJQ+20] is only able to obtain algorithms for direct sum codes with rate $2^{-(\log(1/\varepsilon))^2} \ll \varepsilon^{2+\beta}$.

The conceptual contribution of this work can be viewed as identifying a different recursive structure in direct sums generated by expander walks, which allows us to view the construction as giving a sequence of codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_\ell$. Here, $\mathcal{C}_0$ is the starting code $\mathcal{C}$ and $\mathcal{C}_\ell$ is the final desired code, and each element in the sequence can be viewed as being obtained via a direct sum operation on the preceding code. Instead of considering a "one-shot" decoding task of finding an element of $\mathcal{C}_0$, this facilitates an iterative approach where at each step we reduce the task of decoding the code $\mathcal{C}_i$ to decoding for $\mathcal{C}_{i-1}$, using the above framework from [AJQ+20]. Such an iterative approach with a sequence of codes was also used (in a very different setting) in a work of Guruswami and Indyk [GI03] constructing codes over a large alphabet which are list decodable in linear time via spectral algorithms.

Another simple and well-known (see e.g., [GI04]) observation, which is very helpful in our setting, is the use of list decoding algorithms for unique decoding. For a code with distance $1/2 - \varepsilon/2$, unique decoding can be obtained by list decoding at a much smaller error radius of (say) $1/2 - 1/8$. This permits a much more efficient application of the framework from [AJQ+20], with a milder dependence on the expansion of the graphs G and H in Ta-Shma's construction, resulting in higher rates. We give a more detailed overview of our approach in Section 4.3.

Known results for random ensembles. While the focus in this work is on explicit constructions, there are several known (non-explicit) constructions of random ensembles of binary codes near or achieving the Gilbert–Varshamov bound (e.g., Table 4.1). Although it is usually straightforward to ensure the desired rate in such constructions, the distance only holds with high probability. Given a sample code from such ensembles, certifying

the minimum distance is usually not known to be polynomial time in the block length. Derandomizing such constructions is also a possible avenue for obtaining optimal codes, although such results remain elusive to this date (to the best of our knowledge).

One of the simplest constructions is that of random binary linear codes in which the generator matrix is sampled uniformly. This random ensemble achieves the GV bound with high probability, but its decoding is believed to be computationally hard [MMT11].

Much progress has been made on binary codes by using results for larger alphabet codes [Gur09]. Codes over non-binary alphabets with optimal (or nearly optimal) parameters are available [vL99, Sti08, GR06] and thanks to this availability a popular approach to constructing binary codes has been to concatenate such large alphabet codes with binary ones. Thommesen [Tho83] showed that by concatenating Reed–Solomon (RS) codes with random binary codes (one random binary code for each position of the outer RS code) it is possible to achieve the GV bound. Note that Thommesen codes arise from a more structured ensemble than random binary linear codes. This additional structure enabled Guruswami and Indyk [GI04] to obtain efficient decoding algorithms for the non-explicit Thommesen codes (whose minimum distance is not known to admit efficient certification). This kind of concatenation starting from a large alphabet code and using random binary codes, which we refer as Thommesen-like, has been an important technique in tackling binary code constructions with a variety of properties near or at the GV bound. An important drawback in several such Thommesen-like code constructions is that they end up being non-explicit (unless efficient derandomization or brute-force is viable).

Using a Thommesen-like construction, Gopi et al. [GKO$^+$17] showed non-explicit constructions of locally testable and locally correctable binary codes approaching the GV bound. More recently, again with a Thommesen-like construction, Hemenway et al. [HRW17] obtained non-explicit near linear time unique decodable codes at the GV

bound improving the running time of Guruswami and Indyk [GI04] (and also the decoding rates). We summarize the results discussed so far in Table 4.1.

Binary Code Results near the Gilbert–Varshamov bound						
Who?	**Construction**	**GV**	**Explicit**	**Concatenated**	**Decoding**	**Local**
[Gil52, Var57]	existential	yes	no	no	no	n/a
[Tho83]	Reed–Solomon + random binary	yes	no	yes	no	n/a
[GI04]	Thommesen [Tho83]	yes	no	yes	unique decoding	n/a
[GKO$^+$17]	Thommesen-like	yes	no	yes	unique decoding	LTC/LCC
[HRW17]	Thommesen-like	yes	no	yes	near linear time unique decoding	n/a
[TS17]	Expander-based	$\Omega(\varepsilon^{2+\beta})$	yes	no	no	n/a
this paper	Ta-Shma [TS17]	$\Omega(\varepsilon^{2+\beta})$	yes	no	gentle list decoding	n/a

Table 4.1: GV bound related results for binary codes.

There are also non-explicit constructions known to achieve list decoding capacity [GR08, MRRZ$^+$19] (being concatenated or LDPC/Gallager [Gal62] is not an obstruction to achieve capacity). Contrary to the other results in this subsection, Guruswami and Rudra [Gur05, GR06, Gur09], also using a Thommesen-like construction, obtained explicit codes that are efficiently list decodable from radius $1/2 - \varepsilon$ with rate $\Omega(\varepsilon^3)$. This was done by concatenating the so-called folded Reed–Solomon codes with a derandomization of a binary ensemble of random codes.

Results for non-adversarial error models. All the results mentioned above are for the adversarial error model of Hamming [Ham50, Gur10]. In the setting of random corruptions (Shannon model), the situation seems to be better understood thanks to the seminal result on explicit polar codes of Arikan [Ari09]. More recently, in another breakthrough Guruswami et al. [GRY19] showed that polar codes can achieve almost linear time decoding with near optimal convergence to capacity for the binary symmetric channel. This result gives an explicit code construction achieving parameter trade-offs similar to Shannon's randomized construction [Sha48] while also admitting very efficient encoding and decoding. Explicit capacity-achieving constructions are also known for bounded memory

channels [SKS19] which restrict the power of the adversary and thus interpolate between the Shannon and Hamming models.

4.2 Preliminaries and Notation

4.2.1 Codes

We briefly recall some standard code terminology. Given $z, z' \in \mathbb{F}_2^n$, recall that the relative Hamming distance between z and z' is $\Delta(z, z') := \left|\{i \mid z_i \neq z'_i\}\right| / n$. A binary code is any subset $\mathcal{C} \subseteq \mathbb{F}_2^n$. 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 if $\mathcal{C}$ is a linear subspace of $\mathbb{F}_2^n$. The rate of $\mathcal{C}$ is $\log_2(|\mathcal{C}|)/n$.

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

Definition 4.2.1 (Bias)**.** *The* bias *of a word $z \in \mathbb{F}_2^n$ is defined as* $\mathrm{bias}(z) := \left|\mathbb{E}_{i \in [n]}(-1)^{z_i}\right|$*. The bias of a code $\mathcal{C}$ is the maximum bias of any non-zero codeword in $\mathcal{C}$.*

Definition 4.2.2 (ε-balanced Code)**.** *A binary code $\mathcal{C}$ is* ε-balanced *if* $\mathrm{bias}(z + z') \leq \varepsilon$ *for every pair of distinct $z, z' \in \mathcal{C}$.*

Remark 4.2.3. *For linear binary code $\mathcal{C}$, the condition* $\mathrm{bias}(\mathcal{C}) \leq \varepsilon$ *is equivalent to $\mathcal{C}$ being an ε-balanced code.*

4.2.2 Direct Sum Lifts

Starting from a code $\mathcal{C} \subseteq \mathbb{F}_2^n$, we amplify its distance by considering the *direct sum lifting* operation based on a collection $W(k) \subseteq [n]^k$. The direct sum lifting maps each codeword of $\mathcal{C}$ to a new word in $\mathbb{F}_2^{|W(k)|}$ by taking the k-XOR of its entries on each element of $W(k)$.

Definition 4.2.4 (Direct Sum Lifting)**.** *Let $W(k) \subseteq [n]^k$. For $z \in \mathbb{F}_2^n$, we define the* direct sum lifting *as* $\mathrm{dsum}_{W(k)}(z) = y$ *such that* $y_{\mathfrak{s}} = \sum_{i \in \mathfrak{s}} z_i$ *for all* $\mathfrak{s} \in W(k)$*. The direct sum lifting of a code* $\mathcal{C} \subseteq \mathbb{F}_2^n$ *is*

$$\mathrm{dsum}_{W(k)}(\mathcal{C}) = \{\mathrm{dsum}_{W(k)}(z) \mid z \in \mathcal{C}\}.$$

We will omit $W(k)$ from this notation when it is clear from context.

Remark 4.2.5. *We will be concerned with collections $W(k) \subseteq [n]^k$ arising from length-$(k-1)$ walks on expanding structures (mostly on the s-wide replacement product of two expander graphs).*

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 4.2.6 (Parity Sampler)**.** *A collection $W(k) \subseteq [n]^k$ is called an* $(\varepsilon_0, \varepsilon)$-parity sampler *if for all $z \in \mathbb{F}_2^n$ with* $\mathrm{bias}(z) \leq \varepsilon_0$*, we have* $\mathrm{bias}(\mathrm{dsum}_{W(k)}(z)) \leq \varepsilon$*.*

4.2.3 Linear Algebra Conventions

All vectors considered in this paper are taken to be column vectors, and are multiplied on the left with any matrices or operators acting on them. Consequently, given an indexed sequence of operators $\mathsf{G}_{k_1}, \ldots, \mathsf{G}_{k_2}$ (with $k_1 \leq k_2$) corresponding to steps k_1 through k_2 of a walk, we expand the product $\prod_{i=k_1}^{k_2} G_i$ as

$$\prod_{i=k_1}^{k_2} G_i \;:=\; G_{k_2} \cdots G_{k_1}\,.$$

Unless otherwise stated, all inner products for vectors in coordinate spaces are taken to be with respect to the (uniform) probability measure on the coordinates. Similarly, all inner products for functions are taken to be with respect to the uniform measure on the inputs. All operators considered in this paper are normalized to have singular values at most 1.

4.3 Proof Overview

The starting point for our work is the framework developed in [AJQ+20] for decoding direct sum codes, obtained by starting from a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ and considering all parities corresponding to a set of t-tuples $W(t) \subseteq [n]^t$. Ta-Shma's near optimal ε-balanced codes are constructed by starting from a code with constant rate and constant distance and considering such a direct sum lifting. The set of tuples $W(t)$ in his construction corresponds to a set of walks of length $t-1$ on the s-wide replacement product of an expanding graph G with vertex set $[n]$ and a smaller expanding graph H. The s-wide replacement product can be thought of here as a way of constructing a much smaller pseudorandom subset of the set of all walks of length $t-1$ on G, which yields a similar distance amplification for the lifted code.

The simplified construction with expander walks. While we analyze Ta-Shma's construction later in the paper, it is instructive to first consider a $W(t)$ simply consisting of all walks of length $t-1$ on an expander. This construction, based on a suggestion of Rozenman and Wigderson [Bog12], was also analyzed by Ta-Shma [TS17] and can be used to obtain ε-balanced codes with rate $\Omega(\varepsilon^{4+o(1)})$. It helps to illustrate many of the conceptual ideas involved in our proof, while avoiding some technical issues.

Let G be a d-regular expanding graph with vertex set $[n]$ and the (normalized) second singular value of the adjacency operator A_G being λ. Let $W(t) \subseteq [n]^t$ denote the set of t-tuples corresponding to all walks of length $t-1$, with $N = |W(t)| = n \cdot d^{t-1}$. Ta-Shma proves that for all $z \in \mathbb{F}_2^n$, $W(t)$ satisfies

$$\mathrm{bias}(z) \;\leq\; \varepsilon_0 \quad \Rightarrow \quad \mathrm{bias}(\mathrm{dsum}_{W(t)}(z)) \;\leq\; (\varepsilon_0 + 2\lambda)^{\lfloor (t-1)/2 \rfloor},$$

i.e., $W(t)$ is an $(\varepsilon_0, \varepsilon)$-*parity sampler* for $\varepsilon = (\varepsilon_0 + 2\lambda)^{\lfloor (t-1)/2 \rfloor}$. Choosing $\varepsilon_0 = 0.1$ and

$\lambda = 0.05$ (say), we can choose $d = O(1)$ and obtain the ε-balanced code $\mathcal{C}' = \mathrm{dsum}_{W(t)}(\mathcal{C})$ with rate $d^{-(t-1)} = \varepsilon^{O(1)}$ (although the right constants matter a lot for optimal rates).

Decoding as constraint satisfaction. The starting point for our work is the framework in [AJQ+20] which views the task of decoding $\tilde{y}$ with $\Delta(\mathcal{C}', \tilde{y}) < (1-\varepsilon)/4 - \delta$ (where the distance of $\mathcal{C}'$ is $(1-\varepsilon)/2$) as an instance of the MAX t-XOR problem (see Fig. 4.1). The goal is to find

$$\operatorname*{argmin}_{z \in \mathcal{C}} \Delta\left(\mathrm{dsum}_{W(t)}(z), \tilde{y}\right),$$

which can be rephrased as

$$\operatorname*{argmax}_{z \in \mathcal{C}} \mathop{\mathbb{E}}_{w=(i_1,\ldots,i_t) \in W(t)} \left[\mathbb{1}_{\{z_{i_1} + \cdots + z_{i_t} = \tilde{y}_w\}}\right].$$

It is possible to ignore the condition that $z \in \mathcal{C}$ if the collection $W(t)$ is a slightly stronger parity sampler. For any solution $\tilde{z} \in \mathbb{F}_2^n$ (not necessarily in $\mathcal{C}$) such that

$$\Delta(\mathrm{dsum}_{W(t)}(\tilde{z}), \tilde{y}) < \frac{1-\varepsilon}{4} + \delta,$$

we have

$$\Delta(\mathrm{dsum}_{W(t)}(\tilde{z}), \mathrm{dsum}_{W(t)}(z)) < \frac{1-\varepsilon}{2}$$

by the triangle inequality, and thus $\mathrm{bias}(\mathrm{dsum}_{W(t)}(z - \tilde{z})) > \varepsilon$. If $W(t)$ is not just an $(\varepsilon_0, \varepsilon)$-parity sampler, but in fact a $((1+\varepsilon_0)/2, \varepsilon)$-parity sampler, this would imply $\mathrm{bias}(z - \tilde{z}) > (1+\varepsilon_0)/2$. Thus, $\Delta(z, \tilde{z}) < (1-\varepsilon_0)/4$ (or $\Delta(z, \bar{\tilde{z}}) < (1-\varepsilon_0)/4$) and we can use a unique decoding algorithm for $\mathcal{C}$ to find z given $\tilde{z}$.

The task of finding such a $z \in \mathcal{C}$ boils down to finding a solution $\tilde{z} \in \mathbb{F}_2^n$ to a MAX t-XOR instance, up to a an additive loss of $O(\delta)$ in the fraction of constraints satisfied by the optimal solution. While this is hard to do in general [Hås01, Gri01], [AJQ+20]

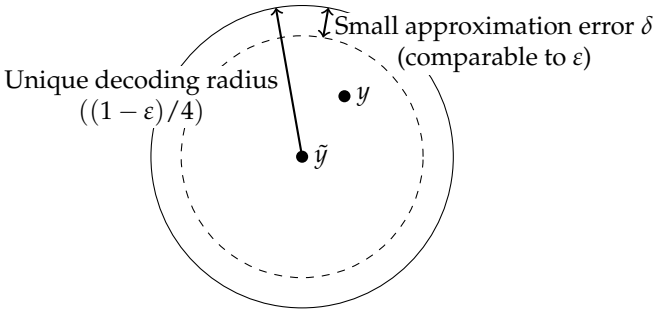

Figure 4.1: Unique decoding ball along with error from approximation.

(building on [AJT19]) show that this can be done if the instance satisfies a special property called *splittability*. To define this, we let $W[t_1, t_2] \subset [n]^{t_2-t_1+1}$ denote the collection of $(t_2 - t_1 + 1)$-tuples obtained by considering the indices between t_1 and t_2 for all tuples in $W(t)$. We also assume that all $w \in W[t_1, t_2]$ can be extended to the same number of tuples in $W(t)$ (which is true for walks).

Definition 4.3.1 (Splittability (informal))**.** *A collection $W(t) \subseteq [n]^t$ is said to be τ-splittable, if $t = 1$ (base case) or there exists $t' \in [t-1]$ such that:*

1. *The matrix $\mathsf{S} \in \mathbb{R}^{W[1,t'] \times W[t'+1,t]}$ defined by $\mathsf{S}(w, w') = \mathbb{1}_{\{ww' \in W\}}$ has normalized second singular value at most τ (where ww' denotes the concatenated tuple).*

2. *The collections $W[1, t']$ and $W[t'+1, t]$ are τ-splittable.*

For example, considering walks in G of length 3 ($t = 4$) and $t' = 2$, we get that $W[1,2] = W[3,4] = E$, the set of oriented edges in G. Also $\mathsf{S}(w, w') = 1$ if and only if the second vertex of w and first vertex of w' are adjacent in G. Thus, up to permutation of rows and columns, we can write the normalized version of S as $\mathsf{A}_G \otimes \mathsf{J}_d/d$ where A_G is normalized adjacency matrix of G and J_d denotes the $d \times d$ matrix of 1s. Hence such a $W(t)$ satisfies $\sigma_2(\mathsf{S}) \leq \tau$ with $\tau = \sigma_2(\mathsf{A}_G)$, and a similar proof works for walks of all lengths.

The framework in [AJQ+20] and [AJT19] gives that if $W(t)$ is τ-splittable for $\tau =$

$(\delta/2^t)^{O(1)}$, then the above instance of MAX t-XOR can be solved to additive error $O(\delta)$ using the Sum-of-Squares (SOS) SDP hierarchy. Broadly speaking, splittability allows one to (recursively) treat instances as expanding instances of problems with two "tuple variables" in each constraint, which can then be analyzed using known algorithms for 2-CSPs [BRS11, GS11] in the SOS hierarchy. Combined with parity sampling, this yields a unique decoding algorithm. Crucially, this framework can also be extended to perform *list decoding*[3] up to a radius of $1/2 - \sqrt{\varepsilon} - \delta$ under a similar condition on τ, which will be very useful for our application.

While the above can yield decoding algorithms for suitably expanding G, the requirement on τ (and hence on λ) makes the rate much worse. We need $\delta = O(\varepsilon)$ (for unique decoding) and $t = O(\log(1/\varepsilon))$ (for parity sampling), which requires $\lambda = \varepsilon^{\Omega(1)}$, yielding only a quasipolynomial rate for the code (recall that we could take $\lambda = O(1)$ earlier yielding polynomial rates).

Unique decoding: weakening the error requirement. We first observe that it is possible to get rid of the dependence $\delta = O(\varepsilon)$ above by using the *list decoding* algorithm for unique decoding. It suffices to take $\delta = 0.1$ and return the closest element from the the list of all codewords up to an error radius $1/2 - \sqrt{\varepsilon} - 0.1$, if we are promised that $\Delta(\tilde{y}, \mathcal{C})$ is within the unique decoding radius (see Fig. 4.2). However, this alone does not improve the rate as we still need the splittability (and hence λ) to be $2^{-\Omega(t)}$ with $t = O(\log(1/\varepsilon))$.

Code cascades: handling the dependence on walk length. To avoid the dependence of the expansion on the length $t - 1$ of the walk (and hence on ε), we avoid the "one-shot" decoding above, and instead consider a sequence of intermediate codes between $\mathcal{C}$ and

3. While unique decoding can be thought of as recovering a single solution to a constraint satisfaction problem, the goal in the list decoding setting can be thought of as obtaining a "sufficiently rich" set of solutions which forms a good cover. This is achieved in the framework by adding an entropic term to the semidefinite program, which ensures that the SDP solution satisfies such a covering property.

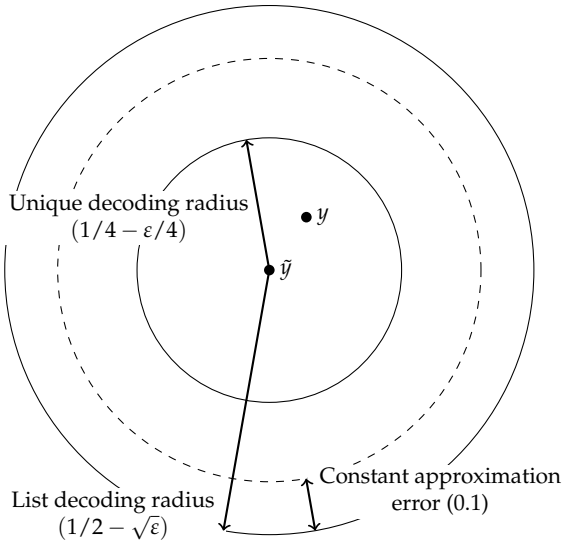

Figure 4.2: Unique decoding and list decoding balls along with error from approximation. Note that the list decoding ball contains the unique decoding ball even after allowing for a relatively large amount of error.

$\mathcal{C}'$. Consider the case when $t = k^2$, and instead of computing t-wise sums of bits in each $z \in \mathbb{F}_2^n$, we first compute k-wise sums according to walks of length $k-1$ on G, and then a k-wise sum of these values. In fact, the second sum can also be thought of as arising from a length $k-1$ walk on a different graph, with vertices corresponding to (directed) walks with k vertices in G, and edges connecting w and w' when the last vertex of w is connected to the first one in w' (this is similar to the matrix considered for defining splittability). We can thus think of a sequence of codes $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ with $\mathcal{C}_0 = \mathcal{C}$ and $\mathcal{C}_2 = \mathcal{C}'$, and both $\mathcal{C}_1$ and $\mathcal{C}_2$ being k-wise direct sums. More generally, when $t = k^\ell$ for an appropriate constant k we can think of a sequence $\mathcal{C} = \mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_\ell = \mathcal{C}'$, where each is an k-wise direct sum of the previous code, obtained via walks of length $k-1$ (hence k vertices) in an appropriate graph. We refer to such sequences (defined formally in Section 4.5) as *code cascades* (see Fig. 4.3).

Instead of applying the decoding framework above to directly reduce the decoding of a corrupted codeword from $\mathcal{C}'$ to the unique decoding problem in $\mathcal{C}$, we apply it at each level of a cascade, reducing the unique decoding problem in $\mathcal{C}_i$ to that in $\mathcal{C}_{i-1}$. If the direct

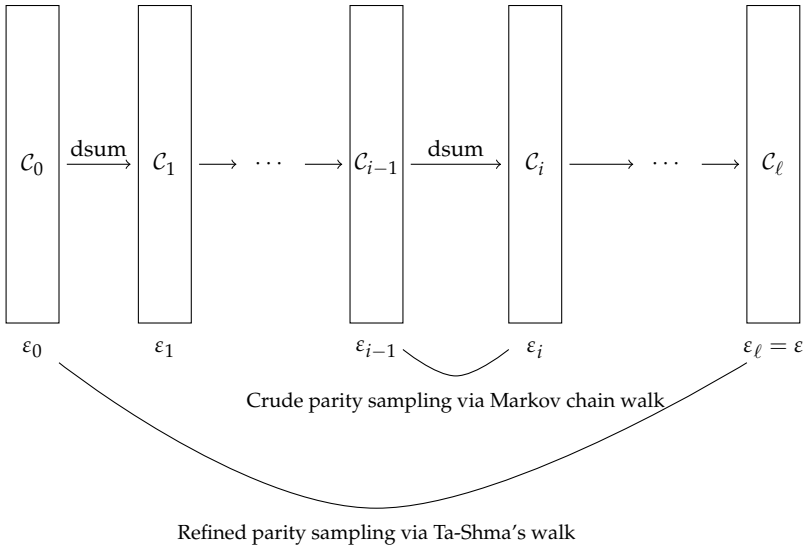

Figure 4.3: Code cascading.

sum at each level of the cascade is an (η_0, η)-parity sampler, the list decoding algorithm at radius $1/2 - \sqrt{\eta}$ suffices for unique decoding even if η is a (sufficiently small) constant independent of ε. This implies that we can take k to be a (suitably large) constant. This also allows the splittability (and hence λ) to be $2^{-O(k)} = \Omega(1)$, yielding polynomial rates. We present the reduction using cascades in Section 4.6 and the parameter choices in Section 4.8. The specific versions of the list decoding results from [AJQ+20] needed here are instantiated in Section 4.9.

While the above allows for polynomial rate, the *running time* of the algorithm is still exponential in the number of levels ℓ (which is $O(\log t) = O(\log\log(1/\varepsilon))$) since the list decoding for each level potentially produces a list of size $\mathrm{poly}(n)$, and recursively calls the decoding algorithm for the previous level on each element of the list. We obtain a fixed polynomial time algorithm by "pruning" the list at each level of the cascade before invoking the decoding algorithm for the previous level, while only slightly increasing the parity sampling requirements. The details are contained in Section 4.6.

Working with Ta-Shma's construction. Finally, to obtain near-optimal rates, we need to work with with Ta-Shma's construction, where the set of tuples $W(t) \subseteq [n]^t$ corresponds to walks arising from an s-wide replacement product of G with another expanding graph H. One issue that arises is that the collection of walks $W(t)$ as defined in [TS17] does not satisfy the important splittability condition required by our algorithms. However, this turns out to be easily fixable by modifying each step in Ta-Shma's construction to be exactly according to the zig-zag product of Reingold, Vadhan and Wigderson [RVW00]. We present Ta-Shma's construction and this modification in Section 4.4.

We also verify that the tuples given by Ta-Shma's construction satisfy the conditions for applying the list decoding framework, in Section 4.7. While the sketch above stated this in terms of splittability, the results in [AJQ+20] are in terms of a more technical condition called *tensoriality*. We show in Section 4.7 that this is indeed implied by splittability, and also prove splittability for (the modified version of) Ta-Shma's construction.

4.4 Ta-Shma's Construction: A Summary and Some Tweaks

In this section, we first discuss the s-wide replacement product that is central to Ta-Shma's construction of optimal ε-balanced codes, and then we describe the construction itself (we refer the reader to [TS17] for formal details beyond those we actually need here).

As mentioned before, we will also need to modify Ta-Shma's construction [TS17] a little to get *splittability* which is a notion of expansion of a collection $W(k) \subseteq [n]^k$ (and it is formally defined in Definition 4.7.9). The reason for this simple modification is that this *splittability* property is required by the list decoding framework. Note that we are not improving the Ta-Shma code parameters; in fact, we need to argue why with this modification we can still achieve Ta-Shma's parameters. Fortunately, this modification is simple enough that we will be able to essentially reuse Ta-Shma's original analysis.

In Appendix D.1.3, we will also have the opportunity to discuss, at an informal level, the intuition behind some parameter trade-offs in Ta-Shma codes which should provide enough motivation when we instantiate these codes in Section 4.8.

4.4.1 *The s-wide Replacement Product*

Ta-Shma's code construction is based on the so-called s-wide replacement product [TS17]. This is a derandomization of random walks on a graph G that will be defined via a product operation of G with another graph H (see Definition D.1.3 for a formal definition). We will refer to G as the *outer* graph and H as the *inner* graph in this construction.

Let G be a d_1-regular graph on vertex set $[n]$ and H be a d_2-regular graph on vertex set $[d_1]^s$, where s is any positive integer. Suppose the neighbors of each vertex of G are labeled $1, 2, \ldots, d_1$. For $v \in V(G)$, let $v_G[j]$ be the j-th neighbor of v. The s-wide replacement product is defined by replacing each vertex of G with a copy of H, called a "cloud". While the edges within each cloud are determined by H, the edges between clouds are based on the edges of G, which we will define via operators $\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_{s-1}$. The i-th operator G_i specifies one inter-cloud edge for each vertex $(v, (a_0, \ldots, a_{s-1})) \in V(G) \times V(H)$, which goes to the cloud whose G component is $v_G[a_i]$, the neighbor of v in G indexed by the i-th coordinate of the H component. (We will resolve the question of what happens to the H component after taking such a step momentarily.)

Walks on the s-wide replacement product consist of steps with two different parts: an intra-cloud part followed by an inter-cloud part. All of the intra-cloud substeps simply move to a random neighbor in the current cloud, which corresponds to applying the operator $\mathsf{I} \otimes \mathsf{A}_H$, where A_H is the normalized adjacency matrix of H. The inter-cloud substeps are all deterministic, with the first moving according to G_0, the second according to G_1, and so on, returning to G_0 for step number $s+1$. The operator for such a walk

taking $t-1$ steps on the s-wide replacement product is

$$\prod_{i=0}^{t-2} \mathsf{G}_{i \bmod s}(\mathsf{I} \otimes \mathsf{A}_H).$$

Observe that a walk on the s-wide replacement product yields a walk on the outer graph G by recording the G component after each step of the walk. The number of $(t-1)$-step walks on the s-wide replacement product is

$$|V(G)| \cdot |V(H)| \cdot d_2^{t-1} = n \cdot d_1^s \cdot d_2^{t-1},$$

since a walk is completely determined by its intra-cloud steps. If d_2 is much smaller than d_1 and t is large compared to s, this is less than nd_1^{t-1}, the number of $(t-1)$-step walks on G itself. Thus the s-wide replacement product will be used to simulate random walks on G while requiring a reduced amount of randomness (of course this simulation is only possible under special conditions, namely, when we are uniformly distributed on each cloud).

To formally define the s-wide replacement product, we must consider the labeling of neighbors in G more carefully.

Definition 4.4.1 (Rotation Map)**.** *Suppose G is a d_1-regular graph on $[n]$. For each $v \in [n]$ and $j \in [d_1]$, let $v_G[j]$ be the j-th neighbor of v in G. Based on the indexing of the neighbors of each vertex, we define the rotation map* [4] $\mathrm{rot}_G \colon [n] \times [d_1] \to [n] \times [d_1]$ *such that for every $(v, j) \in [n] \times [d_1]$,*

$$\mathrm{rot}_G((v, j)) = (v', j') \Leftrightarrow v_G[j] = v' \text{ and } v'_G[j'] = v.$$

4. This kind of map is denoted rotation map in the zig-zag terminology [RVW00].

Furthermore, if there exists a bijection $\varphi\colon [d_1] \to [d_1]$ *such that for every* $(v, j) \in [n] \times [d_1]$,

$$\mathrm{rot}_G((v, j)) = (v_G[j], \varphi(j)),$$

then we call rot_G locally invertible.

If G has a locally invertible rotation map, the cloud label after applying the rotation map only depends on the current cloud label, not the vertex of G. In the s-wide replacement product, this corresponds to the H component of the rotation map only depending on a vertex's H component, not its G component. We define the s-wide replacement product as described before, with the inter-cloud operator G_i using the i-th coordinate of the H component, which is a value in $[d_1]$, to determine the inter-cloud step.

Definition 4.4.2 (s-wide replacement product)**.** *Suppose we are given the following:*

- *A* d_1*-regular graph* $G = ([n], E)$ *together with a locally invertible rotation map* $\mathrm{rot}_G\colon [n] \times [d_1] \to [n] \times [d_1]$.
- *A* d_2*-regular graph* $H = ([d_1]^s, E')$.

And we define:

- *For* $i \in \{0, 1, \ldots, s-1\}$, *we define* $\mathrm{Rot}_i\colon [n] \times [d_1]^s \to [n] \times [d_1]^s$ *as, for every* $v \in [n]$ *and* $(a_0, \ldots, a_{s-1}) \in [d_1]^s$,

  $$\mathrm{Rot}_i((v, (a_0, \ldots, a_{s-1}))) \coloneqq (v', (a_0, \ldots, a_{i-1}, a_i', a_{i+1}, \ldots, a_{s-1})),$$

  where $(v', a_i') = \mathrm{rot}_G(v, a_i)$.
- *Denote by* G_i *the operator realizing* Rot_i *and let* A_H *be the normalized random walk operator of* H. *Note that* G_i *is a permutation operator corresponding to a product of transpositions.*

Then $t-1$ steps of the s-wide replacement product are given by the operator

$$\prod_{i=0}^{t-2} \mathsf{G}_{i \bmod s}(\mathsf{I} \otimes \mathsf{A}_H).$$

Ta-Shma instantiates the s-wide replacement product with an outer graph G that is a Cayley graph, for which locally invertible rotation maps exist generically.

Remark 4.4.3. *Let R be a group and $A \subseteq R$ where the set A is closed under inversion. For every Cayley graph $\mathrm{Cay}(R, A)$, the map $\varphi\colon A \to A$ defined as $\varphi(g) = g^{-1}$ gives rise to the locally invertible rotation map*

$$\mathrm{rot}_{\mathrm{Cay}(R,A)}((r, a)) = (r \cdot a, a^{-1}),$$

for every $r \in R$, $a \in A$.

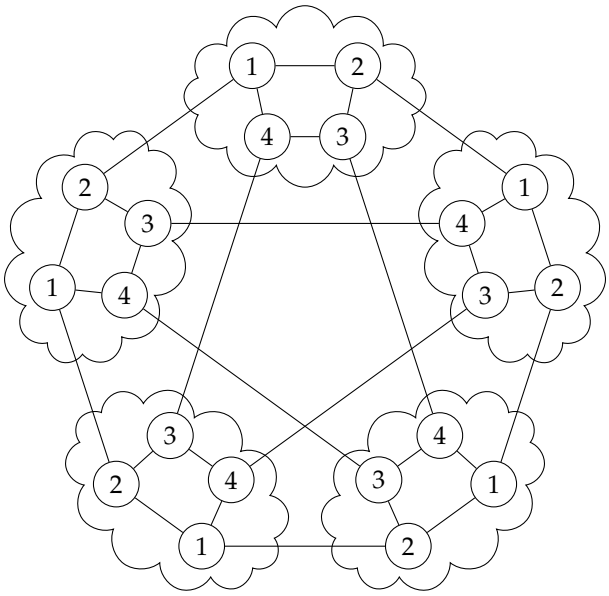

Figure 4.4: An example of the 1-wide replacement product with outer graph $G = K_5$ and inner graph $H = C_4$. Vertices are labeled by their H components. Note that the rotation map is locally invertible, with $\varphi(1) = 2$, $\varphi(2) = 1$, $\varphi(3) = 4$, and $\varphi(4) = 3$.

4.4.2 The Construction

Ta-Shma's code construction works by starting with a constant bias code $\mathcal{C}_0$ in $\mathbb{F}_2^n$ and boosting to arbitrarily small bias using direct sum liftings. Recall that the direct sum lifting is based on a collection $W(t) \subseteq [n]^t$, which Ta-Shma obtains using $t-1$ steps of random walk on the s-wide replacement product of two regular expander graphs G and H. The graph G is on n vertices (same as blocklength of the base code) and other parameters like degrees d_1 and d_2 of G and H respectively are chosen based on target code parameters.

To elaborate, every $t-1$ length walk on the replacement product gives a sequence of t outer vertices or G-vertices, which can be seen as an element of $[n]^t$. This gives the collection $W(t)$ with $|W(t)| = n \cdot d_1^s \cdot d_2^{t-1}$ which means the rate of lifted code is smaller than the rate of $\mathcal{C}_0$ by a factor of $d_1^s d_2^{t-1}$. However, the collection $W(t)$ is a parity sampler and this means that the bias decreases (or the distance increases). The relationship between this decrease in bias and decrease in rate with some careful parameter choices allows Ta-Shma to obtain nearly optimal ε-balanced codes.

4.4.3 Tweaking the Construction

Recall the first s steps in Ta-Shma's construction are given by the operator

$$\mathsf{G}_{s-1}(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{s-2} \cdots G_1(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_0(\mathsf{I} \otimes \mathsf{A}_H).$$

Naively decomposing the above operator into the product of operators $\prod_{i=0}^{s-1} \mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ is not good enough to obtain the *splittability* property which would hold provided $\sigma_2(\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H))$ was small for every i in $\{0, \ldots, s-1\}$. However, each $\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ has $|V(G)|$ singular values equal to 1 since G_i is an orthogonal operator and $(\mathsf{I} \otimes \mathsf{A}_H)$ has $|V(G)|$ singular

values equal to 1. To avoid this issue we will tweak the construction to be the following product

$$\prod_{i=0}^{s-1}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H).$$

The operator $(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)$ is exactly the walk operator of the zig-zag product $G\textcircled{z}H$ of G and H with a rotation map given by the (rotation map) operator G_i. This tweaked construction is slightly simpler in the sense that $G\textcircled{z}H$ is an undirected graph. We know by the zig-zag analysis that $(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)$ is expanding as long G and H are themselves expanders. More precisely, we have a bound that follows from [RVW00].

Fact 4.4.4. *Let G be an outer graph and H be an inner graph used in the s-wide replacement product. For any integer $0 \leq i \leq s-1$,*

$$\sigma_2((\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)) \leq \sigma_2(G) + 2\cdot\sigma_2(H) + \sigma_2(H)^2.$$

This bound will imply *splittability* as shown in Appendix D.1.4. We will need to argue that this modification still preserves the correctness of the parity sampling and that it can be achieved with similar parameter trade-offs.

The formal definition of a length-t walk on this slightly modified construction is given below.

Definition 4.4.5. *Let $t \in \mathbb{N}$, G be a d_1-regular graph and H be a d_2-regular graph on d_1^s vertices. Given a starting vertex $(v,h) \in V(G)\times V(H)$, a $(t-1)$-step walk on the tweaked s-wide replacement product of G and H is a tuple $((v_0,h_0),\ldots,(v_{t-1},h_{t-1})) \in (V(G)\times V(H))^t$ such that*

- *$(v_0,h_0) = (v,h)$, and*
- *for every $0 \leq i < t-1$, we have (v_i,h_i) adjacent to (v_{i+1},h_{i+1}) in $(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_{i \bmod s}(\mathsf{I}\otimes\mathsf{A}_H)$.*

Note that each $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{i \bmod s}(\mathsf{I} \otimes \mathsf{A}_H)$ *is a walk operator of a* d_2^2*-regular graph. Therefore, the starting vertex* (v, h) *together with a degree sequence* $(m_1, \ldots, m_t) \in [d_2^2]^{t-1}$ *uniquely defines a* $(t-1)$*-step walk.*

Parity Sampling

We argue informally why parity sampling still holds with similar parameter trade-offs. Later in Section 4.4.3, we formalize a key result underlying parity sampling and, in Section 4.8, we compute the new trade-off between bias and rate in some regimes. In Appendix D.1.1, the definition of the original s-wide replacement product as a purely graph theoretic operation was given. Now, we explain how Ta-Shma used this construction for parity sampling obtaining codes near the GV bound.

For a word $z \in \mathbb{F}_2^{V(G)}$ in the base code, let P_z be the diagonal matrix, whose rows and columns are indexed by $V(G) \times V(H)$, with $(\mathsf{P}_z)_{(v,h),(v,h)} = (-1)^{z_v}$. Proving parity sampling requires analyzing the operator norm of the following product

$$\mathsf{P}_z \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H), \tag{4.1}$$

when $\mathrm{bias}(z) \leq \varepsilon_0$. Let $\mathbf{1} \in \mathbb{R}^{V(G)\times V(H)}$ be the all-ones vector and W be the collection of all $(t-1)$-step walks on the tweaked s-wide replacement product. Ta-Shma showed (and it is not difficult to verify) that

$$\mathrm{bias}\left(\mathrm{dsum}_W(z)\right) = \left| \left\langle \mathbf{1}, \mathsf{P}_z \prod_{i=0}^{t-2} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{i \bmod s}\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H)\mathbf{1} \right\rangle \right|.$$

From the previous equation, one readily deduces that

$$\mathrm{bias}\left(\mathrm{dsum}_W(z)\right) \leq \sigma_1 \left(\mathsf{P}_z \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H) \right)^{\lfloor (t-1)/s \rfloor}.$$

Set $\mathsf{B} := \mathsf{P}_z \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H)$. To analyze the operator norm of B, we will first need some notation. Note that B is an operator acting on the space $\mathcal{V} = \mathbb{R}^{V(G)} \otimes \mathbb{R}^{V(H)}$. Two of its subspaces play an important role in the analysis, namely,

$$\mathcal{W}^{\|} = \text{span}\{a \otimes b \in \mathbb{R}^{V(G)} \otimes \mathbb{R}^{V(H)} \mid b = \mathbf{1}\} \text{ and } \mathcal{W}^{\perp} = (\mathcal{W}^{\|})^{\perp}.$$

Note that the complement subspace is with respect to the standard inner product. Observe that $\mathcal{V} = \mathcal{W}^{\|} \oplus \mathcal{W}^{\perp}$. Given arbitrary unit vectors $v, w \in \mathcal{V}$, Ta-Shma considers the inner product

$$\left\langle v, \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H)w \right\rangle. \tag{4.2}$$

Each time an operator $(\mathsf{I} \otimes \mathsf{A}_H)$ appears in the above expression, the next step of the walk can take one out of d_2 possibilities and thus the rate suffers a multiplicative decrease of $1/d_2$. We think that we are "paying" d_2 for this step of the walk. The whole problem lies in the trade-off between rate and distance, so the crucial question now is how much the norm decreases as we pay d_2. For a moment, suppose that the norm always decreases by a factor of $\lambda_2 := \sigma_2(H)$ per occurrence of $(\mathsf{I} \otimes \mathsf{A}_H)$. If in this hypothetical case we could further assume $\lambda_2 = 1/\sqrt{d_2}$, then if B was a product containing $\lceil \log_{\lambda_2}(\varepsilon) \rceil$ factors of $(\mathsf{I} \otimes \mathsf{A}_H)$, the final bias would be at most ε and the rate would have suffered a multiplicative decrease of (essentially) ε^2 and we would be done.

Of course, this was an oversimplification. The general strategy is roughly the above, but a beautiful non-trivial step is needed. Going back to the bilinear form Eq. (4.2), if $w \in \mathcal{W}^{\perp}$ (or $v \in \mathcal{W}^{\perp}$), we pay d_2 and we do obtain a norm decrease of λ_2. More generally, note that can decompose $w = w^{\|} + w^{\perp}$ with $w^{\|} \in \mathcal{W}^{\|}$ and $w^{\perp} \in \mathcal{W}^{\perp}$ (decompose $v = v^{\|} + v^{\perp}$ similarly) and we can carry this process iteratively collecting factors of λ_2.

However, we are stuck with several terms of the form for $0 \le k_1 \le k_2 < s$,

$$\left\langle v_{k_1}^{\|}, \prod_{i=k_1}^{k_2} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H) w_{k_2}^{\|} \right\rangle,$$

with $v_{k_1}^{\|}, w_{k_2}^{\|} \in \mathcal{W}^{\|}$, and for which the preceding naive norm decrease argument fails. This is the point in the analysis where the structure of the s-wide replacement product is used. Since $v_{k_1}^{\|}, w_{k_2}^{\|} \in \mathcal{W}^{\|}$, these vectors are uniform on each "cloud", i.e., copy of H. Recall that a vertex in H is an s-tuple $(m_1, \ldots, m_s) \in [d_1]^s$. Ta-Shma leverages the fact of having a uniform such tuple to implement $k_2 - k_1 + 1$ (up to s) steps of random walk on G. More precisely, Ta-Shma obtains the following beautiful result:

Theorem 4.4.6 (Adapted from Ta-Shma [TS17])**.** *Let G be a locally invertible graph of degree d_1, H be a Cayley graph on $\mathbb{F}_2^{s \log d_1}$, and $0 \le k_1 \le k_2 < s$ be integers. If $v^{\|} = v \otimes 1$ and $w^{\|} = w \otimes 1$, then*

$$\left\langle v^{\|}, \prod_{i=k_1}^{k_2} \mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{P}_z w^{\|} \right\rangle = \left\langle v, (\mathsf{A}_G\mathsf{M}_z)^{k_2-k_1+1}\, w \right\rangle$$

where $\mathsf{M}_z \in \mathbb{R}^{V(G)\times V(G)}$ is the diagonal matrix defined as $(\mathsf{M}_z)_{v,v} := (-1)^{z_v}$ for $v \in V(G)$.

Remark 4.4.7. *Note that the walk operator in this theorem corresponds to the original construction. Theorem 4.4.6 was used by Ta-Shma to obtain Fact D.1.7 whose Corollary D.1.8 corresponds to the modified construction.*

Ta-Shma proved Theorem 4.4.6 under the more general condition that H is 0-pseudorandom. Roughly speaking, this property means that if we start with a distribution that is uniform over the clouds, and walk according to fixed H-steps $j_0, j_1, \cdots, j_{s-1}$, then the distribution of G-vertices obtained will be identical to the distribution obtained if we were doing the usual random walk on G. We will always choose H to be a Cayley graph on $\mathbb{F}_2^{s \log d_1}$,

which will imply that H is also 0-pseudorandom. The proof of Theorem 4.4.6 crucially uses the product structure of $\mathbb{F}_2^{s \log d_1}$: every vertex of H can be represented by s registers of $\log d_1$ bits each, and both inter-cloud and intra-cloud steps can be seen as applying register-wise bijections using some canonical mapping between $[d_1]$ and $\mathbb{F}_2^{\log d_1}$.

Ta-Shma's original parity sampling proof required $\varepsilon_0 + 2\theta + 2\sigma_2(G) \leq \sigma_2(H)^2$, where ε_0 is the initial bias and θ is an error parameter arising from a number theoretic construction of Ramanujan graphs for the outer graph G. This is because $\varepsilon_0 + 2\theta + 2\sigma_2(G)$ is the reduction of bias in every two steps while taking a walk on G (see Theorem 4.5.2). Having $\varepsilon_0 + 2\theta + 2\sigma_2(G) \leq \sigma_2(H)^2$ ensured that after establishing Theorem 4.4.6, we were collecting enough reduction for d_2^2 price we paid for two steps. In the modified construction, we now have d_2^2 possibilities for each step in $(\mathsf{I} \otimes \mathsf{A}_H^2)$ (so d_2^4 price for two steps), and so if instead we have $\varepsilon_0 + 2\theta + 2\sigma_2(G) \leq \sigma_2(H)^4$ in the modified construction, we claim that the correctness of the parity sampling analysis is preserved as well as (essentially) the trade-off between walk length and norm decay. Fortunately, Ta-Shma's parameters decouple and we can choose parameters to satisfy the above requirement.

Remark 4.4.8. *This modification on the s-replacement product of G and H essentially* [5] *amounts to taking a different inner graph H which can be factored as $H = \sqrt{H}\sqrt{H}$ (and is still 0-pseudorandom).*

Spectral Analysis of the Modified Construction

We formally show that we don't loose much by going from Ta-Shma's original s-wide product construction to its tweaked version. The key technical result obtained by Ta-Shma is the following, which is used to analyze the bias reduction as a function of the total number walk steps $t - 1$.

5. Except at the first and last factors in the product of operators.

Fact 4.4.9 (Theorem 24 abridged [TS17]). *If H is a Cayley graph on $\mathbb{F}_2^{s \log d_1}$ and $\varepsilon_0 + 2 \cdot \theta + 2 \cdot \sigma_2(G) \leq \sigma_2(H)^2$, then*

$$\left\| \prod_{i=0}^{s-1} \mathsf{P}_z \mathsf{G}_i (\mathsf{I} \otimes \mathsf{A}_H) \right\|_{\mathrm{op}} \leq \sigma_2(H)^s + s \cdot \sigma_2(H)^{s-1} + s^2 \cdot \sigma_2(H)^{s-3},$$

where $\mathsf{P}_z \in \mathbb{R}^{(V(G) \times V(H)) \times (V(G) \times V(H))}$ is the sign operator *of a ε_0 biased word $z \in \mathbb{F}_2^{V(G)}$ defined as a diagonal matrix with $(P_z)_{(v,h),(v,h)} = (-1)^{z_v}$ for every $(v, h) \in V(G) \times V(H)$.*

We reduce the analysis of Ta-Shma's tweaked construction to Fact D.1.7. In doing so, we only lose one extra step as shown below.

Corollary 4.4.10. *If H^2 is a Cayley graph on $\mathbb{F}_2^{s \log d_1}$ and $\varepsilon_0 + 2 \cdot \theta + 2 \cdot \sigma_2(G) \leq \sigma_2(H)^4$, then*

$$\left\| \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H) \mathsf{P}_z \mathsf{G}_i (\mathsf{I} \otimes \mathsf{A}_H) \right\|_{\mathrm{op}} \leq \sigma_2(H^2)^{s-1} + (s-1) \cdot \sigma_2(H^2)^{s-2} + (s-1)^2 \cdot \sigma_2(H^2)^{s-4},$$

where P_z is the sign operator *of an ε_0-biased word $z \in \mathbb{F}_2^{V(G)}$ as in Fact D.1.7.*

Proof. We have

$$\begin{aligned}
\left\| \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H) \mathsf{P}_z \mathsf{G}_i (\mathsf{I} \otimes \mathsf{A}_H) \right\|_{\mathrm{op}} &\leq \|(\mathsf{I} \otimes \mathsf{A}_H)\|_{\mathrm{op}} \left\| \prod_{i=1}^{s-1} \mathsf{P}_z \mathsf{G}_i (\mathsf{I} \otimes \mathsf{A}_H^2) \right\|_{\mathrm{op}} \|\mathsf{P}_z \mathsf{G}_0 (\mathsf{I} \otimes \mathsf{A}_H)\|_{\mathrm{op}} \\
&\leq \left\| \prod_{i=1}^{s-1} \mathsf{P}_z \mathsf{G}_i (\mathsf{I} \otimes \mathsf{A}_H^2) \right\|_{\mathrm{op}} \\
&\leq \sigma_2(H^2)^{s-1} + (s-1) \cdot \sigma_2(H^2)^{s-2} + (s-1)^2 \cdot \sigma_2(H^2)^{s-4},
\end{aligned}$$

where the last inequality follows from Fact D.1.7. ∎

Remark 4.4.11. *We know that in the modified construction H^2 is a Cayley graph since H is a Cayley graph.*

From this point onward, we will be working exclusively with the modified construction instead of using it in its original form. Any references to Ta-Shma's construction or the s-wide replacement product will actually refer to the modified versions described in this section.

4.5 Code Cascading

A code cascade is a sequence of codes generated by starting with a base code $\mathcal{C}_0$ and recursively applying lifting operations.

Definition 4.5.1. *We say that a sequence of codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_\ell$ is a* code cascade *provided $\mathcal{C}_i = \mathrm{dsum}_{W_i(t_i)}(\mathcal{C}_{i-1})$ for every $i \in [\ell]$. Each $W_i(t_i)$ is a subset of $[n_{i-1}]^{t_i}$, where $n_{i-1} = |W_{i-1}(t_{i-1})|$ is the block length of the code $\mathcal{C}_{i-1}$.*

Let us see how code cascades may be useful for decoding. Suppose we wish to lift the code $\mathcal{C}_0$ to $\mathcal{C}_\ell$, and there is some $W(t) \subseteq [n_0]^t$ such that $\mathcal{C}_\ell = \mathrm{dsum}_{W(t)}(\mathcal{C}_0)$. In our case of bias boosting, this t will depend on the target bias ε. However, the expansion requirement of the list-decoding framework of [AJQ+20] has a poor dependence on t. A way to work around this issue is to go from $\mathcal{C}_0$ to $\mathcal{C}_\ell$ via a code cascade as above such that each t_i is a constant independent of the final bias but $\prod_{i=1}^{\ell} t_i = t$ (which means ℓ depends on ε). The final code $\mathcal{C}_\ell$ of the cascade is the same as the code obtained from length-$(t-1)$ walks. While decoding will now become an ℓ-level recursive procedure, the gain from replacing t by t_i will outweigh this loss, as we discuss below.

4.5.1 Warm-up: Code Cascading Expander Walks

We now describe the code cascading construction and unique decoding algorithm in more detail. Let $G = (V, E)$ be a d-regular graph with uniform distribution over the edges. Let

m be a sufficiently large positive integer, which will be the number of vertices of the walks used for the lifting between consecutive codes in the cascade. At first, it will be crucial that we can take $m = O(1)$ so that the triangle inequality arising from the analysis of the lifting between two consecutive codes involves a constant number of terms. We construct a recursive family of codes as follows.

- Start with a code $\mathcal{C}_0$ which is linear and has constant bias ε_0.
- Define the code $\mathcal{C}_1 = \mathrm{dsum}_{W(m)}(\mathcal{C}_0)$, which is the direct sum lifting over the collection $W(m)$ of all length-$(m-1)$ walks on G using the code $\mathcal{C}_0$.
- Let $\widehat{G}_i = (V_i, E_i)$ be the (directed) graph where V_i is the collection of all walks on m^i vertices on G with two walks $(v_1, \ldots, v_{m^i})$ and $(u_1, \ldots, u_{m^i})$ connected iff v_{m^i} is adjacent to u_1 in G.
- Define $\mathcal{C}_i$ to be the direct sum lifting on the collection $W_i(m)$ of all length-$(m-1)$ walks on G_{i-1} using the code $\mathcal{C}_{i-1}$, i.e., $\mathcal{C}_i = \mathrm{dsum}_{W_i(m)}(\mathcal{C}_{i-1})$.
- Repeat this process to yield a code cascade $\mathcal{C}_0, \ldots, \mathcal{C}_\ell$.

Thanks to the definition of the graphs $\widehat{G}_i$ and the recursive nature of the construction, the final code $\mathcal{C}_\ell$ is the same as the code obtained from $\mathcal{C}_0$ by taking the direct sum lifting over all walks on $t = m^\ell$ vertices of G. We can use Ta-Shma's analysis (building on the ideas of Rozenman and Wigderson [Bog12]) for the simpler setting of walks over a single expander graph to determine the amplification in bias that occurs in going from $\mathcal{C}_0$ all the way to $\mathcal{C}_\ell$.

Theorem 4.5.2 (Adapted from Ta-Shma [TS17])**.** *Let $\mathcal{C}$ be an ε_0-balanced linear code, and let $\mathcal{C}' = \mathrm{dsum}_{W(t)}(\mathcal{C})$ be the direct sum lifting of $\mathcal{C}$ over the collection of all length-$(t-1)$ walks*

$W(t)$ on a graph G. Then

$$\text{bias}(\mathcal{C}') \leq (\varepsilon_0 + 2\sigma_2(G))^{\lfloor (t-1)/2 \rfloor}.$$

If $\sigma_2(G) \leq \varepsilon_0/2$ and $\ell = \left\lceil \log_m(2\log_{2\varepsilon_0}(\varepsilon) + 3) \right\rceil$, taking $t = m^\ell \geq 2\log_{2\varepsilon_0}(\varepsilon) + 3$ in the above theorem shows that the final code $\mathcal{C}_\ell$ is ε-balanced. Observe that the required expansion of the graph G only depends on the constant initial bias ε_0, not on the desired final bias ε. It will be important for being able to decode with better parameters that both $\sigma_2(G)$ and m are constant with respect to ε; only ℓ depends on the final bias (with more care we can make $\sigma_2(G)$ depend on ε, but we restrict this analysis to Ta-Shma's refined construction on the s-wide replacement product).

As mentioned before, to uniquely decode $\mathcal{C}_\ell$ we will inductively employ the list decoding machinery for expander walks from [AJQ+20]. The list decoding algorithm can decode a direct sum lifting $\mathcal{C}' = \text{dsum}_{W(m)}(\mathcal{C})$ as long as the graph G is sufficiently expanding, the walk length $m-1$ is large enough, and the base code $\mathcal{C}$ has an efficient unique decoding algorithm (see Theorem 4.6.1 for details).

The expansion requirement ultimately depends on the desired list decoding radius of $\mathcal{C}'$, or more specifically, on how close the list decoding radius is to $1/2$. If the distance of $\mathcal{C}'$ is at most $1/2$, its unique decoding radius is at most $1/4$, which means list decoding at the unique decoding radius is at a constant difference from $1/2$ and thus places only a constant requirement on the expansion of G. In the case of the code cascade $\mathcal{C}_i = \text{dsum}_{W_i(m)}(\mathcal{C}_{i-1})$, unique decoding of $\mathcal{C}_{i-1}$ is guaranteed by the induction hypothesis. It is not too difficult to see that each graph $\widehat{G}_i$ will have the same second singular value as G, so we can uniquely decode $\mathcal{C}_i$ if G meets the constant expansion requirement and m is sufficiently large.

4.5.2 Code Cascading Ta-Shma's Construction

We will now describe how to set up a code cascade based on walks on an s-wide replacement product. Consider the s-wide replacement product of the outer graph G with the inner graph H. The first s walk steps are given by the walk operator

$$\prod_{i=0}^{s-1}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H).$$

Let $\mathsf{A}_{s-1} \coloneqq (\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_{s-2}(\mathsf{I}\otimes\mathsf{A}_H)\cdots(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_0(\mathsf{I}\otimes\mathsf{A}_H)$. If the total walk length $t-1$ is a multiple of s, the walks are generated using the operator

$$((\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_{s-1}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{A}_{s-1})^{(t-1)/s}.$$

Here $(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_{s-1}(\mathsf{I}\otimes\mathsf{A}_H)$ is used as a "binding" operator to connect two walks containing s vertices at level $\mathcal{C}_2$, s^2 vertices at level $\mathcal{C}_3$, and so on. More precisely, we form the following code cascade.

- $\mathcal{C}_0$ is an ε_0-balanced linear code efficiently uniquely decodable from a constant radius.

- $\mathcal{C}_1 = \mathrm{dsum}_{W_1(s)}(\mathcal{C}_0)$, where $W_1(s)$ is the set of length-(s-1) walks given by the operator

$$\underbrace{(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_{s-2}(\mathsf{I}\otimes\mathsf{A}_H)}_{(s-2)\text{th step}}\cdots\underbrace{(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_0(\mathsf{I}\otimes\mathsf{A}_H)}_{0\text{th step}}.$$

- $\mathcal{C}_2 = \mathrm{dsum}_{W_2(s)}(\mathcal{C}_1)$, where $W_2(s)$ is the set of length-$(s-1)$ walks over the vertex set $W_1(s)$ (with the latter being the set of length-$(s-1)$ walks on the replacement product graph as mentioned above).

- $\mathcal{C}_{i+1} = \mathrm{dsum}_{W_{i+1}(s)}(\mathcal{C}_i)$, where $W_{i+1}(s)$ is the set of length-$(s-1)$ walks [6] over the vertex set $W_i(s)$. Similarly to the cascade of expander walks above, the lift can be thought of as being realized by taking walks using a suitable operator analogous to $\widehat{G}_i$. Since its description is more technical we postpone its definition (see Definition 4.7.2) to Appendix D.1.4 where it is actually used.

- $\mathcal{C}_\ell$ denotes the final code in the sequence, which will later be chosen so that its bias is at most ε.

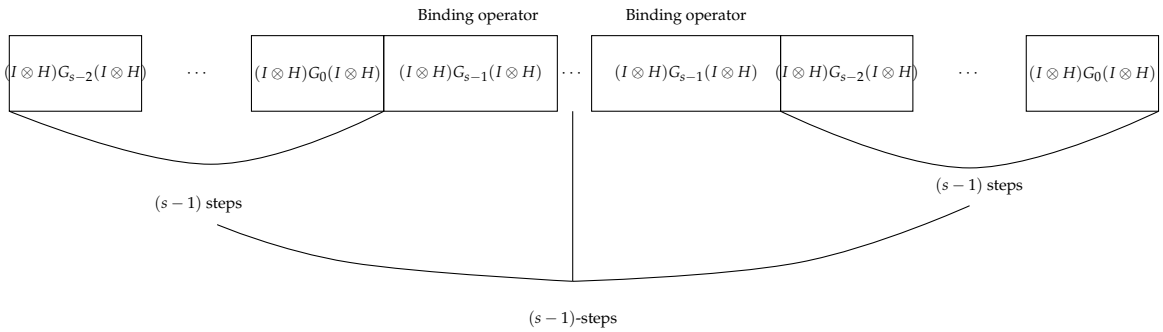

Figure 4.5: Two levels of code cascading for Ta-Shma's construction involving codes $\mathcal{C}_0$, $\mathcal{C}_1$ and $\mathcal{C}_2$ (to make the notation compact we used H to denote A_H).

4.6 Unique Decoding of Ta-Shma Codes

We show how code cascading together with list decoding for each level of the cascade allow us to obtain an efficient unique decoding algorithm for Ta-Shma's construction. We obtain a sequence of results of increasing strength culminating in Theorem 5.1.1 (which we recall below for convenience). The approach is as follows: we use several different instantiations of Ta-Shma's construction, each yielding a value of s (for the s-wide replacement product) and expansion parameters for the family of outer and inner graphs,

6. For simplicity we chose the number of vertices in all walks of the cascade to be s, but it could naturally be some $s_i \in \mathbb{N}$ depending on i.

and show how the list decoding framework can be invoked in the associated cascade for each one.

Theorem 5.1.1 (Unique Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$ where $\beta = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) a unique decoding algorithm with running time $N^{O_{\varepsilon,\beta}(1)}$.

Furthermore, if instead we take $\beta > 0$ to be an arbitrary constant, the running time becomes $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$ (fixed polynomial time).

In this section, we will fit these objects and tools together assuming the parameters are chosen to achieve the required rates and the conditions for applying the list decoding results are satisfied. The concrete instantiations of Ta-Shma codes are done in Section 4.8. Establishing that the list decoding framework can be applied to this construction is done in Section 4.7 after which the framework is finally instantiated in Section 4.9.

Ta-Shma uses the direct sum lifting on an s-wide replacement product graph to construct a family of ε-balanced codes $\mathcal{C}_{N,\varepsilon,\beta}$ with rate $\Omega(\varepsilon^{2+\beta})$ and finds parameters for such codes to have the required bias and rate. We will discuss unique decoding results for several versions of these codes. Throughout this section, we will use collections $W(k)$ which will always be either the set of walks with $k = s$ vertices on an s-wide replacement product graph (corresponding to the first level of the code cascade), which we denote $W[0, s-1]$, or a set of walks where the vertices are walks on a lower level of the code cascade.

4.6.1 Unique Decoding via Code Cascading

To perform unique decoding we will use the machinery of list decoding from Theorem 4.6.1 (proven later in Section 4.9), which relies on the list decoding framework of [AJQ+20]. Proving that this framework can be applied to Ta-Shma's construction is one of our technical contributions.

Theorem 4.6.1 (List decoding direct sum lifting)**.** *Let $\eta_0 \in (0, 1/4)$ be a constant, $\eta \in (0, \eta_0)$, and*

$$k \geq k_0(\eta) := \Theta(\log(1/\eta)).$$

Suppose $\mathcal{C} \subseteq \mathbb{F}_2^n$ is an η_0-balanced linear code and $\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ is the direct sum lifting of $\mathcal{C}$ on a τ-splittable collection of walks $W(k)$. There exists an absolute constant $K > 0$ such that if

$$\tau \leq \tau_0(\eta, k) := \frac{\eta^8}{K \cdot k \cdot 2^{4k}},$$

then the code $\mathcal{C}'$ is η-balanced and can be efficiently list decoded in the following sense:

If $\tilde{y}$ is $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}'$, then we can compute the list

$$\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') := \left\{ (z, \mathrm{dsum}_{W(k)}(z)) \mid z \in \mathcal{C}, \Delta\left(\mathrm{dsum}_{W(k)}(z), \tilde{y}\right) \leq \frac{1}{2} - \sqrt{\eta} \right\}$$

in time

$$n^{O(1/\tau_0(\eta,k)^4)} \cdot f(n),$$

where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}$. Otherwise, we return $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') = \emptyset$ with the same running time of the preceding case.

Note that the requirement on k in the above theorem is necessary for the lifted code $\mathcal{C}'$ to be η-balanced. Splittability will imply that the walk collection $W(k)$ is expanding, which gives us parity sampling for large k. Specifically, k must be large enough for $W(k)$

to be a $(1/2 + \eta_0/2, \eta)$-parity sampler.

Applying the list decoding tool above, we can perform unique decoding in the regime of η_0, η, and k being constant. With these choices the expansion required for splittability and the parity sampling strength are only required to be constants.

Lemma 4.6.2 (Decoding Step)**.** *Let $\eta_0 \in (0, 1/4)$ and $\eta < \min\{\eta_0, 1/16\}$. If $W(k)$ is a walk collection on vertex set $[n]$ with $k \geq k_0(\eta)$ and splittability $\tau \leq \tau_0(\eta, k)$, where k_0 and τ_0 are as in Theorem 4.6.1, we have the following unique decoding property:*

If $\mathcal{C} \subseteq \mathbb{F}_2^n$ is an η_0-balanced linear code that can be uniquely decoded in time $f(n)$, then $\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ is an η-balanced code that can be uniquely decoded in time $n^{O(1/\tau_0(\eta,k)^4)} \cdot f(n)$.

Proof. Using Theorem 4.6.1, we can list decode $\mathcal{C}'$ up to a radius of $1/2 - \sqrt{\eta}$ for any η if we have the appropriate parameters k and τ. Let $\tilde{y} \in \mathbb{F}_2^{W(k)}$ be a received word within the unique decoding radius of $\mathcal{C}'$. To perform unique decoding, we simply run the list decoding algorithm on $\tilde{y}$ and return the codeword on the resulting list which is closest to $\tilde{y}$; this will yield the correct result as long as the list decoding radius is larger than the unique decoding radius. It suffices to have $1/2 - \sqrt{\eta} > 1/4 \geq \Delta(\mathcal{C}')/2$. We choose parameters as follows:

1. Take $\eta < 1/16$ to ensure $1/2 - \sqrt{\eta} > 1/4$.
2. Let $k_0 = \Theta(\log(1/\eta))$ be the smallest integer satisfying the assumption in Theorem 4.6.1 with the chosen η. Take $k \geq k_0$.
3. Take $\tau \leq \tau_0(\eta, k) = \eta^8/(K \cdot k \cdot 2^{4k})$.

Note that k and τ satisfy the conditions of Theorem 4.6.1, so we can use this theorem to list decode a received word $\tilde{y}$ in time $n^{O(1/\tau_0(\eta,k)^4)} \cdot f(n)$. To unique decode, we return the closest y on the list to $\tilde{y}$ (or failure if the list is empty). ∎

Iteratively using the decoding step given by Lemma 4.6.2 above, we obtain unique decodability of all codes in a cascade (under suitable assumptions).

Lemma 4.6.3 (Code Cascade Decoding)**.** *Let $\eta_0 \in (0, 1/4)$ and $\eta < \min\{\eta_0, 1/16\}$. Suppose $\mathcal{C}_0 \subseteq \mathbb{F}_2^{n_0}, \mathcal{C}_1 \subseteq \mathbb{F}_2^{n_1}, \ldots, \mathcal{C}_\ell \subseteq \mathbb{F}_2^{n_\ell}$ is a code cascade where $\mathcal{C}_0$ is an η_0-balanced linear code that can be uniquely decoded in time $g(n_0)$.*

If for every $i \in [\ell]$ we have that $\mathcal{C}_i$ is obtained from $\mathcal{C}_{i-1}$ by a τ_i-splittable walk collection $W_i(k_i)$ on vertex set $[n_{i-1}]$ with $k_i \geq k_0(\eta)$ and $\tau_i \leq \tau_0(\eta, k_i)$, where k_0 and τ_0 are as in Theorem 4.6.1, then $\mathcal{C}_\ell$ is uniquely decodable in time

$$g(n_0) \cdot \prod_{i=1}^{\ell} n_{i-1}^{O(1/\tau_0(\eta,k_i)^4)}.$$

Proof. Induct on $i \in [\ell]$ applying Lemma 4.6.2 as the induction step. The code $\mathcal{C}_i$ produced during each step will have bias at most $\eta < \eta_0$, so the hypothesis of Lemma 4.6.2 will be met at each level of the cascade. ∎

We are almost ready to prove our first main theorem establishing decodability close to the Gilbert–Varshamov bound. We will need parameters for an instantiation of Ta-Shma's code that achieves the desired distance and rate (which will be developed in Section 4.8.1) and a lemma relating splittability to the spectral properties of the graphs used in the construction (to be proven in Appendix D.1.4).

Lemma 4.6.4 (Ta-Shma Codes I)**.** *For any $\beta > 0$, there are infinitely many values of $\varepsilon \in (0, 1/2)$ (with 0 as an accumulation point) such that for infinitely many values of $N \in \mathbb{N}$, there are explicit binary Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced), and

(ii) rate $\Omega(\varepsilon^{2+\beta})$.

Furthermore, $\mathcal{C}_{N,\varepsilon,\beta}$ is the direct sum lifting of a base code $\mathcal{C}_0 \subseteq \mathbb{F}_2^{n_0}$ using the collection of walks $W[0, t-1]$ on the s-wide replacement product of two graphs G and H, with the following parameters:

- *$s \geq s_0 := \max\{128, 26/\beta\}$.*
- *The inner graph H is a regular graph with $\sigma_2(H) \leq \lambda_2$, where $\lambda_2 = (16s^3 \log s)/s^{2s^2}$.*
- *The outer graph G is a regular graph with $\sigma_2(G) \leq \lambda_1$, where $\lambda_1 = \lambda_2^4/6$.*
- *The base code $\mathcal{C}_0$ is unique decodable in time $n_0^{O(1)}$ and has bias $\varepsilon_0 \leq \lambda_2^4/3$.*
- *The number of vertices t in the walks satisfies $\lambda_2^{2(1-5/s)(1-1/s)(t-1)} \leq \varepsilon$.*

Lemma 4.6.5. *Let $W(k)$ be either the collection $W[0, s-1]$ of walks of length s on the s-wide replacement product with outer graph G and inner graph H or the collection of walks over the vertex set $W[0, r]$, where $r \equiv -1 \pmod{s}$. Then $W(k)$ is τ-splittable with $\tau = \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2$.*

The statement of this first decoding theorem is more technical than Theorem 5.1.1, but it will be easier to prove while the latter will build on this theorem with a more careful tuning of parameters.

Theorem 4.6.6 (Main I)**.** *For every $\beta > 0$, there are infinitely many values $\varepsilon \in (0, 1/2)$ (with 0 an accumulation point) such that for infinitely many values of $N \in \mathbb{N}$ there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$, and

(iii) a unique decoding algorithm with running time $N^{O_\beta(\log(\log(1/\varepsilon)))}$.

Proof. We proceed as follows:

1. Let $\eta_0 = 1/10$ and $\eta = 1/100$ (these choices are arbitrary; we only need $\eta_0 < 1/4$, $\eta < 1/16$, and $\eta < \eta_0$). Let $k_0 = k_0(\eta)$ be the constant from Theorem 4.6.1 with this value of η.

2. Given $\beta > 0$, Lemma 4.6.4 provides a value s_0 such that the direct sum lifting on the s-wide replacement product with $s \geq s_0$ can achieve a rate of $\Omega(\varepsilon^{2+\beta})$ for infinitely many $\varepsilon \in (0, 1/2)$. Choose s to be an integer larger than both k_0 and s_0 that also satisfies
$$s^2 \cdot \left(\frac{s}{16}\right)^{-s^2} \leq \frac{\eta^8}{4K}, \tag{4.3}$$
where K is the constant from Theorem 4.6.1.

3. Use Lemma 4.6.4 with this value of s to obtain graphs G and H and a base code $\mathcal{C}_0$ having the specified parameters λ_1, λ_2, ε_0, and t, with the additional requirement that $t = s^\ell$ for some integer ℓ. These parameter choices ensure that the resulting code $\mathcal{C}_{N,\varepsilon,\beta}$ has the desired distance and rate. Since $s \geq 128$, we have $\lambda_2 = (16s^3 \log s)/s^{2s^2} \leq s^{-s^2}$. From the choice of t satisfying $\lambda_2^{2(1-5/s)(1-1/s)(t-1)} \leq \varepsilon$, we deduce that $\ell = O(\log(\log(1/\varepsilon)))$. Note also that the bias ε_0 of the code $\mathcal{C}_0$ is smaller than η_0.

4. Create a code cascade with ℓ levels using the s-wide replacement product of the graphs G and H as in Section 4.5.2, starting with $\mathcal{C}_0$ and ending with the final code $\mathcal{C}_\ell = \mathcal{C}_{N,\varepsilon,\beta}$. As the total number of vertices in a walk is $t = s^\ell$, each level of the code cascade will use walks with s vertices. Let $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_\ell$ be the sequence of codes in this cascade.

5. In order to satisfy the splittability requirement of Lemma 4.6.3, the walk collection $W_i(s)$ at each level of the code cascade must be τ-splittable, where $\tau \leq \tau_0(\eta, s^2)$. (We

use $k = s^2$ instead of $k = s$ in the requirement for a technical reason that will be clear in Section 4.8.2.) The bounds on the singular values of G and H and Lemma 4.6.5 ensure that

$$\tau = \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2 \leq 4\lambda_2 \leq 4s^{-s^2},$$

which is smaller than $\tau_0(\eta, s^2) = \eta^8/(K \cdot s^2 \cdot 2^{4s^2})$ by Eq. (4.3)

6. As all hypotheses of Lemma 4.6.3 are satisfied by the code cascade, we apply it to conclude that $\mathcal{C}_{N,\varepsilon,\beta}$ is uniquely decodable in time

$$g(n_0) \cdot \prod_{i=1}^{\ell} n_{i-1}^{O(1/\tau_0(\eta,s)^4)} \leq N^{O(1)} \cdot \prod_{i=1}^{\ell} N^{O_\beta(1)} = N^{O_\beta(\log(\log(1/\varepsilon)))},$$

where we use that $\mathcal{C}_0$ is uniquely decodable in time $n_0^{O(1)}$, $1/\tau_0(\eta, s) = 2^{O(1/\beta)}$, $n_{i-1} < n_\ell = N$ for every $i \in [\ell]$, and $\ell = O(\log(\log(1/\varepsilon)))$.

■

In the code cascade constructed in Theorem 4.6.6, the final number of vertices in a walk is $t = s^\ell$, where s is a sufficiently large constant that does not depend on ε. The limited choices for t place some restrictions on the values of the final bias ε that can be achieved. To achieve any bias ε for $\mathcal{C}_\ell$ we need to choose the parameters more carefully, which will be done in Section 4.8.2 to yield our next main result.

Theorem 4.6.7 (Main II). *For every $\beta > 0$ and every $\varepsilon > 0$ with β and ε sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$, and

(iii) a unique decoding algorithm with running time $N^{O_\beta(\log(\log(1/\varepsilon)))}$.

Ta-Shma obtained codes of rate $\Omega(\varepsilon^{2+\beta})$ with vanishing β as ε goes to zero. We obtain a unique decoding algorithm for this regime (with slightly slower decreasing β as ε vanishes). More precisely, using the parameters described in Section 4.8.3 and the running time analysis in Section 4.6.2, we obtain the following theorem which is our main result for unique decoding.

Theorem 4.6.8 (Main Unique Decoding (restatement of Theorem 5.1.1))**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$ where $\beta = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) a unique decoding algorithm with running time $N^{O_{\varepsilon,\beta}(1)}$.

Furthermore, if instead we take $\beta > 0$ to be an arbitrary constant, the running time becomes $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$ (fixed polynomial time).

Theorem 5.1.2 about gentle list decoding is proved in Section 4.8.4 after instantiating Ta-Shma codes in some parameter regimes in the preceding parts of Section 4.8.

4.6.2 Fixed Polynomial Time

In Theorem 4.6.7, a running time of $N^{O_\beta(\log(\log(1/\varepsilon)))}$ was obtained to decode Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta}$ of distance $1/2 - \varepsilon/2$ and rate $\Omega(\varepsilon^{2+\beta})$ for constant $\beta > 0$ and block length N. The running time contains an exponent which depends on the bias ε and is therefore not fixed polynomial time. We show how to remove this dependence in this regime of $\beta > 0$ being an arbitrary constant. More precisely, we show the following.

Theorem 4.6.9 (Fixed PolyTime Unique Decoding). *Let $\beta > 0$ be an arbitrary constant. For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$, and

(iii) a unique decoding algorithm with fixed polynomial running time $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$.

The list decoding framework finds a list of pairs $(z, y = \mathrm{dsum}(z))$ of size at most $N^{(1/\tau_0(\eta,k))^{O(1)}}$ at each level of the code cascade and recursively issues decoding calls to all z in this list. Since the number of lifts in the cascade is $\Omega(\log(\log(1/\varepsilon)))$, we end up with an overall running time of $N^{O_\beta(\log(\log(1/\varepsilon)))}$.

We will describe a method of pruning these lists which will lead to fixed polynomial running time. Let $1/2 - \sqrt{\eta}$, where $\eta > 0$ is a constant, be the list decoding radius used for a unique decoding step in the cascade. To achieve fixed polynomial time we will prune this polynomially large list of words to a constant size at each inductive step in Lemma 4.6.3. As we are working with parameters within the Johnson bound, the actual list of codewords has constant size $(1/\eta)^{O(1)}$. At every step, we will be able to find a small sublist whose size only depends on η that has a certain covering property, and then issue decoding calls to this much smaller list.

Definition 4.6.10 (ζ-cover). *Let $W(k) \subseteq [n]^k$, $\mathcal{C} \subseteq \mathbb{F}_2^n$, $A \subseteq \mathcal{C}$, and $\mathcal{L} = \{(z, \mathrm{dsum}_{W(k)}(z)) \mid z \in A\}$. We say that $\mathcal{L}' = \{(z^{(1)}, \mathrm{dsum}_{W(k)}(z^{(1)})), \ldots, (z^{(m)}, \mathrm{dsum}_{W(k)}(z^{(m)}))\}$ is a* ζ-cover *of $\mathcal{L}$ if for every $(z, y) \in \mathcal{L}$, there exists some $(z', y') \in \mathcal{L}'$ with $\mathrm{bias}(z - z') > 1 - 2\zeta$ (that is, either $\Delta(z, z') < \zeta$ or $\Delta(z, z') > 1 - \zeta$).*

Lemma 4.6.11 (Cover Compactness). *Let $W(k) \subseteq [n]^k$, $\mathcal{C} \subseteq \mathbb{F}_2^n$ be a linear η_0-balanced code,*

$\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ *be an* η*-balanced code, and* $\tilde{y} \in \mathbb{F}_2^{W(k)}$*. Define*

$$\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') := \left\{ (z, \mathrm{dsum}_{W(k)}(z)) \mid z \in \mathcal{C}, \Delta\left(\mathrm{dsum}_{W(k)}(z), \tilde{y}\right) \leq \frac{1}{2} - \sqrt{\eta} \right\}.$$

Suppose $\mathcal{L}'$ *is a* ζ*-cover of* $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}')$ *for some* $\zeta < 1/2$*. Further, suppose that for every* $(z', y') \in \mathcal{L}'$*, we have* $\Delta(y', \tilde{y}) \leq 1/2 - \sqrt{\eta}$*. If* $W(k)$ *is a* $(1 - 2\zeta, \eta)$*-parity sampler, then there exists* $\mathcal{L}'' \subseteq \mathcal{L}'$ *with* $\left|\mathcal{L}''\right| \leq 1/\eta$ *which is a* (2ζ)*-cover of* $\mathcal{L}$*.*

Proof. Build a graph where the vertices are pairs $(z', y') \in \mathcal{L}'$ and two vertices $(z^{(i)}, y^{(i)})$, $(z^{(j)}, y^{(j)})$ are connected iff $\mathrm{bias}(z^{(i)} - z^{(j)}) > 1 - 2\zeta$. Let $\mathcal{L}''$ be any maximal independent set of this graph. Any two vertices $(z^{(i)}, y^{(i)}), (z^{(j)}, y^{(j)}) \in \mathcal{L}''$ have $\mathrm{bias}(z^{(i)} - z^{(j)}) \leq 1 - 2\zeta$ and thus $\mathrm{bias}(y^{(i)} - y^{(j)}) \leq \eta$ since $W(k)$ is a $(1 - 2\zeta, \eta)$-parity sampler. This means that $\{y'' \mid (z'', y'') \in \mathcal{L}''\}$ is a code of distance at least $1/2 - \eta/2$. By the condition that $\Delta(y'', \tilde{y}) \leq 1/2 - \sqrt{\eta}$ for all $(z'', y'') \in \mathcal{L}''$ and the Johnson bound, we have $\left|\mathcal{L}''\right| \leq 1/\eta$.

Finally, we will show that $\mathcal{L}''$ is a (2ζ)-cover of $\mathcal{L}$. Let $(z, y) \in \mathcal{L}$. As $\mathcal{L}'$ is a ζ-cover of $\mathcal{L}$, there exists a pair $(z', y') \in \mathcal{L}$ with $\mathrm{bias}(z - z') > 1 - 2\zeta$, so z is within distance ζ of either z' or its complement $\overline{z'}$. The construction of $\mathcal{L}''$ as a maximal independent set ensures that there is some $(z'', y'') \in \mathcal{L}''$ with $\mathrm{bias}(z' - z'') > 1 - 2\zeta$, so z'' is also within distance ζ of either z' or its complement $\overline{z'}$. Applying the triangle inequality in all of the possible cases, we see that either $\Delta(z, z'') < 2\zeta$ or $\Delta(z, z'') > 1 - 2\zeta$, which implies $\mathcal{L}''$ is a (2ζ)-cover of $\mathcal{L}$. ∎

To decode in fixed polynomial time, we use a modification of the list decoding result Theorem 4.6.1 that outputs a ζ-cover $\mathcal{L}'$ of the list of codewords $\mathcal{L}$ instead of the list itself. Theorem 4.6.1 recovers the list $\mathcal{L}$ by finding $\mathcal{L}'$ and unique decoding every element of it. To get $\mathcal{L}'$, we use the same algorithm, but stop before the final decoding step. This removes the unique decoding time $f(n)$ of the base code from the running time of the list decoding algorithm. We will apply Lemma 4.6.11 after each time we call this ζ-cover

algorithm to pare the list down to a constant size before unique decoding; note that this loses a factor of 2 in the strength of the cover. To compensate for this, we will use a collection $W(k)$ with stronger parity sampling than required for Theorem 4.6.1. In that theorem, $W(k)$ was a $(1/2 + \eta_0/2, \eta)$-parity sampler to ensure that we obtained words within the list decoding radius $(1/4 - \eta_0/4)$ of the base code. By using a stronger parity sampler, the words in the pruned list $\mathcal{L}''$ will still be within the unique decoding radius even after accounting for the loss in the bias from cover compactness, which means decoding will still be possible at every level of the cascade. Fortunately, improving the parity sampling only requires increasing the walk length k by a constant multiplicative factor. The cover retrieval algorithm below will be proven in Section 4.9.

Theorem 4.6.12 (Cover retrieval for direct sum lifting)**.** *Let $\eta_0 \in (0, 1/4)$ be a constant, $\eta \in (0, \eta_0)$, $\zeta = 1/8 - \eta_0/8$, and*

$$k \geq k_0'(\eta) := \Theta(\log(1/\eta)).$$

Suppose $\mathcal{C} \subseteq \mathbb{F}_2^n$ is an η_0-balanced linear code and $\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ is the direct sum lifting of $\mathcal{C}$ on a τ-splittable collection of walks $W(k)$. There exists an absolute constant $K > 0$ such that if

$$\tau \leq \tau_0(\eta, k) := \frac{\eta^8}{K \cdot k \cdot 2^{4k}},$$

then the code $\mathcal{C}'$ is η-balanced, $W(k)$ is a $(1 - 2\zeta, \eta)$-parity sampler, and we have the following:

If $\tilde{y}$ is $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}'$, then we can compute a ζ-cover $\mathcal{L}'$ of the list

$$\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') := \left\{ (z, \mathrm{dsum}_{W(k)}(z)) \mid z \in \mathcal{C}, \Delta\left(\mathrm{dsum}_{W(k)}(z), \tilde{y}\right) \leq \frac{1}{2} - \sqrt{\eta} \right\}$$

in which $\Delta(y', \tilde{y}) \leq 1/2 - \sqrt{\eta}$ *for every* $(z', y') \in \mathcal{L}'$*, in time*

$$n^{O(1/\tau_0(\eta,k)^4)}.$$

Otherwise, we return $\mathcal{L}' = \emptyset$ *with the same running time of the preceding case.*

We now have all of the pieces necessary to prove Theorem 4.6.9. The process is essentially the same as our earlier unique decoding algorithm, except we use the cover retrieval algorithm from Theorem 4.6.12 instead of the full list decoding from Theorem 4.6.1. This allows us to insert a list pruning step in between obtaining the ζ-cover and calling the unique decoding algorithm for the previous level of the cascade.

Proof of Theorem 4.6.9. We use the code $\mathcal{C}_{N,\varepsilon,\beta}$ from Theorem 4.6.7 to get the desired distance and rate, with the slight modification of ensuring s is larger than k_0' from Theorem 4.6.12 rather than k_0 from Theorem 4.6.1.

Each level of the code cascade between $\mathcal{C}_{i-1}$ and $\mathcal{C}_i$ uses constant $\eta_0 < 1/4$ and $\eta < \min\{\eta_0, 1/16\}$, which allows for decoding in a similar fashion to Lemma 4.6.2 and Lemma 4.6.3. The difference is that we use Theorem 4.6.12 as the decoding step to obtain a ζ-cover $\mathcal{L}'$ of the list $\mathcal{L}(\tilde{y}, \mathcal{C}_{i-1}, \mathcal{C}_i)$ for $\tilde{y} \in \mathbb{F}_2^{n_i}$, where $\zeta = 1/8 - \eta_0/8$. By Lemma 4.6.11 and the fact that the walk collection is a $(1 - 2\zeta, \eta)$-parity sampler, $\mathcal{L}$ has a (2ζ)-cover $\mathcal{L}'' \subseteq \mathcal{L}'$ of size at most $1/\eta$. The covering property says that for every $(z, y) \in \mathcal{L}$, there exists some $(z'', y'') \in \mathcal{L}''$ such that z is within distance $2\zeta = 1/4 - \eta_0/4$ of either z'' or its complement $\overline{z''}$. This is the unique decoding radius of the η_0-balanced code $\mathcal{C}_{i-1}$, so we can recursively decode the list

$$\mathcal{L}'' \cup \{(\overline{z''}, \mathrm{dsum}(\overline{z''})) \mid (z'', \mathrm{dsum}(z'')) \in \mathcal{L}''\}$$

to obtain the complete list of codewords in $\mathcal{C}_{i-1}$.

Now we analyze the running time. On each level of the code cascade, we run the cover retrieval algorithm once to get $\mathcal{L}'$, prune the cover to get $\mathcal{L}''$, and then feed the union of $\mathcal{L}''$ and its complement (which has size at most $2/\eta$) into the unique decoding algorithm for the next level of the cascade. Letting $T_i(n_i)$ be the running time of unique decoding a single word in the code $\mathcal{C}_i \subseteq \mathbb{F}_2^{n_i}$, we have the following recurrence:

$$T_i(n_i) \leq n_i^{O(1/\tau_0(\eta,k)^4)} + \frac{2}{\eta} \cdot T_{i-1}(n_{i-1}) \quad \text{and} \quad T_0(n_0) = n_0^{O(1)}.$$

Note that the base code $\mathcal{C}_0$ has constant bias ε_0 and thus it has a fixed polynomial time decoding algorithm (e.g. Theorem 4.6.7). The height of the recursive call tree is the number of levels in the code cascade, which is $\ell = O(\log(\log(1/\varepsilon)))$, as in the proof of Theorem 4.6.6. Each node of this tree has a constant branching factor of $2/\eta$. Thus, the tree has $(\log(1/\varepsilon))^{O(1)}$ nodes, each of which costs at most $n_i^{O(1/\tau_0(\eta,k)^4)} \leq N^{O(1/\tau_0(\eta,k)^4)}$ time. Furthermore, in this regime of $\beta > 0$ being a constant, k is constant as well as η, so we have $N^{O(1/\tau_0(\eta,k)^4)} = N^{O_\beta(1)}$ and the total running time is $(\log(1/\varepsilon))^{O(1)} \cdot N^{O_\beta(1)}$. ∎

4.7 Satisfying the List Decoding Framework Requirements

The list decoding framework of [AJQ+20] is capable of decoding codes obtained from direct sum liftings, provided they satisfy a few requisite properties. The framework was originally shown to work for expander walks; we need to adapt it to our case of a code cascade based on walks on the s-wide replacement product. We will start with a broad overview of the list decoding algorithm and point out where various requirements arise.

The problem of finding a list of codewords in a direct sum lifting close to a received word can be viewed as finding approximate solutions to a k-XOR instance. This is done by solving a particular SOS program and rounding the resulting solution. The algorithm is unable to perform rounding if the k-XOR instance is based on an arbitrary collection

of walks $W(k)$; it can only handle liftings in which $W(k)$ satisfies a property called *tensoriality*. If $W(k)$ is tensorial, the SOS local variables in the solution can be approximated by product distributions, which will allow us to obtain a list of solutions by independent rounding. Tensoriality for expander walks is a consequence of a simpler property known as *splittability*, which is a certain measure of the expansion of a walk collection.

Unfortunately, the list returned by the rounding process will not contain codewords directly—instead, we only get a guarantee that all of the codewords we are looking for have a weak agreement (just over $1/2$) with something on this list. We will find the desired codewords by relying on the parity sampling of $W(k)$. If $W(k)$ is a sufficiently strong parity sampler, weak agreement in the lifted space corresponds to a much stronger agreement in the ground space. This will allow us to recover the codewords using the unique decoding algorithm of the base code.

To recap, applying the list decoding framework in our setting requires doing the following:

1. Proving parity sampling for the walks used in the code cascade (Section 4.7.1).
2. Showing that the walk collection of the s-wide replacement product is splittable (Appendix D.1.4).
3. Making Ta-Shma's construction compatible with the Sum-of-Squares machinery (Section 4.7.3) and then obtaining tensoriality from splittability (Section 4.7.4).

An additional complication is introduced by using a code cascade instead of a single decoding step: the above requirements need to be satisfied at every level of the cascade. The details of the proofs will often differ between the first level of the cascade, which is constructed using walks on the s-wide replacement product, and higher levels, which are walks on a directed graph whose vertices are walks themselves. Once we have es-

tablished all of the necessary properties, we will instantiate the list decoding framework in Section 4.9.

We will first define some convenient notation which will be used throughout this section.

Notation 4.7.1. *Let G be a d_1-regular outer graph and H be a d_2-regular inner graph used in Ta-Shma's s-wide replacement product.*

Let $0 \leq k_1 \leq k_2$ be integers. We define $W[k_1, k_2]$ to be the set of all walks starting at time k_1 and ending at time k_2 in Ta-Shma's construction. More precisely, since G and H are regular graphs, the collection $W[k_1, k_2]$ contains all walks obtained by sampling a uniform vertex $(v, h) \in V(G) \times V(H)$ and applying the operator

$$(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2-1}(\mathsf{I} \otimes \mathsf{A}_H) \cdots (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_1}(\mathsf{I} \otimes \mathsf{A}_H),$$

where the index i of each G_i is taken modulo s. Observe that when $k_1 = k_2$, we have $W[k_1, k_2] = V(G) \times V(H)$.

We define a family of Markov operators which will play a similar role to the graphs $\widehat{G}_i$ from the cascade described in Section 4.5.1, but for Ta-Shma's construction rather than expander walks.

Definition 4.7.2 (Split Operator)**.** *Let $0 \leq k_1 \leq k_2 < k_3$. We define the graph walk split operator*

$$\mathsf{S}_{k_1,k_2,k_3} \colon \mathbb{R}^{W[k_2+1,k_3]} \to \mathbb{R}^{W[k_1,k_2]}$$

such that for every $f \in \mathbb{R}^{W[k_2+1,k_3]}$,

$$\left(\mathsf{S}_{k_1,k_2,k_3}(f)\right)(w) := \mathbb{E}_{w' : ww' \in W[k_1,k_3]}[f(w')],$$

where ww' denotes the concatenation of the walks w and w'. The operator S_{k_1,k_2,k_3} can be defined

more concretely in matrix form such that for every $w \in W[k_1, k_2]$ and $w' \in W[k_2+1, k_3]$,

$$\left(\mathsf{S}_{k_1,k_2,k_3}\right)_{w,w'} = \frac{\mathbb{1}_{ww' \in W[k_1,k_3]}}{|\{\tilde{w} : w\tilde{w} \in W[k_1,k_3]\}|} = \frac{\mathbb{1}_{ww' \in W[k_1,k_3]}}{d_2^{2(k_3-k_2)}}.$$

4.7.1 Parity Sampling for the Code Cascade

To be able to apply the list decoding machinery to the code cascade $\mathcal{C}_0 \subseteq \mathbb{F}_2^{n_0}, \mathcal{C}_1 \subseteq \mathbb{F}_2^{n_1}, \ldots, \mathcal{C}_\ell \subseteq \mathbb{F}_2^{n_\ell}$, we need the direct sum lifting at every level to be a parity sampler. The first level in the cascade uses walks directly on the s-wide replacement product, which we can show is a good parity sampler using the spectral properties proven in Appendix D.1.3. However, it will be more convenient for calculating parameters later on to prove a weaker result, which will suffice for our purposes since we only need to obtain constant bias for every level of the cascade. We analyze the parity sampling of these walks with the same strategy Ta-Shma employed to show parity sampling for walks on expander graphs (which resulted in Theorem 4.5.2).

Claim 4.7.3. *Let $W[0, s-1]$ be the collection of walks on the s-wide replacement product of the graphs G and H and $z \in \mathbb{F}_2^{V(G)}$ be a word with $\mathrm{bias}(z) \leq \eta_0$. Let P_z be the diagonal matrix with entries $(\mathsf{P}_z)_{(v,h),(v,h)} = (-1)^{z_v}$ for $(v,h) \in V(G) \times V(H)$. If $\sigma_2((\mathsf{I} \otimes \mathsf{A}_H)G_i(\mathsf{I} \otimes \mathsf{A}_H)) \leq \gamma$ for all $0 \leq i \leq s-2$, then*

$$\left\| \prod_{i=0}^{s-2} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{P}_z \right\|_2 \leq (\eta_0 + 2\gamma)^{\lfloor (s-1)/2 \rfloor}.$$

Proof. Let $0 \leq j < s-2$ be even. Take a vector $v \in \mathbb{R}^{V(G)\times V(H)}$ with $\|v\|_2 = 1$ and let $v^{\|}$ and $v^{\perp}$ be its parallel and orthogonal components to the all ones vector. For $0 \leq i \leq s-2$, let $\mathsf{A}_i = (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$. Consider two terms $\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j\mathsf{P}_z$ of the product appearing in

the claim. Since P_z is unitary, $\left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j\mathsf{P}_z\right\|_2 = \left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j\right\|_2$. We have

$$\begin{aligned}
\left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j v\right\|_2 &\leq \left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j v^{\|}\right\|_2 + \left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j v^{\perp}\right\|_2 \\
&\leq \left\|\mathsf{A}_{j+1}\mathsf{P}_z\mathsf{A}_j v^{\|}\right\|_2 + \left\|\mathsf{A}_j v^{\perp}\right\|_2 \\
&\leq \left\|\mathsf{A}_{j+1}\mathsf{P}_z v^{\|}\right\|_2 + \sigma_2(\mathsf{A}_j) \\
&\leq \left\|\mathsf{A}_{j+1}(\mathsf{P}_z v^{\|})^{\|}\right\|_2 + \left\|\mathsf{A}_{j+1}(\mathsf{P}_z v^{\|})^{\perp}\right\|_2 + \sigma_2(\mathsf{A}_j) \\
&\leq \left\|(\mathsf{P}_z v^{\|})^{\|}\right\|_2 + \sigma_2(\mathsf{A}_{j+1}) + \sigma_2(\mathsf{A}_j) \\
&\leq \eta_0 + 2\gamma.
\end{aligned}$$

Applying this inequality to every two terms of the product, the result follows. ∎

Corollary 4.7.4. *Let $W[0, s-1]$ be the collection of walks on the s-wide replacement product of the graphs G and H and $\eta_0 > 0$. If $\sigma_2((\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)) \leq \gamma$ for all $0 \leq i \leq s-2$, then $W[0, s-1]$ is an (η_0, η)-parity sampler, where $\eta = (\eta_0 + 2\gamma)^{\lfloor (s-1)/2 \rfloor}$.*

Proof. Let $z \in \mathbb{F}_2^n$ have bias at most η_0. The bias of $\mathrm{dsum}_{W[0,s-1]}(z)$ is given by [7]

$$\mathrm{bias}(\mathrm{dsum}_{W[0,s-1]}(z)) = \left|\left\langle \mathbf{1}, \mathsf{P}_z \left(\prod_{i=0}^{s-2}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{P}_z\right)\mathbf{1}\right\rangle\right|,$$

where P_z is the diagonal matrix with entries $(\mathsf{P}_z)_{(v,h),(v,h)} = (-1)^{z_v}$ for $(v,h) \in V(G)\times V(H)$ and $\mathbf{1}$ is the all-ones vector. Since P_z is unitary, we have

$$\mathrm{bias}(\mathrm{dsum}_{W[0,s-1]}(z)) \leq \left\|\prod_{i=0}^{s-2}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{P}_z\right\|_2 \leq (\eta_0 + 2\gamma)^{\lfloor (s-1)/2 \rfloor} = \eta$$

by Claim 4.7.3. Hence $W[0, s-1]$ is an (η_0, η)-parity sampler. ∎

7. This is slightly different from the expression for the bias given in Appendix D.1.3, but both are equal since moving on the H component of the graph doesn't affect the bit assigned to a vertex.

For higher levels of the cascade, we need to prove parity sampling for collections of walks over walks. Since the walks on the first level contain s vertices, when we take walks on higher levels, the operator linking different walks together will always use G_{s-1} as the walk operator for the G step. Thus we can consider a more specific form of the split operator where we split at a time parameter that is one less than a multiple of s.

Definition 4.7.5. *Let $r \equiv -1 \pmod{s}$ be a positive integer. We define the operator $\mathsf{S}^{\triangle}_{r,r}$ as*

$$\mathsf{S}^{\triangle}_{r,r} = \mathsf{S}_{k_1,k_2,k_3},$$

where $k_1 = 0$, $k_2 = r$, and $k_3 = 2r+1$. In this case, $W[k_1, k_2] = W[k_2+1, k_3]$.

All levels of the code cascade beyond the first use walks generated by the directed operator $\mathsf{S}^{\triangle}_{r,r}$. Proving parity sampling for these walks is analogous to the proof of Corollary 4.7.4, but slightly simpler since the walk operator doesn't change with each step.

Claim 4.7.6. *Let $r \equiv -1 \pmod{s}$ be a positive integer and $z \in \mathbb{F}_2^{W[0,r]}$ be a word with $\mathrm{bias}(z) \leq \eta_0$. Let $\widetilde{\mathsf{P}}_z$ be the diagonal matrix with entries $(\widetilde{\mathsf{P}}_z)_{w,w} = (-1)^{z_w}$ for $w \in W[0,r]$. For every integer $k \geq 1$, we have*

$$\left\| \left(\mathsf{S}^{\triangle}_{r,r} \widetilde{\mathsf{P}}_z \right)^{k-1} \right\|_2 \leq \left(\eta_0 + 2 \cdot \sigma_2 \left(\mathsf{S}^{\triangle}_{r,r} \right) \right)^{\lfloor (k-1)/2 \rfloor}.$$

Proof. Take a vector $v \in \mathbb{R}^{W[0,r]}$ with $\|v\|_2 = 1$ and let $v^{\parallel}$ and $v^{\perp}$ be its parallel and orthogonal components to the all ones vector. Since $\widetilde{\mathsf{P}}_z$ is unitary, $\left\| \mathsf{S}^{\triangle}_{r,r} \widetilde{\mathsf{P}}_z \mathsf{S}^{\triangle}_{r,r} \widetilde{\mathsf{P}}_z \right\|_2 =$

$\left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\mathsf{S}^{\triangle}_{r,r}\right\|_2$. We have

$$\begin{aligned}
\left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\mathsf{S}^{\triangle}_{r,r}v\right\|_2 &\le \left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\mathsf{S}^{\triangle}_{r,r}v^{\|}\right\|_2 + \left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\mathsf{S}^{\triangle}_{r,r}v^{\perp}\right\|_2 \\
&\le \left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\mathsf{S}^{\triangle}_{r,r}v^{\|}\right\|_2 + \left\|\mathsf{S}^{\triangle}_{r,r}v^{\perp}\right\|_2 \\
&\le \left\|\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z v^{\|}\right\|_2 + \sigma_2(\mathsf{S}^{\triangle}_{r,r}) \\
&\le \left\|\mathsf{S}^{\triangle}_{r,r}(\widetilde{\mathsf{P}}_z v^{\|})^{\|}\right\|_2 + \left\|\mathsf{S}^{\triangle}_{r,r}(\widetilde{\mathsf{P}}_z v^{\|})^{\perp}\right\|_2 + \sigma_2(\mathsf{S}^{\triangle}_{r,r}) \\
&\le \left\|(\widetilde{\mathsf{P}}_z v^{\|})^{\|}\right\|_2 + \sigma_2(\mathsf{S}^{\triangle}_{r,r}) + \sigma_2(\mathsf{S}^{\triangle}_{r,r}) \\
&\le \eta_0 + 2\cdot\sigma_2(\mathsf{S}^{\triangle}_{r,r}).
\end{aligned}$$

As $\left\|(\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z)^{k-1}\right\|_2 \le \left\|(\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z)^2\right\|^{\lfloor (k-1)/2\rfloor}$, the result follows. ∎

Corollary 4.7.7. *Let $r \equiv -1 \pmod{s}$ be a positive integer and $\eta_0 > 0$. The collection of walks $W(k)$ with k vertices over the vertex set $W[0,r]$ using random walk operator $\mathsf{S}^{\triangle}_{r,r}$ is an (η_0,η)-parity sampler, where $\eta = (\eta_0 + 2\cdot\sigma_2(\mathsf{S}^{\triangle}_{r,r}))^{\lfloor (k-1)/2\rfloor}$.*

Proof. Let $z \in \mathbb{F}_2^{W[0,r]}$ have bias at most η_0. The bias of the direct sum lifting of z is given by

$$\mathrm{bias}(\mathrm{dsum}_{W(k)}(z)) = \left|\left\langle \mathbf{1}, \widetilde{\mathsf{P}}_z(\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z)^{k-1}\mathbf{1}\right\rangle\right|,$$

where $\widetilde{\mathsf{P}}_z$ is the diagonal matrix with entries $(\widetilde{\mathsf{P}}_z)_{w,w} = (-1)^{z_w}$ for $w \in W[0,r]$ and $\mathbf{1}$ is the all-ones vector. Since $\widetilde{\mathsf{P}}_z$ is unitary, we have

$$\left|\left\langle \mathbf{1}, \widetilde{\mathsf{P}}_z(\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z)^{k-1}\mathbf{1}\right\rangle\right| \le \left\|\left(\mathsf{S}^{\triangle}_{r,r}\widetilde{\mathsf{P}}_z\right)^{k-1}\right\|_2 \le \left(\eta_0 + 2\cdot\sigma_2\left(\mathsf{S}^{\triangle}_{r,r}\right)\right)^{\lfloor (k-1)/2\rfloor} = \eta$$

by Claim 4.7.6. Hence $W(k)$ is an (η_0,η)-parity sampler. ∎

4.7.2 Splittability of Ta-Shma's Construction

We investigate the splittability of the collection of walks generated by Ta-Shma's construction. In order to formally define this property, we will need the concept of an interval splitting tree, which describes how a walk is split into smaller and smaller pieces.

Definition 4.7.8 (Interval Splitting Tree)**.** *We say that a binary rooted tree $\mathcal{T}$ is a k-interval splitting tree if it has exactly k leaves and*

- *the root of $\mathcal{T}$ is labeled with $(0, m, k-1)$ for some $m \in \{0, 1, \ldots, k-2\}$, and*
- *each non-leaf non-root vertex v of $\mathcal{T}$ is labeled with (k_1, k_2, k_3) for some integer $k_2 \in [k_1, k_3 - 1]$. Suppose (k_1', k_2', k_3') is the label assigned to the parent of v. If v is a left child, we must have $k_1 = k_1'$ and $k_3 = k_2'$; otherwise, we must have $k_1 = k_2' + 1$ and $k_3 = k_3'$.*

Given an interval splitting tree $\mathcal{T}$, we can naturally associate a split operator S_{k_1,k_2,k_3} to each internal node (k_1, k_2, k_3). The splittability of a collection $W[0, k-1]$ of k-tuples is a notion of expansion at every node in the splitting tree.

Definition 4.7.9 ($(\mathcal{T}, \tau)$-splittability)**.** *The collection $W[0, k-1]$ is said to be $(\mathcal{T}, \tau)$-splittable if $\mathcal{T}$ is a k-interval splitting tree and*

$$\sigma_2(\mathsf{S}_{k_1,k_2,k_3}) \leq \tau$$

for every internal node (k_1, k_2, k_3) of $\mathcal{T}$.

If there exists some k-interval splitting tree $\mathcal{T}$ such that $W[0, k-1]$ is $(\mathcal{T}, \tau)$-splittable, then $W[0, k-1]$ will be called τ-splittable.

In order to prove that the collection of walks in Ta-Shma's construction is splittable, a split operator S_{k_1,k_2,k_3} can be related to the walk operator $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)$ as shown

below. This structural property will allow us to deduce spectral properties of S_{k_1,k_2,k_3} from the spectrum of $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)$.

Lemma 4.7.10. *Let $0 \leq k_1 \leq k_2 < k_3$. Suppose G is a d_1-regular outer graph on vertex set $[n]$ with walk operator G_{k_2} used at step k_2 of a walk on the s-wide replacement product and H is a d_2-regular inner graph on vertex set $[m]$ with normalized random walk operator A_H. Then there are orderings of the rows and columns of the representations of S_{k_1,k_2,k_3} and A_H as matrices such that*

$$\mathsf{S}_{k_1,k_2,k_3} = \Big((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)\Big) \otimes \mathsf{J}/d_2^{2(k_3-k_2-1)},$$

where $\mathsf{J} \in \mathbb{R}^{[d_2]^{2(k_2-k_1)} \times [d_2]^{2(k_3-k_2-1)}}$ is the all ones matrix.

Proof. Partition the set of walks $W[k_1, k_2]$ into the sets $W_{1,1}, \ldots, W_{n,m}$, where $w \in W_{i,j}$ if the last vertex of the walk $w_{k_2} = (v_{k_2}, h_{k_2})$ satisfies $v_{k_2} = i$ and $h_{k_2} = j$. Similarly, partition $W[k_2 + 1, k_3]$ into the sets $W'_{1,1}, \ldots, W'_{n,m}$, where $w' \in W'_{i,j}$ if the first vertex of the walk $w'_1 = (v_1, h_1)$ satisfies $v_1 = i$ and $h_1 = j$. Note that $\left|W_{i,j}\right| = d_2^{2(k_2-k_1)}$ and $\left|W'_{i,j}\right| = d_2^{2(k_3-k_2-1)}$ for all $(i, j) \in [n] \times [m]$, since there are d_2^2 choices for each step of the walk.

Now order the rows of the matrix S_{k_1,k_2,k_3} so that all of the rows corresponding to walks in $W_{1,1}$ appear first, followed by those for walks in $W_{1,2}$, and so on in lexicographic order of the indices (i, j) of $W_{i,j}$, with an arbitrary order within each set. Do a similar re-ordering of the columns for the sets $W'_{1,1}, \ldots, W'_{1,m}$. Observe that

$$\begin{aligned}\left(\mathsf{S}_{k_1,k_2,k_3}\right)_{w,w'} &= \frac{\mathbb{1}_{ww' \in W[k_1,k_3]}}{d_2^{2(k_3-k_2)}} \\ &= \frac{d_2^2 \cdot (\text{weight of transition from } (v_{k_2}, h_{k_2}) \text{ to } (v'_1, h'_1) \text{ in } (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H))}{d_2^{2(k_3-k_2)}},\end{aligned}$$

which only depends on the adjacency of the last vertex of w and the first vertex of w'. If

the vertices $w_{k_2} = (v_{k_2}, h_{k_2})$ and $w'_1 = (v_1, h_1)$ are adjacent, then

$$\left(\mathsf{S}_{k_1,k_2,k_3}\right)_{w,w'} = \left((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)\right)_{(v_{k_2},h_{k_2}),(v'_1,h'_1)} / d_2^{2(k_3-k_2-1)},$$

for every $w \in W_{w_{k_2}}$ and $w' \in W'_{w_{k_1}}$; otherwise, $\left(\mathsf{S}_{k_1,k_2,k_3}\right)_{w,w'} = 0$. Since the walks in the rows and columns are sorted according to their last and first vertices, respectively, the matrix S_{k_1,k_2,k_3} exactly matches the tensor product $((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)) \otimes \mathsf{J}/d_2^{2(k_3-k_2-1)}$.

■

Corollary 4.7.11. *Let $0 \leq k_1 \leq k_2 < k_3$. Suppose G is a d_1-regular outer graph with walk operator G_{k_2} used at step k_2 of a walk on the s-wide replacement product and H is a d_2-regular inner graph with normalized random walk operator A_H. Then*

$$\sigma_2(\mathsf{S}_{k_1,k_2,k_3}) = \sigma_2((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)).$$

Proof. Using Lemma D.1.10 and the fact that

$$\sigma_2(((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)) \otimes \mathsf{J}/d_2^{2(k_3-k_2-1)}) = \sigma_2((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)),$$

the result follows. ■

Remark 4.7.12. *Corollary D.1.11 is what causes the splittability argument to break down for Ta-Shma's original construction, as $\sigma_2(\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)) = 1$.*

By combining this result with the spectral bound from Fact D.1.5, we find that the collection of walks of length s on the s-wide replacement product is $(\mathcal{T}, \tau)$-splittable for any splitting tree $\mathcal{T}$, where τ is controlled by the second singular values of the graphs G and H. This analysis can also be applied to walks on higher levels of the cascade where the vertex set is $W[0, r]$.

Corollary 4.7.13 (Restatement of Lemma 4.6.5)**.** *The collection of walks $W[0, s-1]$ on the s-wide replacement product with outer graph G and inner graph H and the collection of walks $W(k)$ on the vertex set $W[0, r]$ with random walk operator $\mathsf{S}^{\triangle}_{r,r}$ and $r \equiv -1 \pmod{s}$ are both τ-splittable with $\tau = \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2$.*

Proof. By Corollary D.1.11 and Fact D.1.5, the split operator S_{k_1,k_2,k_3} for any $0 \leq k_1 \leq k_2 < k_3$ satisfies

$$\sigma_2(\mathsf{S}_{k_1,k_2,k_3}) = \sigma_2((\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{k_2}(\mathsf{I} \otimes \mathsf{A}_H)) \leq \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2,$$

so $W[0, s-1]$ is τ-splittable with $\tau = \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2$, as any internal node (k_1, k_2, k_3) of any s-interval splitting tree will have $\sigma_2(\mathsf{S}_{k_1,k_2,k_3}) \leq \tau$. The split operators of any k-interval splitting tree for the collection $W(k)$ are of the form S_{k_1,k_2,k_3} with $k_1 \equiv 0 \pmod{s}$ and $k_2, k_3 \equiv -1 \pmod{s}$, which means $W(k)$ is τ-splittable as well. ∎

4.7.3 *Integration with Sum-of-Squares*

Before defining tensoriality and obtaining it in our setting, we examine how the Sum-of-Squares hierarchy is used in the list decoding algorithm in more detail.

SOS Preliminaries: p-local PSD Ensembles

The 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 *p-local distributions* to describe the relaxations.

Solutions to a semidefinite relaxation of a CSP on n boolean variables using p levels

of the SOS hierarchy induce probability distributions μ_S over $\mathbb{F}_2^S$ for any set $S \subseteq [n]$ with $|S| \leq p$. 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 $\mathbb{F}_2$ such that for any set S with $|S| \leq p$, the collection of variables $\{\mathbf{Z}_i\}_{i \in S}$ has 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 at most p. We will refer to the collection $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ as a *p-local ensemble* of random variables.

For any $T \subseteq [n]$ with $|T| \leq p-2$ and any $\xi \in \mathbb{F}_2^T$, we can define a $(p - |T|)$-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 $|S| \leq p - |T|$, 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

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

is positive semidefinite, where $|S_1|, |S_2| \leq (p - |T|)/2$. 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 p-local ensembles such that for every conditioning on a set of size at most $(p-2)$, the conditional covariance matrix is PSD. We will refer to these as *p-local PSD ensembles*. We will also need a simple corollary of the above definitions.

Fact 4.7.14. *Let $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ be a p-local PSD ensemble and $W(k) \subseteq [n]^k$ For $1 \leq i < k$, define $W(i) \subseteq [n]^i$ to be the collection of tuples of size i appearing in elements of $W(k)$. For all $p' \leq p/2$, the collection $\left\{\mathbf{Z}_{\mathrm{set}(w)}\right\}_{w \in W(\leq p')}$ is a (p/p')-local PSD ensemble, where $W(\leq p') = \bigcup_{i=1}^{p'} W(i)$.*

For random variables $\mathbf{Z}_S$ in a p-local PSD ensemble, we use the notation $\{\mathbf{Z}_S\}$ to denote the distribution of $\mathbf{Z}_S$ (which exists when $|S| \leq p$). As we will work with ordered tuples of variables instead of sets, we define $\mathbf{Z}_w$ for $w \in [n]^k$ based on the set $S_w = \text{set}(w)$, taking care that repeated elements of w are always assigned the same value.

Definition 4.7.15 (Plausible assignment)**.** *Given $w = (w_1, \ldots, w_k) \in [n]^k$ and an assignment $\alpha \in \mathbb{F}_2^w$, we say that α is plausible for w if there are no distinct $i, j \in [k]$ such that $w_i = w_j$ but $\alpha_i \neq \alpha_j$.*

The distribution $\{\mathbf{Z}_w\} = \mu_w$ is defined as $\mu_w(\alpha) = \mu_{S_w}(\alpha|_{S_w})$ if $\alpha \in \mathbb{F}_2^w$ is plausible for w, and $\mu_w(\alpha) = 0$ otherwise.

Tensoriality

A key algorithm in the list decoding framework is propagation rounding (Algorithm 4.7.16), which solves a CSP to find solutions close to a codeword. Suppose $W(k) \subseteq [n]^k$ is a collection of walks, or more generally, a collection of any k-tuples. The algorithm starts with a local PSD ensemble $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ which is the solution to an SOS program for list decoding. Propagation rounding takes this solution and conditions some of the variables according to a random assignment to these variables to yield another local PSD ensemble $\mathbf{Z}'$.

Algorithm 4.7.16 (Propagation Rounding Algorithm, adapted from [AJQ+20]).

Input *An $(L+2k)$-local PSD ensemble $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$ and collection $W(k) \subseteq [n]^k$.*

Output *A random assignment $(\sigma_1, \ldots, \sigma_n) \in \mathbb{F}_2^n$ and $2k$-local PSD ensemble $\mathbf{Z}'$.*

1. *Choose $m \in \{1, \ldots, L/k\}$ uniformly at random.*
2. *For $j = 1, \ldots, m$, sample a walk w_j independently and uniformly from $W(k)$.*
3. *Write $S = \bigcup_{j=1}^{m} \mathrm{set}(w_j)$ for the set of the seed vertices.*
4. *Sample an assignment $\sigma : S \to \mathbb{F}_2$ 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 $i \in [n]$, sample independently $\sigma_i \sim \{\mathbf{Z}'_i\}$.*
7. *Output $(\sigma_1, \ldots, \sigma_n)$ and $\mathbf{Z}'$.*

If the collection $W(k) \subseteq [n]^k$ used in the direct sum lifting is amenable to SOS rounding, the conditioned ensemble $\mathbf{Z}'$ will be able to recover a word close to some codeword on the list. This is quantified by the following *tensorial* properties. We will see shortly how splittability will be used to obtain tensoriality in our setting.

Definition 4.7.17 (Tensorial Walk Collection). *Let $W(k) \subseteq [n]^k$, $\mu \in [0,1]$, and $L \in \mathbb{N}$. Define Ω to be the set of all tuples (m, S, σ) obtainable in propagation rounding (Algorithm 4.7.16) on $W(k)$ with SOS degree parameter L. We say that $W(k)$ is (μ, L)-tensorial if the local PSD ensemble $\mathbf{Z}'$ returned by propagation rounding satisfies*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w \in W(k)} \left\| \{\mathbf{Z}'_w\} - \left\{\mathbf{Z}'_{w(1)}\right\} \cdots \left\{\mathbf{Z}'_{w(k)}\right\} \right\|_1 \leq \mu. \tag{4.4}$$

The framework actually uses a strengthening of the above property, in which variables for pairs of walks chosen independently approximately behave as a product.

Definition 4.7.18 (Two-Step Tensorial Walk Collection)**.** *Let $W(k) \subseteq [n]^k$, $\mu \in [0,1]$, and $L \in \mathbb{N}$. Define Ω to be the set of all tuples (m, S, σ) obtainable in propagation rounding (Algorithm 4.7.16) on $W(k)$ with SOS degree parameter L. We say that $W(k)$ is (μ, L)-two-step tensorial if it is (μ, L)-tensorial and the local PSD ensemble $\mathbf{Z}'$ returned by propagation rounding satisfies the additional condition*

$$\mathop{\mathbb{E}}_{\Omega} \mathop{\mathbb{E}}_{w,w' \in W(k)} \left\| \{\mathbf{Z}'_w \mathbf{Z}'_{w'}\} - \{\mathbf{Z}'_w\}\{\mathbf{Z}'_{w'}\} \right\|_1 \leq \mu.$$

From Directed to Undirected

In order to apply the list decoding framework using the directed split operator S_{k_1,k_2,k_3}, we will replace it with the symmetrized version

$$\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3}) = \begin{pmatrix} 0 & \mathsf{S}_{k_1,k_2,k_3} \\ \left(\mathsf{S}_{k_1,k_2,k_3}\right)^\dagger & 0 \end{pmatrix}$$

and show how $\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})$ corresponds to a particular undirected graph.

Definition 4.7.19. *Let $0 \leq k_1 \leq k_2 < k_3$. We define the operator $\mathfrak{S}_{k_2,k_3,k_1} : \mathbb{R}^{W[k_1,k_2]} \to \mathbb{R}^{W[k_2+1,k_3]}$ such that for every $f \in \mathbb{R}^{W[k_1,k_2]}$,*

$$\left(\mathfrak{S}_{k_2,k_3,k_1}(f)\right)(w') := \mathbb{E}_{w : ww' \in W[k_1,k_3]}[f(w)],$$

for every $w' \in W[k_2+1, k_3]$.

The operator $\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})$ defines an undirected weighted bipartite graph on the vertices $W[k_1,k_2] \cup W[k_2+1,k_3]$. We can see that $\mathfrak{S}_{k_2,k_3,k_1}$ is the adjoint of S_{k_1,k_2,k_3}, which means that 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 $W[k_1,k_2]$ and the other is in

$W[k_2+1, k_3]$.

Claim 4.7.20.

$$\left(\mathsf{S}_{k_1,k_2,k_3}\right)^\dagger = \mathfrak{S}_{k_2,k_3,k_1}.$$

Proof. Let $f \in C^{W[k_1,k_2]}$ and $g \in C^{W[k_2+1,k_3]}$. For $i \leq j$, define $\Pi_{i,j}$ to be the uniform distribution on $W[i,j]$. We show that $\left\langle f, \mathsf{S}_{k_1,k_2,k_3} g \right\rangle = \left\langle \mathfrak{S}_{k_2,k_3,k_1} f, g \right\rangle$. On one hand we have

$$\begin{aligned}
\left\langle f, \mathsf{S}_{k_1,k_2,k_3} g \right\rangle &= \mathbb{E}_{w \in W[k_1,k_2]} \left[f(w) \mathbb{E}_{w' : ww' \in W[k_1,k_3]} [g(w')] \right] \\
&= \mathbb{E}_{w \in W[k_1,k_2]} \left[f(w) \sum_{w' \in W[k_2+1,k_3]} \frac{\Pi_{k_1,k_3}(ww')}{\Pi_{k_1,k_2}(w)} g(w') \right] \\
&= \sum_{w \in W[k_1,k_2]} \Pi_{k_1,k_2}(w) f(w) \sum_{w' \in W[k_2+1,k_3]} \frac{\Pi_{k_1,k_3}(ww')}{\Pi_{k_1,k_2}(w)} g(w') \\
&= \sum_{ww' \in W[k_1,k_3]} f(w) g(w') \Pi_{k_1,k_3}(ww').
\end{aligned}$$

On the other hand we have

$$\begin{aligned}
\left\langle \mathfrak{S}_{k_2,k_3,k_1} f, g \right\rangle &= \mathbb{E}_{w' \in W[k_2+1,k_3]} \left[\mathbb{E}_{w : ww' \in W[k_1,k_3]} [f(w)] g(w') \right] \\
&= \mathbb{E}_{w' \in W[k_2+1,k_3]} \left[\sum_{w \in W[k_1,k_2]} \frac{\Pi_{k_1,k_3}(ww')}{\Pi_{k_2+1,k_3}(w')} f(w) g(w') \right] \\
&= \sum_{w' \in W[k_2+1,k_3]} \Pi_{k_2+1,k_3}(w') \sum_{w \in W[k_1,k_2]} \frac{\Pi_{k_1,k_3}(ww')}{\Pi_{k_2+1,k_3}(w')} f(w) g(w') \\
&= \sum_{ww' \in W[k_1,k_3]} f(w) g(w') \Pi_{k_1,k_3}(ww').
\end{aligned}$$

Hence, $\mathfrak{S}_{k_2,k_3,k_1} = (\mathsf{S}_{k_1,k_2,k_3})^\dagger$ as claimed. ∎

Variables for Walks on the s-wide Replacement Product

When analyzing walks on the s-wide replacement product, we actually need to use two separate, but related, local PSD ensembles. In Ta-Shma's construction, the vertices of the outer graph G correspond to positions in the base code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$, where $n = |V(G)|$. Given a vertex $(v, h) \in V(G) \times V(H)$ in the s-wide replacement product and codeword $z \in \mathcal{C}_0$, (v, h) is assigned bit z_v, regardless of the vertex h of the inner graph. We will enforce this property by working with variables in $V(G)$ rather than the full $V(G) \times V(H)$. The local PSD ensemble $\mathbf{Z} = \{\mathbf{Z}_v\}_{v \in V(G)}$ contains one variable for every vertex of G, with local distributions for sets of variables up to a given size. For a walk w on the s-wide replacement product, we will use $\mathbf{Z}_w$ as an abbreviation for $\mathbf{Z}_{S_w}$, where S_w is the set of all G-components of vertices visited on the walk.

The constraints of the CSP are placed on walks on the s-wide replacement product that do care about the H-component of the vertices, so we define a second local PSD ensemble $\mathbf{Y} = \{\mathbf{Y}_{(v,h)}\}_{(v,h) \in V(G) \times V(H)}$ with a variable for each vertex of the s-wide replacement product of G and H. It is this collection $\mathbf{Y}$ for which we need to prove tensoriality in order to use the list decoding framework. When we perform propagation rounding, we condition the ensemble $\mathbf{Z}$ on a random assignment σ to a subset $S \subseteq V(G)$, rather than conditioning $\mathbf{Y}$ on a random assignment to a subset of $V(G) \times V(H)$. Working with $\mathbf{Z}$ ensures that the rounded assignments will be consistent on each cloud of the s-wide replacement product. Since the bit assigned to a vertex (v, h) only depends on v, independent rounding of $\{\mathbf{Z} \mid \mathbf{Z}_S = \sigma\}$ will also yield the desired rounding of $\{\mathbf{Y} \mid \mathbf{Z}_S = \sigma\}$.

We can define $\mathbf{Y}$ based on the ensemble $\mathbf{Z}$ more concretely. Suppose $S' \subseteq V(G) \times V(H)$ is a subset of size at most p, where p is the locality of the ensemble, and define $T = \{v \mid (v, h) \in S'\}$. The distribution $\mu_{S'}$ of $\mathbf{Y}_{S'}$ is defined based on the distribution μ_T of $\mathbf{Z}_T$ by $\mu_{S'}(\alpha) = \mu_T(\alpha|_T)$, where $\alpha \in \mathbb{F}_2^{S'}$ is an assignment to S' whose value on each vertex (v, h) only depends on v.

Observe that the introduction of the ensemble $\mathbf{Y}$ is only necessary on the first level of the Ta-Shma code cascade between the codes $\mathcal{C}_0$ and $\mathcal{C}_1$, which takes place on the s-wide replacement product. Higher levels of the cascade use walks on graphs whose vertices are the walks from the level below. The association of the bits of a codeword to the vertices of this graph has no consistency requirement, so we simply use a single local ensemble $\mathbf{Z}$ with a variable for each vertex.

4.7.4 *Splittability Implies Tensoriality*

The connection between splittability and tensoriality will be made with the help of a version of the triangle inequality.

Claim 4.7.21 (Triangle inequality, adapted from [AJQ+20])**.** *Let $s \in \mathbb{N}^+$ and $\mathcal{T}$ be an s-interval splitting tree. Then*

$$\mathop{\mathbb{E}}_{w\in W[0,s-1]} \left\| \{\mathbf{Z}_w\} - \prod_{i=0}^{s-1} \left\{\mathbf{Z}_{w(i)}\right\} \right\|_1 \leq \sum_{(k_1,k_2,k_3)\in\mathcal{T}} \mathop{\mathbb{E}}_{w\in W[k_1,k_3]} \left\| \{\mathbf{Z}_w\} - \left\{\mathbf{Z}_{w(k_1,k_2)}\right\}\left\{\mathbf{Z}_{w(k_2+1,k_3)}\right\} \right\|_1,$$

where the sum is taken over the labels of the internal nodes of $\mathcal{T}$.

To prove tensoriality, we will use the method of [BRS11] and [AJT19] to show that we can break correlations over expanding collections of tuples arising in the s-wide replacement product of the form

$$\mathop{\mathbb{E}}_{\substack{ww'\in W[k_1,k_3] \\ w\in W[k_1,k_2], w'\in W[k_2+1,k_3]}} \|\{\mathbf{Z}_{ww'}\} - \{\mathbf{Z}_w\}\{\mathbf{Z}_{w'}\}\|_1$$

appearing on the right-hand side of the triangle inequality.

The First Level of the Cascade

We now check the technical details to obtain tensoriality for the first lifting in the code cascade between the codes $\mathcal{C}_0$ and $\mathcal{C}_1$, which corresponds to taking s steps in Ta-Shma's construction. Recall that in order to obtain an assignment $z' \in \mathbb{F}_2^n$ whose lifting is consistent on vertices with the same G-component, we need to prove tensoriality for the ensemble $\mathbf{Y}$ with a variable for each vertex in $V(G) \times V(H)$.

The proof of tensoriality will make use of a specific entropic potential function. For an arbitrary random variable $\mathbf{X}$ taking values in a finite set $[q]$, define the function $\mathcal{H}(\mathbf{X})$ as

$$\mathcal{H}(\mathbf{X}) \coloneqq \frac{1}{q} \sum_{a \in [q]} \mathrm{H}(\mathbb{1}_{[\mathbf{X}=a]}) = \mathbb{E}_{a \in [q]} \mathrm{H}(\mathbb{1}_{[\mathbf{X}=a]}),$$

where H is the binary entropy function. Using this, we define a potential function for a weighted undirected graph G.

Definition 4.7.22 (Graph Potential)**.** *Let $G = (V, E)$ be a weighted graph with edge distribution Π_E. Let Π_V be the marginal distribution on V. Suppose that $\{\mathbf{Y}_i\}_{i \in V}$ is a p-local PSD ensemble for some $p \geq 1$. We define Φ^G to be*

$$\Phi^G \coloneqq \mathop{\mathbb{E}}_{i \sim \Pi_V} [\mathcal{H}(\mathbf{Y}_i)] .$$

Let $\mathcal{T}$ be an s-interval splitting tree associated with the s-wide replacement product of graphs G and H. We define

$$\Phi^{\mathcal{T}} \coloneqq \sum_{(k_1,k_2,k_3) \in \mathcal{T}} \Phi^{\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})},$$

where $\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})$ is the associated bipartite undirected graph of the operator S_{k_1,k_2,k_3}.

Lemma 4.7.23 (Splittability Implies Tensoriality)**.** *Let $W[0, s-1]$ be the walk collection of the*

s-wide replacement product of two graphs G and H. If $L \geq 128 \cdot (s^4 \cdot 2^{4s}/\mu^4)$ and $W[0, s-1]$ is τ-splittable with $\tau \leq \mu/(4s \cdot 2^{4s})$, then $W[0, s-1]$ is (μ, L)-tensorial.

Proof. We need to show that

$$\mathop{\mathbb{E}}_{w \in W[0,s-1]} \left\| \{\mathbf{Y}'_w\} - \prod_{i=0}^{s-1} \left\{ \mathbf{Y}'_{w(i)} \right\} \right\|_1 \leq \mu,$$

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

$$\Phi_m = \mathop{\mathbb{E}}_{S \sim \Pi_m} \mathop{\mathbb{E}}_{\sigma \sim \{\mathbf{Z}_S\}} \Phi^{\mathcal{T}}_{|\mathbf{Z}_S = \sigma}, \tag{4.5}$$

where the distribution Π_m on $S \subseteq V(G)$ is obtained from the process of choosing S in propagation rounding (Algorithm 4.7.16) once m has been fixed. Consider the error term

$$\mu_m := \mathop{\mathbb{E}}_{S \sim \Pi_m} \mathop{\mathbb{E}}_{\sigma \sim \{\mathbf{Z}_S\}} D(S, \sigma), \tag{4.6}$$

where $D(S, \sigma) := \mathbb{E}_{w \in W[0,s-1]} \left\| \{\mathbf{Y}_w \mid \mathbf{Z}_S = \sigma\} - \prod_{i=0}^{s-1} \left\{ \mathbf{Y}_{w(i)} \mid \mathbf{Z}_S = \sigma \right\} \right\|_1$. If $\mu_m \geq \mu/2$, then

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

For each choice of S and σ such that $D(S, \sigma) \geq \mu/2$, applying the triangle inequality from Claim 4.7.21 to the conditioned variables gives us

$$\begin{aligned} \frac{\mu}{2} &\leq \mathop{\mathbb{E}}_{w \in W[0,s-1]} \left\| \{\mathbf{Y}_w \mid \mathbf{Z}_S = \sigma\} - \prod_{i=0}^{s-1} \left\{ \mathbf{Y}_{w(i)} \mid \mathbf{Z}_S = \sigma \right\} \right\|_1 \\ &\leq \sum_{(k_1,k_2,k_3) \in \mathcal{T}} \mathop{\mathbb{E}}_{w \in W[k_1,k_3]} \left\| \{\mathbf{Y}_w \mid \mathbf{Z}_S = \sigma\} - \left\{ \mathbf{Y}_{w(k_1,k_2)} \mid \mathbf{Z}_S = \sigma \right\} \left\{ \mathbf{Y}_{w(k_2+1,k_3)} \mid \mathbf{Z}_S = \sigma \right\} \right\|_1. \end{aligned}$$

Hence, there exists (k_1, k_2, k_3) such that

$$\frac{\mu}{2s} \leq \mathop{\mathbb{E}}_{w \in W[k_1,k_3]} \left\| \{\mathbf{Y}_w \mid \mathbf{Z}_S = \sigma\} - \left\{\mathbf{Y}_{w(k_1,k_2)} \mid \mathbf{Z}_S = \sigma\right\}\left\{\mathbf{Y}_{w(k_2+1,k_3)} \mid \mathbf{Z}_S = \sigma\right\} \right\|_1.$$

Note that choosing $w \in W[0, s-1]$ uniformly and restricting to $w(k_1, k_3)$ gives a uniformly random element of $W[k_1, k_3]$. If we choose $w(k_1, k_2)$ or $w(k_2 + 1, k_3)$ with equal probability, then the final walk is distributed according to the stationary measure of $\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})$. Let w' denote the chosen walk. Observe that $\mathbf{Y}_{w'}$ is a deterministic function of $\mathbf{Z}_{w'} \mid \mathbf{Z}_S = \sigma$. Now, we sample $\mathbf{Z}_{w'} \mid \mathbf{Z}_S = \sigma$, which gives us a sample of $\mathbf{Y}_{w'}$. Applying Lemma 3.6.4, we have

$$\Phi^{\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})}_{|\{\mathbf{Y}_{w'}|\mathbf{Z}_S=\sigma\}} \leq \Phi^{\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})}_{\mathbf{Z}_S=\sigma} - \frac{\mu^2}{16s^2 \cdot 2^{4s}}.$$

This conditioning on an assignment to $\mathbf{Z}_{\mathrm{set}(w')} \mid \mathbf{Z}_S = \sigma$ does not increase the other terms of $\Phi^{\mathcal{T}}$ associated to split operators other than $\mathcal{U}(\mathsf{S}_{k_1,k_2,k_3})$ since entropy is non-increasing under conditioning. Similarly, conditioning on the remaining variables that are part of w but not w' does not increase $\Phi^{\mathcal{T}}$. Then, we obtain

$$\Phi_m - \Phi_{m+1} \geq \mathop{\mathbb{P}}_{S \sim \Pi_m, \sigma \sim \{\mathbf{Z}_S\}} [D(S, \sigma) \geq \mu_m/2] \cdot \frac{\mu^2}{16s^2 \cdot 2^{4s}}.$$

Since $s \geq \Phi_1 \geq \cdots \geq \Phi_{L/(s+1)} \geq 0$, there can be at most $32s^3 \cdot 2^{4s}/\mu^3$ indices $m \in [L/s]$ such that $\mu_m \geq \mu/2$. In particular, since the total number of indices is L/s, we have

$$\mathop{\mathbb{E}}_{m \in [L/s]} [\mu_m] \leq \frac{\mu}{2} + \frac{s}{L} \cdot \frac{32s^3 \cdot 2^{4s}}{\mu^3}.$$

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

Applying the list decoding framework will require the stronger property of two-step

tensoriality, which we can obtain under the same assumptions.

Lemma 4.7.24 (Splittability Implies Two-step Tensoriality)**.** *Let $W[0, s-1]$ be the walk collection of the s-wide replacement product of two graphs G and H. If $L \geq 128 \cdot (s^4 \cdot 2^{4s}/\mu^4)$ and $W[0, s-1]$ is τ-splittable with $\tau \leq \mu/(4s \cdot 2^{4s})$, then $W[0, s-1]$ is (μ, L)-two-step tensorial.*

Proof. Under our assumptions the (μ, L)-tensorial property follows from Lemma 4.7.23 (which is the only place where the assumption on τ is used), so we only need to show

$$\mathop{\mathbb{E}}_{w,w' \in W[0,s-1]} \left\| \{\mathbf{Y}'_w \mathbf{Y}'_{w'}\} - \{\mathbf{Y}'_w\}\{\mathbf{Y}'_{w'}\} \right\|_1 \leq \mu,$$

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

$$\Phi_m = \mathop{\mathbb{E}}_{S \sim \Pi_m} \mathop{\mathbb{E}}_{\sigma \sim \{\mathbf{Z}_S\}} \mathop{\mathbb{E}}_{w \in W[0,s-1]} \mathcal{H}(\mathbf{Y}_w \mid \mathbf{Z}_S = \sigma), \tag{4.7}$$

where the distribution Π_m on $S \subseteq V(G)$ is obtained from the process of choosing S in propagation rounding (Algorithm 4.7.16) once m has been fixed. Consider the error term

$$\mu_m := \mathop{\mathbb{E}}_{S \sim \Pi_m} \mathop{\mathbb{E}}_{\sigma \sim \{\mathbf{Z}_S\}} D(S, \sigma), \tag{4.8}$$

where $D(S, \sigma) := \mathbb{E}_{w,w' \in W[0,s-1]}[\|\{\mathbf{Y}_w \mathbf{Y}_{w'} \mid \mathbf{Z}_S = \sigma\} - \{\mathbf{Y}_w | \mathbf{Z}_S = \sigma\}\{\mathbf{Y}_{w'} | \mathbf{Z}_S = \sigma\}\|_1]$. If $\mu_m \geq \mu/2$, then

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

Let $G' = (V = W[0, s-1], E)$ be the graph with edges $E = \{\{w, w'\} \mid w, w' \in W[0, s-1]\}$. Local correlation (expectation over the edges) on this graph G' is the same as global correlation (expectation over two independent copies of vertices). Then, we

obtain [8]

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

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

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

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

We have already established that $W[0,s-1]$ is τ-splittable with $\tau = \sigma_2(G) + 2\sigma_2(H) + \sigma_2(H)^2$ in Corollary 4.7.13, so we can obtain (μ, L)-two-step tensoriality for any μ if this quantity is small enough.

Higher Levels of the Cascade

We now discuss tensoriality of the other levels of the cascade between $\mathcal{C}_{i-1}$ and $\mathcal{C}_i$ for $i \geq 2$. Tensorial properties are simpler to establish here than on the first level of the cascade. The relevant split operators are special cases of S_{k_1,k_2,k_3} where $k_1 \equiv 0 \pmod{s}$ and $k_2, k_3 \equiv -1 \pmod{s}$. The main difference now is that we can associate the parity bits of $\mathcal{C}_{i-1}$ with the vertices of $\mathcal{U}(\mathsf{S}^{\triangle}_{r,r})$, which themselves represent walks. As this association of parity bits doesn't need to satisfy a consistency condition, we only need to work with a single ensemble $\mathbf{Z}$ instead of working with two different ensembles as in the previous case. The proofs of Lemma 4.7.23 and Lemma 4.7.24 with these slight modifications give us two-step tensoriality.

Lemma 4.7.25 (Two-step Tensoriality for Higher Levels)**.** *Let $W(k)$ be the set of walks defined using $(k-1)$ steps of the operator $\mathsf{S}^{\triangle}_{r,r}$. If $L \geq 128\cdot(k^4\cdot 2^{4k}/\mu^4)$ and $W(k)$ is τ-splittable with*

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

$\tau \leq \mu/(4k \cdot 2^{4k})$, *then* $W(k)$ *is* (μ, L)*-two-step tensorial.*

We know from Corollary 4.7.13 that the collection of walks obtained from $\sigma_2(\mathsf{S}^{\triangle}_{r,r})$ is $(\sigma_2(G) + 2 \cdot \sigma_2(H) + \sigma_2(H)^2)$-splittable, so the parameters necessary to obtain two-step tensoriality are the same as in the first level of the cascade.

4.8 Choosing Parameters for Ta-Shma's Construction

We explore how some choices of parameters for Ta-Shma's construction interact with the requirements of our decoding algorithm. The analysis is divided into rounds of increasingly stronger decoding guarantees with later rounds relying on the codes obtained in previous rounds. Naturally, the stronger guarantees come with more delicate and technical considerations. We briefly summarize the goals of each round and some key parameters.

1. Round I: For any constant $\beta > 0$, we obtain *efficient unique decodable* codes $\mathcal{C}_\ell$ with distance at least $1/2 - \varepsilon$ and rate $\Omega(\varepsilon^{2+\beta})$ for infinitely many *discrete* values of $\varepsilon > 0$ (with ε as close to 0 as desired). In this regime, it suffices for the expansion of H to be constant. This round leads to Theorem 4.6.6.

2. Round II: Similar to Round I, but now ε can be any value in an interval $(0, b)$ with $b < 1/2$ being a function of β. Again the expansion of H can be constant. This round leads to Theorem 4.6.7.

3. Round III: We want β to vanish as ε vanishes (this is qualitatively similar to Ta-Shma's result). In this regime, we make the expansion of H be a function of ε, and we rely on the uniquely decodable codes of Round II. This round leads to Theorem 5.1.1.

4. Round IV: For any constant $\beta_0 > 0$, we obtain *efficient list decodable* codes $\mathcal{C}_\ell$ with list decoding radius $1/2 - \beta_0$ and rate $\Omega(\varepsilon^{2+\beta})$ with $\beta \to 0$ as $\varepsilon \to 0$. In this regime, we

make the expansion of H be a function of ε, and we rely on the uniquely decodable codes of Round III. This round leads to Theorem 5.1.2.

The way we choose parameters for Ta-Shma's construction borrows heavily from Ta-Shma's arguments in [TS17]. We fix some notation common to all rounds. A graph is said to be an (n, d, λ)-graph provided it has n vertices, is d-regular, and has second largest singular value of its normalized adjacency matrix at most λ.

Notation 4.8.1. *We use the following notation for the graphs G and H used in the s-wide replacement product.*

- *The outer graph G will be an (n', d_1, λ_1)-graph.*
- *The inner graph H will be a (d_1^s, d_2, λ_2)-graph.*

The parameters $n', d_1, d_2, \lambda_1, \lambda_2$ and s will be chosen in the subsequent sections.

4.8.1 Round I: Initial Analysis

We are given the dimension D of the desired code and $\varepsilon \in (0, 1/2)$. We set a parameter $\alpha \leq 1/128$ such that (for convenience) $1/\alpha$ is a power of 2 and

$$\frac{\alpha^5}{4\log_2(1/\alpha)} \geq \frac{1}{\log_2(1/\varepsilon)}. \tag{4.9}$$

We can assume that $\alpha \leq 1/128$ satisfy Eq. (D.2) since otherwise ε is a constant and we can use the list decodable codes from [AJQ+20]. The use of Eq. (D.2) will be clear shortly. It becomes a necessity from round III onward. For rounds I and II, the parameter α will be a constant, but it will be useful to establish the analysis in more generality now so that subsequent rounds can reuse it.

The inner graph H. The choice of H is similar to Ta-Shma's choice. More precisely, we set $s = 1/\alpha$ and $d_2 = s^{4s^2}$ (Ta-Shma took $d_2 = s^{4s}$). We obtain a Cayley graph $H = \mathrm{Cay}(\mathbb{F}_2^{4s\log_2(d_2)}, A)$ such that H is an $(n_2 = d_2^{4s}, d_2, \lambda_2)$ graph where $\lambda_2 = b_2/\sqrt{d_2}$ and $b_2 = 4s\log_2(d_2)$. (The set of generators, A, comes from a small bias code derived from a construction of Alon et al. [AGHP92], but we will rely on Ta-Shma's analysis embodied in **??** and not discuss it further.)

The base code $\mathcal{C}_0$. Set $\varepsilon_0 = 1/d_2^2 = \lambda_2^4/b_2^4 \leq \lambda_2^4/3$ (this choice differs from Ta-Shma's and it appears because we are essentially working with H^2 rather than H). We will choose a base code $\mathcal{C}_0$ such that the desired code will be obtained as a direct sum lifting of $\mathcal{C}_0$, and because this lifting preserves the dimension, the dimension of $\mathcal{C}_0$ should be D. We choose $\mathcal{C}_0$ to be an ε_0-balanced code with dimension D and block length $n = O_{\varepsilon_0}(D)$. For instance, we can start with any good (constant rate and relative distance) linear base code $\mathcal{C}_0$ that has an efficient unique decoding algorithm and obtain a ε_0-balanced lifted code that can be efficiently unique decoded (as long as ε_0 is constant) using the framework in [AJQ+20].

The outer graph G. Set $d_1 = d_2^4$ so that $n_2 = d_1^s$ as required by the s-wide replacement product. We apply Ta-Shma's explicit Ramanujan graph **??** with parameters n, d_1 and θ to obtain an (n', d_1, λ_1) Ramanujan graph G with $\lambda_1 \leq 2\sqrt{2}/\sqrt{d_1}$ and $n' \in [(1-\theta)n, n]$ or $n' \in [(1-\theta)2n, 2n]$. Here, θ is an error parameter that we set as $\theta = \lambda_2^4/6$ (this choice of θ differs from Ta-Shma). Because we can construct words with block length $2n$ (if needed) by duplicating each codeword, we may assume w.l.o.g. that n' is close to n and $(n - n') \leq \theta n \leq 2\theta n'$. See **??** for a more formal description of this graph.

Note that $\lambda_1 \leq \lambda_2^4/6$ since $\lambda_1 \leq 3/\sqrt{d_1} = 3/d_2^2 = 3 \cdot \lambda_2^4/b_2^4 \leq \lambda_2^4/6$. Hence, $\varepsilon_0 + 2\theta + 2\lambda_1 \leq \lambda_2^4$.

The walk length. Set the walk length $t-1$ to be the smallest integer such that

$$(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(t-1)} \leq \varepsilon.$$

This will imply using Ta-Shma's analysis that the bias of the final code is at most ε as shown later.

$s = 1/\alpha$, such that $\frac{\alpha^5}{4\log_2(1/\alpha)} \geq \frac{1}{\log_2(1/\varepsilon)}$
$H : (n_2, d_2, \lambda_2), \quad n_2 = d_1^s, \quad d_2 = s^{4s^2}, \quad \lambda_2 = \frac{b_2}{\sqrt{d_2}}, \quad b_2 = 4s\log d_2$
$G : (n', d_1, \lambda_1), \quad n' \approx n = O(D/\varepsilon_0^c), \quad d_1 = d_2^4, \quad \lambda_1 \leq \frac{2\sqrt{2}}{d_1}$
t : smallest integer such that $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(t-1)} \leq \varepsilon$

Claim 4.8.2. *We have $t-1 \geq s/\alpha = s^2$.*

Proof. Using $d_2 = s^{4s^2}$ and Eq. (D.2), we have

$$\left(\frac{1}{\lambda_2^2}\right)^{(1-5\alpha)(1-\alpha)s/\alpha} \leq \left(\frac{1}{\lambda_2^2}\right)^{s/\alpha} = \left(\frac{d_2}{b_2^2}\right)^{s/\alpha} \leq (d_2)^{s/\alpha} = s^{4s^3/\alpha}$$
$$= 2^{4s^3\log_2(s)/\alpha} = 2^{4\log_2(1/\alpha)/\alpha^4} \leq 2^{\log_2(1/\varepsilon)} = \frac{1}{\varepsilon}.$$

Hence, $\varepsilon \leq (\lambda_2^2)^{(1-5\alpha)(1-\alpha)s/\alpha}$ and thus $t-1$ must be at least s/α. ∎

Remark 4.8.3. *By our choice of t, we have $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(t-2)} \geq \varepsilon$. Since $1/(t-1) \leq \alpha$, we get $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)^2(t-1)} \geq \varepsilon$.*

Final Bias. We denote by $\mathcal{C}_\ell$ the final code obtained by t steps of the s-wide replacement product. The bias of $\mathcal{C}_\ell$ is given by Corollary D.1.8 (which in turn is a simple corollary of Ta-Shma's Fact D.1.7) as shown next.

Corollary 4.8.4. *The code $\mathcal{C}_\ell$ is ε-balanced.*

Proof. Using Corollary D.1.8, we have that the final bias

$$b := \left(\sigma_2(H^2)^{s-1} + (s-1)\cdot\sigma_2(H^2)^{s-2} + (s-1)^2\cdot\sigma_2(H^2)^{s-4}\right)^{\lfloor (t-1)/s\rfloor}$$

is bounded by

$$\begin{aligned}
b &\le (3(s-1)^2\sigma_2(H^2)^{s-4})^{((t-1)/s)-1} && \text{(Using } \sigma_2(H^2)\le 1/3s^2\text{)}\\
&\le ((\sigma_2(H^2)^{s-5})^{(t-1-s)/s}\\
&= \sigma_2(H^2)^{(1-5/s)(1-s/(t-1))(t-1)}\\
&\le \sigma_2(H^2)^{(1-5\alpha)(1-\alpha)(t-1)}\\
&= \left(\lambda_2^2\right)^{(1-5\alpha)(1-\alpha)(t-1)} \le \varepsilon,
\end{aligned}$$

where the last inequality follows from $s = 1/\alpha$ and $t-1 \ge s/\alpha$, the latter from Claim 4.8.2.

■

Rate. The proof of the rate follows a similar structure of Ta-Shma's original argument except that we take s to be a constant independent of ε so that ε_0, λ_1, and λ_2 are also constants independent of ε. Note that we previously said $\alpha = 1/s$ needs to satisfy Equation D.2, but that implies only an upper bound for s, and smaller (even constant) values for s are still permissible.

Claim 4.8.5. *$\mathcal{C}_\ell$ has rate $\Omega(\varepsilon^{2+26\cdot\alpha})$ provided $\varepsilon_0 > 0$ is constant.*

Proof. The support size is the number of walks of length t on the s-wide replacement

product of G and H (each step of the walk has d_2^2 options), which is

$$\begin{aligned}|V(G)||V(H)|d_2^{2(t-1)} = n' \cdot d_1^s \cdot d_2^{2(t-1)} &= n' \cdot d_2^{2(t-1)+4s} \leq n \cdot d_2^{2(t-1)+4s} \\ &= \Theta_{\varepsilon_0}\left(D \cdot d_2^{2(t-1)+4s}\right) \\ &= \Theta\left(D \cdot (d_2^2)^{t-1+2s}\right) \\ &= O\left(D \cdot (d_2^2)^{(1+2\alpha)(t-1)}\right),\end{aligned}$$

where the penultimate equality follows from the assumption that ε_0 is a constant.

Note that $d_2^\alpha = d_2^{1/s} = s^{4s} \geq b_2$ since $b_2 = 4s\log_2(d_2) = 16s^3\log_2(s) \leq s^4$ (recall that $s = 1/\alpha \geq 128$). Thus,

$$d_2^{1-2\alpha} = \frac{d_2}{d_2^{2\alpha}} \leq \frac{d_2}{b_2^2} = \frac{1}{\sigma_2(H^2)}.$$

We obtain

$$\begin{aligned}(d_2^2)^{(t-1)} &\leq \left(\frac{1}{\sigma_2(H^2)}\right)^{\frac{2(t-1)}{1-2\alpha}} \\ &\leq \left(\frac{1}{\varepsilon}\right)^{\frac{2}{(1-2\alpha)(1-5\alpha)(1-\alpha)^2}} && \text{(Using Remark 4.8.3)} \\ &\leq \left(\frac{1}{\varepsilon}\right)^{2(1+10\alpha)},\end{aligned}$$

which implies a block length of

$$O\left(D \cdot (d_2^2)^{(1+2\alpha)(t-1)}\right) = O\left(D\left(\frac{1}{\varepsilon}\right)^{2(1+10\alpha)(1+2\alpha)}\right) = O\left(D\left(\frac{1}{\varepsilon}\right)^{2(1+13\alpha)}\right).$$

■

Lemma 4.8.6 (Codes Near the GV bound I)**.** *For every constant $\beta > 0$, there exists a sufficiently large constant s in the above analysis so that for any dimension value $D \in \mathbb{N}^+$ (sufficiently large) and $\varepsilon > 0$ (sufficiently small) the final code $\mathcal{C}_{N,\varepsilon,\beta}$, where N is the block length, satisfies*

- $\mathcal{C}_{N,\varepsilon,\beta}$ *is* ε*-balanced,*

- $\mathcal{C}_{N,\varepsilon,\beta}$ *has rate* $\Omega(\varepsilon^{2+\beta})$*, and*

- $\mathcal{C}_{N,\varepsilon,\beta}$ *is a linear code of dimension* D*.*

Remark 4.8.7. *As a consequence of code cascading, the final attainable walk lengths have the form* $s^\ell - 1$ *where* ℓ *is a positive integer. Given* $\beta > 0$*, we have infinitely many values of* ε *attainable by such walk lengths which gives infinitely many codes* $\mathcal{C}_{N,\varepsilon,\beta}$*. This means that although the bias* ε *cannot be arbitrary, we have an infinite sequence of values of* ε *for which the rates of the codes* $\mathcal{C}_{N,\varepsilon,\beta}$ *are near the GV bound. In Section 4.8.2, we show how to bypass this artificial limitation. These codes are used in the proof of Theorem 4.6.6.*

We can view the above analysis as defining a function Γ that receives

- the dimension $D \in \mathbb{N}^+$,

- the final bias $\varepsilon > 0$,

- the approximating error $\alpha \in (0, 1/128]$ with $s := 1/\alpha$ being a power of two, and

- a multiplying factor $Q \in \mathbb{N}^+$ such that $d_2 = s^{4s^2 \cdot Q}$ (in the above Q was 1).

and outputs a tuple of parameters $(t, \varepsilon_0, \theta, d_1, \lambda_1, n')$, graphs G and H (as above) where, in particular, the number of steps $t \in \mathbb{N}^+$ is such that the final code $\mathcal{C}_\ell$ has bias at most ε and rate $\Omega(\varepsilon^{2+26\cdot\alpha})$.

In future rounds, Γ may be called with $Q = s$ instead of $Q = 1$. This will cause d_2 to increase from s^{4s^2} to $s^{4s^2 \cdot Q}$, and so in the proof of Claim 4.8.2, $2^{4\log_2(1/\alpha)/\alpha^4}$ will be replaced by $2^{4\log_2(1/\alpha)/\alpha^5}$. This explains why Eq. (D.2) has a stricter requirement than needed in the $Q = 1$ case above.

4.8.2 Round II: A More Careful Analysis

We are given the dimension of the code D and $\varepsilon \in (0, 1/2)$. As before, we set a parameter $\alpha \leq 1/128$ such that (for convenience) $1/\alpha$ is a power of 2. Set $s = 1/\alpha$ and $Q = s$.

Apply Γ to $(D, \varepsilon, \alpha, Q)$ to obtain all parameters except t. Choose t to be the smallest integer satisfying

$$(\lambda_2^2)^{(1-5\alpha)(1-2\alpha)(1-\alpha)(t-1)} \leq \varepsilon,$$

where observe that an extra $(1 - 2\alpha)$ factor appears in the exponent. This change in t will worsen the rate but by losing a factor of $\frac{1}{1-2\alpha}$ in the exponent, we can lower bound the rate. That is, $(d_2^2)^{-(t-1)} = \Omega(\varepsilon^{\frac{2+26\cdot\alpha}{1-2\alpha}})$.

Set $\ell \in \mathbb{N}^+$ to be the smallest value such that $s^\ell \geq t$ (here we are implicitly assuming that $t > s$). If $s^\ell = t$, we are done since we can use all the parameters returned by Γ for the construction of $\mathcal{C}_\ell$. Now assume $s^\ell > t$ and let $\zeta = t/s^{\ell-1}$. Note that $\zeta \in (1, s)$. Choose P to be the integer in the interval $[Q, s \cdot Q]$ such that

$$0 \leq \frac{P}{Q} - \zeta \leq \frac{1}{Q}.$$

Because $s^\ell > t$, and only powers of s may be chosen for walk length, we might overshoot in walk length by a multiplicative factor of s. This will cause a corresponding decay in rate computation that we cannot afford. To overcome this, in the last level of the cascade between codes $\mathcal{C}_{\ell-1}$ and $\mathcal{C}_\ell$, perform the direct sum over walks of length $(P - 1)$ instead of length $(s - 1)$. The new total number of vertices is $t' = Ps^{\ell-1}$. Note that P can be as large as s^2, so our splittability guarantee of $W(P)$ (the walk collection from the lift between $\mathcal{C}_{\ell-1}$ and $\mathcal{C}_\ell$) has to be strong enough to accommodate this larger arity and not only arity s.

Claim 4.8.8. *We have* $t - 1 \leq \frac{t'-1}{Q} \leq (1 + 2\alpha)(t - 1)$.

Proof. By construction, we have the sequence of implications

$$\begin{aligned}
0 \leq \frac{P}{Q} s^{\ell-1} - \zeta s^{\ell-1} &\leq \frac{s^{\ell-1}}{Q} \\
\Rightarrow 0 \leq \frac{t'}{Q} - t &\leq \frac{s^{\ell-1}}{Q} \leq \frac{t}{Q} \\
\Rightarrow t - \frac{1}{Q} &\leq \frac{t'-1}{Q} \leq (t-1)\left(1 + \frac{1}{Q}\right) + 1,
\end{aligned}$$

from which we obtain

$$t - 1 \leq t - \frac{1}{Q} \leq \frac{t'-1}{Q}$$

and

$$\frac{t'-1}{Q} \leq (t-1)\left(1 + \frac{1}{Q}\right) + 1 = (1+\alpha)(t-1) + 1 < (1+2\alpha)(t-1),$$

the latter using $Q = s = 1/\alpha$. ∎

We apply Γ again but this time to $(D, \varepsilon, \alpha, 1)$ to obtain new parameters $(t'', \varepsilon_0', \theta', d_1', \lambda_1', n'')$, and graphs G' and H'.

Claim 4.8.9. *The code $\mathcal{C}_\ell'$ obtained by t' walk steps on the s-wide replacement product of G' and H' from the second application of Γ has bias at most ε and rate $\Omega(\varepsilon^{2+40\alpha})$.*

Proof. Let $d_2 = s^{4s^2 \cdot Q}$, $b_2 = 4s \log_2(d_2)$ and $\lambda_2 = b_2/\sqrt{d_2}$ be the parameters of the first invocation of Γ. Recall that t was chosen to be the smallest integer satisfying

$$(\lambda_2^2)^{(1-5\alpha)^2(1-\alpha)(t-1)} \leq \varepsilon.$$

Let $d_2' = s^{4s^2}$, $b_2' = 4s \log_2(d_2')$ and $\lambda_2' = b_2'/\sqrt{d_2'}$ be the parameters of the second invoca-

tion of Γ. Observe that

$$(\lambda_2')^Q = \frac{(b_2')^Q}{\sqrt{(d_2')^Q}} = \frac{(b_2'^Q)}{\sqrt{d_2}} = \frac{(16s^3\log_2(s))^Q}{s^{2s^2\cdot Q}}$$
$$\leq \frac{s^{4Q}}{s^{2s^2\cdot Q}} = \frac{1}{s^{2s^2\cdot Q(1-\frac{2}{s^2})}} = \left(\frac{1}{s^{2s^2\cdot Q}}\right)^{1-2\alpha} \leq \left(\frac{b_2}{\sqrt{d_2}}\right)^{1-2\alpha} = \lambda_2^{1-2\alpha}.$$

Then the bias of $\mathcal{C}'_\ell$ is at most

$$(((\lambda_2')^Q)^2)^{(1-5\alpha)(1-\alpha)(t'-1)/Q} \leq (\lambda_2^2)^{(1-5\alpha)(1-2\alpha)(1-\alpha)(t'-1)/Q}$$
$$\leq (\lambda_2^2)^{(1-5\alpha)(1-2\alpha)(1-\alpha)(t-1)} \leq \varepsilon.$$

For the rate computation of $\mathcal{C}'_\ell$, we will lower bound the term $((d_2')^2)^{-(t'-1)}$. Since $d_2 = (d_2')^Q$, $(d_2^2)^{-(t-1)} = \Omega(\varepsilon^{\frac{2+26\cdot\alpha}{1-2\alpha}})$ and $\frac{t'-1}{Q} \leq (1+2\alpha)(t-1)$ (the latter by Claim 4.8.8), the rate of $\mathcal{C}'_\ell$ is

$$\Omega(((d_2')^2)^{-(t'-1)}) = \Omega((d_2^2)^{-(t'-1)/Q}) = \Omega((d_2^2)^{-(1+2\alpha)(t-1)}) = \Omega((\varepsilon^{2+26\cdot\alpha})^{\frac{1+2\alpha}{1-2\alpha}}) = \Omega(\varepsilon^{2+40\cdot\alpha}).$$

■

4.8.3 *Round III: Vanishing β as ε Vanishes*

We are given the dimension of the code D and $\varepsilon \in (0, 1/2)$. As before, we set a parameter $\alpha \leq 1/128$ such that (for convenience) $1/\alpha$ is a power of 2. Set $s := 1/\alpha$.

We will consider the regime where s is a function of ε. As a consequence, the parameters $d_2, \lambda_2, d_1, \lambda_1, \varepsilon_0$ will also depend on ε. Since $x \leq 1/\log_2(1/x)$ for $x \leq 1/2$ (and $\alpha \leq 1/2$), if α satisfies $\alpha^6/4 \geq 1/\log_2(1/\beta)$, it also satisfies Eq. (D.2) (we lose a log factor by replacing $1/\log_2(1/\alpha)$ by α, but we will favor simplicity of parameters). In particular,

we can set α so that s is

$$s = \Theta((\log_2(1/\varepsilon))^{1/6}),$$

and satisfy Eq. (D.2).

We follow the same choices as in Round II except for the base code $\mathcal{C}_0$.

The base code $\mathcal{C}_0$. Set $\varepsilon_0 = 1/d_2^2 = \lambda_2^4/b_2^4 \leq \lambda_2^4/3$. We choose an ε_0-balanced code $\mathcal{C}_0$ with support size $n = O(D/\varepsilon_0^c)$ where $c = 2.001$ (this choice of c is arbitrary, it is enough to have c as a fixed small constant) using the construction from Round II. It is crucial that we can unique decode $\mathcal{C}_0$ (using our algorithm), since this is required in order to apply the list decoding framework.

Note that ε_0 is no longer a constant. For this reason, we need to consider the rate computation of the final code $\mathcal{C}_\ell$ more carefully. The proof will follow an argument similar to Ta-Shma's.

Claim 4.8.10. *$\mathcal{C}_\ell$ has rate $\Omega(\varepsilon^{2+26\cdot\alpha})$ where $\alpha = \Theta(1/(\log_2(1/\varepsilon))^{1/6})$.*

Proof. The support size is the number of walks of length $t-1$ on the s-wide replacement product of G and H (each step of the walk has d_2^2 options), which is

$$\begin{aligned}
|V(G)||V(H)|d_2^{2(t-1)} = n' \cdot d_1^s \cdot d_2^{2(t-1)} &= n' \cdot d_2^{2(t-1)+4s} \leq n \cdot d_2^{2(t-1)+4s} \\
&= \Theta\left(\frac{D}{\varepsilon_0^c} \cdot d_2^{2(t-1)+4s}\right) \\
&= \Theta\left(D \cdot (d_2^2)^{(t-1)+2s+2.001}\right) \\
&= O\left(D \cdot (d_2^2)^{(1+2\alpha)(t-1)}\right).
\end{aligned}$$

From this point the proof continues exactly as the proof of Claim D.1.17. ∎

4.8.4 *Round IV: Arbitrary Gentle List Decoding*

In round III, when we take

$$s = \Theta((\log_2(1/\varepsilon))^{1/6}),$$

we will have $\lambda_2 = 4s\log(s^{4s^2})/s^{2s^2} \leq s^{-s^2}$ provided s is large enough. This non-constant λ_2 will allow us perform "gentle" list decoding with radius arbitrarily close to $1/2$. More precisely, we have the following.

Theorem 4.8.11 (Gentle List Decoding (restatement of Theorem 5.1.2))**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\beta} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\beta})$ where $\beta = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) a list decoding algorithm that decodes within radius $1/2 - 2^{-\Theta((\log_2(1/\varepsilon))^{1/6})}$ in time $N^{O_{\varepsilon,\beta}(1)}$.

Proof. We consider some parameter requirements in order to apply the list decoding framework Theorem 4.9.1 between $\mathcal{C}_{\ell-1}$ and $\mathcal{C}_\ell$. Suppose we want to list decode within radius $1/2 - \sqrt{\eta}$. For parity sampling, we need

$$s \geq \Theta(\log_2(1/\eta)).$$

Since the number of vertices in a walk can be at most s^2, for splittability we need

$$\eta^8/(s^2 \cdot 2^{2s^2}) \geq 2 \cdot s^{-s^2}.$$

In particular, we can take $\eta = 2^{-\Theta(s)}$ and satisfy both conditions above. ∎

4.9 Instantiating the List Decoding Framework

We established the tensoriality (actually two-step tensoriality) and parity sampling properties of every lifting between consecutive codes $\mathcal{C}_{i-1}$ and $\mathcal{C}_i$ in Ta-Shma's cascade. Using these properties, we will be able to invoke the list decoding framework from [AJQ+20] to obtain the following list decoding result.

Theorem 4.9.1 (Restatement of Theorem 4.6.1)**.** *Let $\eta_0 \in (0, 1/4)$ be a constant, $\eta \in (0, \eta_0)$, and*

$$k \geq k_0(\eta) := \Theta(\log(1/\eta)).$$

Suppose $\mathcal{C} \subseteq \mathbb{F}_2^n$ is an η_0-balanced linear code and $\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ is the direct sum lifting of $\mathcal{C}$ on a τ-splittable collection of walks $W(k)$, where $W(k)$ is either the set of walks $W[0, s]$ on an s-wide replacement product graph or a set of walks using the random walk operator $\mathsf{S}^{\triangle}_{r,r}$. There exists an absolute constant $K > 0$ such that if

$$\tau \leq \tau_0(\eta, k) := \frac{\eta^8}{K \cdot k \cdot 2^{4k}},$$

then the code $\mathcal{C}'$ is η-balanced and can be efficiently list decoded in the following sense:

If $\tilde{y}$ is $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}'$, then we can compute the list

$$\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') := \left\{ (z, \mathrm{dsum}_{W(k)}(z)) \mid z \in \mathcal{C}, \Delta\left(\mathrm{dsum}_{W(k)}(z), \tilde{y}\right) \leq \frac{1}{2} - \sqrt{\eta} \right\}$$

in time

$$n^{O(1/\tau_0(\eta,k)^4)} \cdot f(n),$$

where $f(n)$ is the running time of a unique decoding algorithm for $\mathcal{C}$. Otherwise, we return $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') = \emptyset$ with the same running time of the preceding case [9].

9. In the case $\tilde{y}$ is not $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}'$, but the SOS program turns out to be feasible, some of the

4.9.1 List Decoding Framework

We recall the precise statement of the list decoding framework tailored to direct sum lifting.

Theorem 4.9.2 (List Decoding Theorem (Adapted from [AJQ+20])). *Suppose* $\mathrm{dsum}_{W(k)}$ *is an* $(\eta^8/2^{30}, L)$*-two-step tensorial direct sum lifting from an* η_0*-balanced code* $\mathcal{C} \subseteq \mathbb{F}_2^n$ *to* $\mathcal{C}'$ *on a multiset* $W(k) \subseteq [n]^k$ *which is a* $(1/2 + \eta_0/2, \eta)$*-parity sampler.*

Let $\tilde{y} \in \mathbb{F}_2^{W(k)}$ *be* $(1/2 - \sqrt{\eta})$*-close to* $\mathcal{C}'$*. Then the List Decoding algorithm returns the coupled code list* $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}')$*. Furthermore, the running time is* $n^{O(L+k)}\,(\mathrm{polylog}(1/\eta) + f(n))$ *where* $f(n)$ *is the running time of an unique decoding algorithm of* $\mathcal{C}$*.*

We apply the list decoding framework of Theorem 4.9.2 to the liftings arising in the Ta-Shma cascade to obtain Theorem 4.9.1. This requires choosing parameters so that both the parity sampling and tensoriality requirements are met at every level of the cascade, which we do by appealing to our results from Section 4.7.

Proof of Theorem 4.9.1. We want to define parameters for τ-splittability so that $W(k)$ satisfies strong enough parity sampling and tensoriality assumptions to apply Theorem 4.9.2.

For parity sampling, we require $W(k)$ to be an $(1/2 + \eta_0/2, \eta)$-parity sampler. Suppose $W(k)$ is τ-splittable with $\tau < 1/16$. By Corollary 4.7.4 or Corollary 4.7.7 and splittability, the collection of walks $W(k)$ is an (η_0', η')-parity sampler, where $\eta' \leq (\eta_0' + 2\tau)^{\lfloor (k-1)/2 \rfloor}$. To achieve the desired parity sampling, we take $\eta_0' = 1/2 + \eta_0/2$ and choose a value of k large enough so that $\eta' \leq \eta$. Using the assumption $\eta_0 < 1/4$, we

calls to the unique decoding algorithm of $\mathcal{C}$ (issued by the list decoding framework) might be outside all unique decoding balls. Such cases may be handled by returning failure if the algorithm does not terminate by the time $f(n)$. Even if a codeword in $\mathcal{C}$ is found, the pruning step of list decoding [AJQ+20] will return an empty list for $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}')$ since $\tilde{y}$ is not $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}$.

compute

$$\eta' = (\eta_0' + 2\tau)^{\lfloor (k-1)/2 \rfloor} \leq (1/2 + \eta_0/2 + 2\tau)^{k/2-1} < (3/4)^{k/2-1},$$

which will be smaller than η as long as k is at least

$$k_0(\eta) = 2\left(1 + \frac{\log(1/\eta)}{\log(4/3)}\right) = \Theta(\log(1/\eta)).$$

Achieving this level of parity sampling also ensures that the lifted code $\mathcal{C}'$ is η-balanced.

The list decoding theorem also requires $(\eta^8/2^{30}, L)$-two-step tensoriality. Lemma 4.7.24 (with $s = k$) and Lemma 4.7.25 each provide (μ, L)-two-step tensoriality for τ-splittable walk collections on the s-wide replacement product and using $\mathsf{S}^{\triangle}_{r,r}$, respectively, with

$$L \geq \frac{128k^4 \cdot 2^{4k}}{\mu^4} \quad \text{and} \quad \tau \leq \frac{\mu}{4k \cdot 2^{4k}}.$$

To get $\mu = \eta^8/2^{30}$, we require

$$L \geq \frac{K' \cdot k^4 \cdot 2^{4k}}{\eta^{32}} \quad \text{and} \quad \tau \leq \tau_0(\eta, k) = \frac{\eta^8}{K \cdot k \cdot 2^{4k}},$$

where K and K' are (very large) constants. This ensures that τ is small enough for the parity sampling requirement as well. With these parameters, the running time for the list decoding algorithm in Theorem 4.9.2 becomes

$$n^{O(L+k)}(\text{polylog}(1/\eta) + f(n)) = n^{O(L)} \cdot f(n) = n^{O(1/\tau_0(\eta,k)^4)} \cdot f(n).$$

■

For decoding in fixed polynomial time, we also need a variation of list decoding

where we don't run the unique decoding algorithm of the base code and only obtain an approximate list of solutions. The proof is very similar to the proof of Theorem 4.9.1 above.

Theorem 4.9.3 (Restatement of Theorem 4.6.12)**.** *Let $\eta_0 \in (0, 1/4)$ be a constant, $\eta \in (0, \eta_0)$, $\zeta = 1/8 - \eta_0/8$, and*

$$k \geq k_0'(\eta) := \Theta(\log(1/\eta)).$$

Suppose $\mathcal{C} \subseteq \mathbb{F}_2^n$ is an η_0-balanced linear code and $\mathcal{C}' = \mathrm{dsum}_{W(k)}(\mathcal{C})$ is the direct sum lifting of $\mathcal{C}$ on a τ-splittable collection of walks $W(k)$, where $W(k)$ is either the set of walks $W[0, s]$ on an s-wide replacement product graph or a set of walks using the random walk operator $\mathsf{S}^{\triangle}_{r,r}$. There exists an absolute constant $K > 0$ such that if

$$\tau \leq \tau_0(\eta, k) := \frac{\eta^8}{K \cdot k \cdot 2^{4k}},$$

then the code $\mathcal{C}'$ is η-balanced, $W(k)$ is a $(1 - 2\zeta, \eta)$-parity sampler, and we have the following:

If $\tilde{y}$ is $(1/2 - \sqrt{\eta})$-close to $\mathcal{C}'$, then we can compute a ζ-cover $\mathcal{L}'$ of the list

$$\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}') := \left\{ (z, \mathrm{dsum}_{W(k)}(z)) \mid z \in \mathcal{C}, \Delta\left(\mathrm{dsum}_{W(k)}(z), \tilde{y}\right) \leq \frac{1}{2} - \sqrt{\eta} \right\}$$

in which $\Delta(y', \tilde{y}) \leq 1/2 - \sqrt{\eta}$ for every $(z', y') \in \mathcal{L}'$ [10], in time

$$n^{O(1/\tau_0(\eta,k)^4)}.$$

Otherwise, we return $\mathcal{L}' = \emptyset$ with the same running time of the preceding case.

Proof. The list decoding framework produces a cover $\mathcal{L}'$ of the list $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}')$, and, as its final step, corrects the cover to obtain the actual list $\mathcal{L}(\tilde{y}, \mathcal{C}, \mathcal{C}')$ by running the unique de-

10. A randomized rounding will ensure this, but see **??** for obtaining this property deterministically.

coding algorithm of $\mathcal{C}$ on each entry of $\mathcal{L}'$ (see [AJQ+20] for details). Using Theorem 4.9.2 with a $(1-2\zeta,\eta)$-parity sampler and omitting this final step of the algorithm, we can obtain the ζ-cover $\mathcal{L}'$ in time $n^{O(L+k)}\mathrm{polylog}(1/\eta)$.

The tensoriality part of the proof of Theorem 4.9.1 applies here unchanged, so we need only make sure k is large enough to yield the stronger parity sampling necessary for this theorem. As in that proof, we have that $W(k)$ is an (η_0',η')-parity sampler with $\eta' \leq (\eta_0'+2\tau)^{\lfloor (k-1)/2 \rfloor}$. Take $\eta_0' = 1-2\zeta = 3/4+\eta_0/4$. Using $\eta_0 < 1/4$ and assuming $\tau < 1/16$, we have

$$\eta' \leq (\eta_0'+2\tau)^{\lfloor (k-1)/2 \rfloor} \leq (3/4+\eta_0/4+2\tau)^{k/2-1} < (15/16)^{k/2-1},$$

which will be smaller than η as long as k is at least

$$k_0'(\eta) = 2\left(1+\frac{\log(1/\eta)}{\log(16/15)}\right) = \Theta(\log(1/\eta)).$$

■

CHAPTER 5

NEAR-LINEAR TIME DECODING OF TA-SHMA'S CODES VIA SPLITTABLE REGULARITY

5.1 Introduction

A binary code $\mathcal{C} \subseteq \mathbb{F}_2^N$ is said to be ε-balanced if any two distinct codewords $x, y \in \mathcal{C}$ satisfy $\Delta(x,y) \in [(1-\varepsilon)/2, (1+\varepsilon)/2]$, where $\Delta(x,y)$ denotes the relative distance between the two codewords. Finding explicit and optimal constructions of such codes, and indeed of codes where the distances are at least $(1-\varepsilon)/2$ is a central problem in coding theory [Gur10, Gur09], with many applications to the theory of pseudorandomness [Vad12]. Recently, Ta-Shma [TS17] gave a breakthrough construction of (a family of) explicit ε-balanced codes, with near-optimal rates, for arbitrarily small $\varepsilon > 0$. For the case of codes with distance at least $(1-\varepsilon)/2$, the existential rate-distance tradeoffs established by Gilbert [Gil52] and Varshamov [Var57], prove the existence of codes with rate $\Omega(\varepsilon^2)$, while McEliece et al. [MRRW77] prove an upper bound of $O(\varepsilon^2 \log(1/\varepsilon))$ on the rate. On the other hand, Ta-Shma's result yields an *explicit* family of codes with rate $\Omega(\varepsilon^{2+o(1)})$.

Decoding algorithms. The near-optimal ε-balanced codes of Ta-Shma [TS17] (which we will refer as Ta-Shma codes) were not known to be efficiently decodable at the time of their discovery. In later work, polynomial-time unique decoding algorithms for (a slight modification of) these codes were developed in [JQST20] (building on [AJQ+20]) using the Sum-of-Squares (SoS) hierarchy of semidefinite programming (SDP) relaxations. For unique decoding of codes with rates $\Omega(\varepsilon^{2+\alpha})$ (when $\alpha > 0$ is an arbitrarily small constant) these results yield algorithms running in time $N^{O_\alpha(1)}$. These algorithms also extend to the case when α is a vanishing function of ε, and to the problem of list decoding within an error radius of $1/2 - \varepsilon'$ (for ε' larger than a suitable function of ε) with running time

$N^{O_{\varepsilon,\varepsilon',\alpha}(1)}$. However, the $O_\alpha(1)$ exponent of N obtained in the unique decoding case is quite large even for a fixed constant α (say $\alpha = 0.1$), and the exponent in the list decoding case grows with the parameter ε.

In this work, we use a different approach based on new weak regularity lemmas (for structures identified by the SoS algorithms), resulting in near-linear time algorithms for both the above tasks. The algorithms below work in time $\tilde{O}_\varepsilon(N)$ for ε-balanced Ta-Shma codes with rates $\Omega(\varepsilon^{2+\alpha})$, even when α is a (suitable) vanishing function of ε.

Theorem 5.1.1 (Near-linear Time Unique Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\alpha} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\alpha})$ where $\alpha = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) an $r(\varepsilon) \cdot \tilde{O}(N)$ time unique decoding algorithm that that decodes within radius $1/4 - \varepsilon/4$ and works with high probability,

where $r(\varepsilon) = \exp(\exp(\mathrm{polylog}(1/\varepsilon)))$.

We can also obtain list decoding results as in [JQST20], but now in near-linear time.

Theorem 5.1.2 (Near-linear Time Gentle List Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\alpha} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\alpha})$ where $\alpha = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) an $r(\varepsilon) \cdot \tilde{O}(N)$ time list decoding algorithm that decodes within radius $1/2 - 2^{-\Theta((\log_2(1/\varepsilon))^{1/6})}$ and works with high probability,

where $r(\varepsilon) = \exp(\exp(\text{poly}(1/\varepsilon)))$.

While Theorem 5.1.2 yields a list decoding radius close to $1/2$, we remark that the above tradeoff between the list decoding radius and rate, is far from the state-of-the-art of $1/2 - \varepsilon$ radius with rate $\Omega(\varepsilon^3)$ of Guruswami and Rudra [GR06]. Considering a three way trade-off involving distance, rate, and list-decoding radius, Theorem 5.1.2 can be seen as close to optimal with respect to the first two parameters, and quite far off with respect to the third one. Finding an algorithm for codes with optimal tradeoffs in all three parameters, is a very interesting open problem. Another interesting problem is understanding the optimal dependence of the "constant" factors $r(\varepsilon)$ in the running times. We have not tried to optimize these factors in our work.

Direct-Sum Codes and "Structured Pseudorandomness". Ta-Shma's code construction can be viewed as a special case of "distance amplification via direct-sum", an operation with several applications in coding and complexity theory [ABN+92, IW97, GI01, IKW09, DS14, DDG+15, Cha16, DK17, Aro02]. Given a (say) linear code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ and a collection of tuples $W \subseteq [n]^k$, we define it's "direct-sum lifting" as $\mathcal{C} = \text{dsum}_W(\mathcal{C}_0) \subseteq \mathbb{F}_2^{|W|}$ where

$$\text{dsum}_W(\mathcal{C}_0) \;:=\; \left\{ (z_{i_1} + \cdots + z_{i_k})_{(i_1,\ldots,i_k)\in W} \;\mid\; z \in \mathcal{C}_0 \right\}.$$

It is easy to see that if $\mathcal{C}_0$ is ε_0-balanced for a constant ε_0, then taking $W = [n]^k$ results in $\text{dsum}_W(\mathcal{C}_0)$ being ε-balanced with $\varepsilon = \varepsilon_0^k$ (though with vanishing rate). A standard sampling argument shows that a random $W \subseteq [n]^k$ with $|W| = O(n/\varepsilon^2)$ also suffices, while yielding rate $\Omega(\varepsilon^2)$. Rozenman and Wigderson [Bog12] suggested a derandomization of this argument using a "pseudorandom" W constructed from iteratively considering the edges from larger and larger expanders graphs. While this result can be shown to achieve a rate of $\Omega(\varepsilon^{4+o(1)})$, Ta-Shma achieves a rate of $\Omega(\varepsilon^{2+o(1)})$ using a carefully constructed *sub-collection* of walks on an expander with a special form.

The above results show that pseudorandomness can be used to amplify distance, since the collections W above behave *like* a random W. However, finding decoding algorithms for such codes requires understanding properties of these collections which are *unlike* a random W, since random collections yield codes with (essentially) random generator matrices, where we do not expect efficient algorithms.

Our results can be viewed as showing that when the collection W satisfies a form of "structured multi-scale pseudorandomness" property [1] called *splittability* (identified in previous work), it can be exploited for algorithm design. One can think of splittability as capturing properties of the complete set $[n]^k$, which are not present in a (sparse) random $W \subseteq [n]^k$. For the case of $k = 4$, when $W = [n]^4$, if we consider a graph between pairs (i_1, i_2) and (i_3, i_4), which are connected when $(i_1, \ldots, i_4) \in W$, then this defines an expanding (complete) graph when $W = [n]^4$. On the other hand, for a random W of size $O(n)$, such a graph is a matching with high probability. Splittability requires various such graphs defined in terms of W to be expanders.

Definition 5.1.3 (Splittability, informal)**.** *Given $W \subseteq [n]^k$ and $a, b \in [k]$, let $W[a, b] \subseteq [n]^{b-a+1}$ denote the tuples obtained by considering $(i_a, \ldots, i_b)$ for every $(i_1, \ldots, i_k) \in W$. We say W can be τ-split at position t, if the bipartite graph with vertex sets $W[1, t]$ and $W[t+1, k]$, edge-set W, and (normalized) biadjacency matrix $\mathsf{S}_t \in \mathbb{R}^{W[1,t] \times W[t+1,k]}$, is an expander satisfying $\sigma_2(\mathsf{S}_t) \leq \tau$. We say that W is τ-splittable if for all $1 \leq a \leq t < b \leq k$, $W[a, b]$ can be τ-split at position t.*

Note that when $k = 2$, this coincides with the definition of (bipartite) graph expansion. It is also easy to show that collections of length-$(k-1)$ walks on a graph with second singular value λ, satisfy the above property with $\tau = \lambda$. The sub-collections used

1. As discussed later, there are several notions of "structured pseudorandom" for (ordered and unordered) hypergraphs. We describe splittability here, since this is the one directly relevant for our algorithmic applications.

by Ta-Shma can also be shown to splittable (after a a slight modification) and we recall this proof from [JQST20] in Appendix D.1.

The key algorithmic component in our decoding results, is a general *list decoding* result for codes constructed via direct-sum operations, which reduces the task of list decoding for $\mathrm{dsum}_W(\mathcal{C}_0)$ to that of unique decoding for the code $\mathcal{C}_0$, when W is τ-splittable for an appropriate τ. The splittability property was identified and used in previous work [AJQ+20, JQST20], for the analysis of SoS based algorithms, which obtained the above reduction in $N^{O_\varepsilon(1)}$ time. Regularity based methods also allow for near-linear time algorithms in this general setting of direct-sum codes, with a simpler and more transparent proof (and improved dependence of the list decoding radius on τ and k).

Theorem 5.1.4 (List Decoding Direct Sum (informal version of Theorem 5.5.1))**.** *Let $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ be an ε_0-balanced linear code, which is unique-decodable to distance $(1-\varepsilon_0)/4$ in time $\mathcal{T}_0$. Let $W \subseteq [n]^k$ be a τ-splittable collection of tuples. Let $\mathcal{C} = \mathrm{dsum}_W(\mathcal{C}_0)$ be ε-balanced, and let β be such that*

$$\beta \gg \max\left\{\sqrt{\varepsilon},\ \left(\tau \cdot k^3\right)^{1/2},\ \left(\frac{1}{2}+2\varepsilon_0\right)^{k/2}\right\}.$$

Then, there exists a randomized algorithm, which given $\tilde{y} \in \mathbb{F}_2^W$, recovers the list

$$\mathcal{L}_\beta(\tilde{y}) := \{y \in \mathcal{C} \mid \Delta(\tilde{y}, y) \leq 1/2 - \beta\},$$

with probability at least $1 - o(1)$, in time $\tilde{O}(C_{\beta,k,\varepsilon_0} \cdot (|W| + \mathcal{T}_0))$, where $C_{k,\beta,\varepsilon_0}$ only depends on k, β and ε_0.

Splittable Regularity. The technical component of our results is a novel understanding of splittable structures, via weak regularity lemmas. This provides a different way of exploiting "structured pseudorandomness" properties in hypergraphs, which may be of interest beyond applications considered here.

For the case of graphs (i.e., $k = 2$), several weak regularity lemmas are known which can be applied to (say) dense subgraphs of an expanding graph [RTTV08, TTV09, COCF09, BV20]. As in the Frieze-Kannan [FK96] weak regularity lemma for dense graphs, these lemmas decompose the adjacency matrix $A_{W'}$ of a subgraph $W' \subseteq W$, as a weighted sum of a small number of cut matrices ($\mathbf{1}_{S_\ell}\mathbf{1}_{T_\ell}^\top$ for $S_\ell, T_\ell \subseteq [n]$), such that one can use this decomposition to count the number of edges between any subsets $S, T \subseteq [n]$ i.e.,

$$\left| \mathbf{1}_S^\top \left(A_{W'} - \sum_\ell c_\ell \cdot \mathbf{1}_{S_\ell}\mathbf{1}_{T_\ell}^\top \right) \mathbf{1}_T \right| \;\leq\; \varepsilon \cdot |W| \;.$$

This can be thought of as computing an "approximation" of $A_{W'}$ using a small number of cut matrices $\mathbf{1}_{S_j}\mathbf{1}_{T_j}^\top$, which is "indistinguishable" by any cut matrix $\mathbf{1}_S\mathbf{1}_T^\top$.

More generally, one can think of the above results as approximating any function $g : W \to [-1, 1]$ (with $g = \mathbf{1}_{W'}$ in the example above) with respect to a family of "split" functions $\mathcal{F} \subseteq \{f : [n] \to [-1, 1]\}^{\otimes 2}$, where the approximation itself is a sum of a small number of of functions from $\mathcal{F}$ i.e., for all $f_1, f_2 \in \mathcal{F}$

$$\left| \left\langle g - \sum_\ell c_\ell \cdot f_{\ell,1} \otimes f_{\ell,2} \,, f_1 \otimes f_2 \right\rangle \right| \;\leq\; \varepsilon \cdot |W| \;.$$

Our regularity lemma for splittable $W \subseteq [n]^k$, extends the above notion of approximation, using k-wise split functions of the form $f_1 \otimes \cdots \otimes f_k$. We obtain near-linear time weak regularity decompositions for classes of k-wise cut functions of the form

$$\mathrm{CUT}^{\otimes k} := \{\pm \mathbf{1}_{S_1} \otimes \cdots \otimes \mathbf{1}_{S_k} \mid S_1, \ldots, S_k \subseteq [n]\},$$

and also for signed version of these k-wise cut functions

$$\mathrm{CUT}_{\pm}^{\otimes k} := \{\pm \chi_{S_1} \otimes \cdots \otimes \chi_{S_k} \mid S_1, \ldots, S_k \subseteq [n]\},$$

where $\chi_S = (-1)^{\mathbf{1}_S}$. For our decoding results, we will use $\mathrm{CUT}_{\pm}^{\otimes k}$. Our near-linear time weak regularity decomposition result is given next.

Theorem 5.1.5 (Efficient Weak Regularity (informal version of Theorem 5.4.11))**.** *Let $W \subseteq [n]^k$ and let $\mathcal{F}$ be either $\mathrm{CUT}^{\otimes k}$ or $\mathrm{CUT}_{\pm}^{\otimes k}$. Suppose $g \in \mathbb{R}^{[n]^k}$ is supported on W and has bounded norm. For every $\delta > 0$, if W is τ-splittable with $\tau = O(\delta^2/k^3)$, then we can find $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell$ in $\widetilde{O}_{k,\delta}(|W|)$ time, where $p = O(k^2/\delta^2)$, $f_\ell \in \mathcal{F}$ and $c_\ell \in \mathbb{R}$, such that h is a good approximator to g in the following sense*

$$\max_{f \in \mathcal{F}} \quad \langle g - h, f \rangle \;\leq\; \delta \cdot |W|\,,$$

where the inner product is over the counting measure on $[n]^k$.

We note that an existential version of the above theorem follows known abstract versions of the Frieze-Kannan regularity lemma [TTV09, BV20], via a relatively simple use of splittability. However, making a black-box application of known regularity lemmas algorithmic, requires computing a form of "tensor cut-norm", which is believed to be hard to even approximate in general[2] (unlike the matrix case). The nontrivial component of the result above, is obtaining a regularity lemma which allows for a *near-linear time computation*, while still achieving parameters close to the existential version.

Related Work. As discussed above, decoding results for Ta-Shma's codes were derived earlier using algorithms based on the SoS hierarchy [AJQ+20, JQST20]. The biggest advantage of the present work being the near optimal (i.e., near linear) dependence of the running time on block length of the code whereas this dependencce is at best a large

2. Strictly speaking, we only need to approximate this for "splittable" tensors. It is possible that one could use existing regularity lemmas black box, and use splittability to design a fast algorithm for tensor cut-norm. In our proof, we instead choose to use the matrix cut-norm algorithms as black-box, and use splittability to modify the proof of the regularity lemma.

polynomial function in [JQST20]. However, in some regimes the dependence of the decoding time on ε is polylogarithmic in [JQST20] wheares here it is super exponential. Therefore, our work and [JQST20] are incomparable. We will comment more on their difference at the end of this section. A common thread in the SoS algorithms is to relate the task of decoding, to that of solving instances of constraint satisfaction problems with k variables in each constraint (k-CSPs). The original weak regularity lemma of Frieze and Kannan [FK96] was indeed motivated by the question of approximately solving k-CSPs on dense structures (see also [KV09]). Several extensions of the Frieze-Kannan lemma are known, particularly for various families of sparse pseudorandom graphs [KR02, RTTV08, TTV09, OGT15, BV20]. Oveis-Gharan and Trevisan [OGT15] also proved a new weak regularity lemma for "low threshold-rank" graphs, which was used to obtain approximation algorithms for some 2-CSPs, where the previously known algorithms used the SoS hierarchy [BRS11, GS11]. Our work can be viewed as an extension of these ideas to the case of k-CSPs.

Ideas based on regularity lemmas, were also employed in the context of list decoding of Reed-Muller codes, by Bhowmick and Lovett [BL18]. They use analogues of the abstract weak regularity lemma [TTV09] and the Szemerédi regularity lemma over finite fields, but these are only used to prove bounds on the list size, rather than in the algorithm itself. On the other hand, our decoding algorithm crucially uses the decomposition obtained via our weak regularity lemma for (real-valued functions on) splittable structures.

In general, expansion phenomena have a rich history of interaction with coding theory (e.g., [GI01, Gur04, GI05, RWZ20]) including to the study of linear (or near-linear) time decoding backing to the seminal work of Sipser and Spielman [SS96]. The codes in [SS96] were good codes, though not near optimal in terms of distance-rate trade-off. Several other notions of "structured pseudorandomness" for hypergraphs (referred to as high-

dimensional expansion) have also been considered in literature, which also have connections to the decoding of good codes. In particular, the notion of "double sampler" was used to obtain algorithms for the list decoding for direct-product codes [DHK+19]. The notions of local spectral expansion [DK17], cosystolic expansion [EK16b], and multilayer agreement samplers [DDHRZ20], are also used to connect structured pseudorandomness to the design of locally testable codes. The notion of splittability was also studied for unordered hypergraphs in terms of "complement walks" by Dinur and Dikstein [DD19], and in terms of "swap walks" in [AJT19], for high-dimensional expanders defined via local spectral expansion. The concept of splittability also arose in the ealier work of Mossel [Mos10] when giving bounds to the expected value of products of low influence functions[3].

We now give more details on some of the differences between our work and [JQST20] in the case of unique decoding. For a Ta-Shma code of block length N, distance $1/2 - \varepsilon$ and rate $\Omega(\varepsilon^{2+\alpha})$ where $\alpha > 0$ quantifies how far we are from the Gilbert–Varshamov parameters[4], we will consider a few scenarios for the decoding time. The biggest advantage of the present work is a near linear dependence of the running time on the block length N, i.e., $\widetilde{O}(\exp(\exp(\mathrm{polylog}(1/\varepsilon))) \cdot N)$ time, whereas the decoders in [JQST20] take $O_\varepsilon(N^\gamma)$ with γ is at least a large constant (and in some cases γ grows with $1/\varepsilon$). More preceily, for constant $\alpha = O(1)$ their decoders take $O(\log(1/\varepsilon)^{O(1)} \cdot N^{O_\alpha(1)})$, and for $\alpha = \alpha(\varepsilon)$ they take $O(N^{\mathrm{poly}(1/\varepsilon)})$ time. On the other hand, our decoding times have a super exponential dependence on ε whereas this dependence can be polylogarithmic in [JQST20]. Roughly speaking, the use of Sum-of-Squares leads to a larger polynomial dependence on the block length while the use of a regularity based approach leads to large dependence on ε. It is an open problem to find a decoding algorithm having at the same time a linear

3. We thank the announymous reviewer for bringing the work [Mos10] to our attention.

4. In Ta-Shma's construction, the exponent α can be taken to be a constant or a suitable function of ε that vanish with ε.

or near-linear dependence on N and a polynomial dependence on ε.

5.2 A Technical Overview

We now give a more detailed overview of some of the technical components of our proof.

Splittability. The key structural property used for our algorithmic and structural results, is the "structured pseudorandomness" of ordered hypergraphs $W \subseteq [n]^k$, which we call *splittability*. The canonical example one can think of for this case, is a collection of all length-$(k-1)$ walks on a (say) d-regular expander graph G on n vertices. Note that this satisfies $|W[a,b]| = d^{b-a} \cdot n$, where $W[a,b]$ represents the collection of sub-tuples with coordinates between indices a and b i.e., portions of the walks between the a^{th} and b^{th} step. We will restrict our discussion in this paper only to *d-regular collections* $W \subseteq [n]^k$ satisfying $|W[a,b]| = d^{b-a} \cdot n$.

We briefly sketch why the collection of length-3 walks (i.e., the case $k = 4$) is splittable. Recall that splittability requires various graphs with sub-tuples to be expanding, and in particular consider the graph between $W[1,2]$ and $W[3,4]$, with edge-set $W[1,4]$. If $E(G)$ is the set of edges in G included with both orientations, then note that $W[1,2] = W[3,4] = E(G)$, and $(i_1, i_2), (i_3, i_4)$ are connected iff $(i_2, i_3) \in E(G)$. If $M \in \mathbb{R}^{W[1,2] \times W[3,4]}$ denotes the biadjacency matrix of the bipartite graph H on $W[1,2] \times W[3,4]$, then up to permutations of rows and columns, we can write M as $\mathsf{A}_G \otimes \mathsf{J}_d / d$, where J_d denotes the $d \times d$ all-1s matrix and A_G is the normalized adjacency matrix of G, since each tuple $(i_2, i_3) \in E(G)$ contributes d^2 edges in H (for choices of i_1 and i_4). Thus $\sigma_2(M) = \sigma_2(\mathsf{A}_G)$, which is small if G is an expander. A similar argument also works for splits in other positions, and for longer walks.

The above argument can also be extended to show that the sub-collections of walks

considered by Ta-Shma (after a slight modification) are splittable, though the structure and the corresponding matrices are more involved there (see Appendix D.1).

Regularity for graphs and functions. We first consider an analytic form of the Frieze-Kannan regularity lemma (based on [TTV09]). Let $g : \mathcal{X} \to [-1,1]$ be any function on a finite space $\mathcal{X}$ with an associated probability measure μ, and let $\mathcal{F} \subseteq \{f : \mathcal{X} \to [-1,1]\}$ be any class of functions closed under negation. Say we want to construct a "simple approximation/decomposition" h, which is indistinguishable from g, for all functions in f i.e.,

$$\langle g - h, f \rangle_\mu \;=\; \mathop{\mathbb{E}}_{x \sim \mu} [(g(x) - h(x)) \cdot f(x)] \;\leq\; \delta \qquad \forall f \in \mathcal{F}\,.$$

We can view the regularity lemma as saying that such an h can always be constructed as a sum of $1/\delta^2$ functions from $\mathcal{F}$. Indeed, we can start with $h^{(0)} = 0$, and while there exists f_ℓ violating the above condition, we update $h^{(\ell+1)} = h^{(\ell)} + \delta \cdot f_\ell$. The process must stop in $1/\delta^2$ steps, since $\|g - h^{(\ell)}\|^2$ can be shown to decrease by δ^2 in every step.

$$\left\|g - h^{(\ell)}\right\|_\mu^2 - \left\|g - h^{(\ell+1)}\right\|_\mu^2 \;=\; 2\delta \cdot \left\langle g - h^{(\ell)}, f_\ell \right\rangle_\mu - \delta^2 \cdot \|f_\ell\|_\mu^2 \;\geq\; \delta^2\,.$$

In fact, the above can be seen as gradient descent for minimizing the convex function $F(h) = \sup_{f \in \mathcal{F}} \langle g - h, f \rangle_\mu$. Taking $\mathcal{X} = [n]^2$ with μ as uniform on $[n]^2$, $g = \mathbf{1}_{E(G)}$ for a (dense) graph G, and $\mathcal{F}$ as all functions (cut matrices) of the form $\pm \mathbf{1}_S \mathbf{1}_T^\mathsf{T}$ yields the weak regularity lemma for graphs, since we get $h = \sum_\ell c_\ell \cdot f_\ell = \sum_\ell c_\ell \cdot \mathbf{1}_{S_\ell} \mathbf{1}_{T_\ell}^\mathsf{T}$ such that

$$\langle g - h, f \rangle_\mu \leq \delta \quad \forall f \in \mathcal{F} \quad \Leftrightarrow \quad \frac{1}{n^2} \cdot \left| E_G(S,T) - \sum_\ell c_\ell \, |S_\ell \cap S| \, |T_\ell \cap T| \right| \leq \delta \quad \forall S, T \subseteq [n]\,.$$

Note that the inner product in the above analytic argument can be chosen to be according to any measure on $\mathcal{X}$, and not just the uniform measure. In particular, taking $W \subseteq [n]^2$ to be the edge-set of a (sparse) d-regular expander with second singular value (say) λ,

and $\mu = \mu_2$ to be uniform over W, we obtain the regularity lemma for subgraphs of expanders. In this case, after obtaining the approximation with respect to μ, one shows using the expander mixing lemma that if $\langle g - h, f \rangle_{\mu_2} \leq \delta$, then $\langle g - (d/n) \cdot h, f \rangle_{\mu_1 \otimes \mu_1} \leq (d/n) \cdot \delta'$, where μ_1 denotes the uniform measure on $[n]$ and $\delta' = \delta + \lambda$. This gives a sparse regularity lemma, since for $G \subseteq W$ and $g = \mathbf{1}_G$,

$$\left\langle g - \left(\frac{d}{n}\right) h, f \right\rangle_{\mu_1^{\otimes 2}} \leq \frac{d}{n} \cdot \delta' \quad \forall f \in \mathcal{F} \quad \Leftrightarrow \quad \left| E_G(S,T) - \sum_\ell c_\ell \cdot \frac{d}{n} |S_\ell \cap S| \, |T_\ell \cap T| \right| \leq \delta' \cdot nd \quad \forall S, T \,.$$

The *algorithmic step* in the above proofs, is finding an f_ℓ such that $\langle g - h, f_\ell \rangle > \delta$. For the function class $\mathcal{F}$ corresponding to cut matrices, this corresponds to solving a problem of the form $\max_{S,T} \left| \mathbf{1}_S^\mathsf{T} M \mathbf{1}_T \right|$ for an appropriate matrix M at each step. This equals the cut-norm and can be (approximately) computed using the SDP approximation algorithm of Alon and Naor [AN04]. Moreover, this can be implemented in near-linear time in the sparsity of M, using known fast, approximate SDP solvers of Lee and Padmanabhan [LP20] or of Arora and Kale [AK07] (see Section 5.4.5 for details).

Splittable regularity. For our regularity lemma, the class $\mathcal{F}$ comprises of "k-split functions" of the form $f_1 \otimes \cdots \otimes f_k$, where for each f_t can be thought of as $\mathbf{1}_{S_t}$ (or $(-1)^{\mathbf{1}_{S_t}}$) for some $S_t \subseteq [n]$. An argument similar to the one above, with the measure μ_k uniform on $W \subseteq [n]^k$, can yield an *existential version* of the splittable regularity lemma, similar to the one for expander graphs (we now transition from μ_k to $\mu_1^{\otimes k}$ using a simple generalization of the expander mixing lemma to splittable collections). However, the algorithmic step in the above procedure, requires computing

$$\max_{f_1, \ldots, f_k \in \mathcal{F}} \langle g - h, f_1 \otimes \cdots \otimes f_k \rangle$$

Unfortunately, such an algorithmic problem is hard to even approximate in general, as opposed to the 2-split case for graphs. Another approach is to first compute an approxi-

mation of a given $g : W \to [-1,1]$, in terms of 2-split functions of the form $f_1 \otimes f_2$, where $f_1 : W[1,t] \to [-1,1]$ and $f_2 : W[t+1,k] \to [-1,1]$, and then inductively approximate f_1 and f_2 in terms of 2-split functions, and so on. Such an induction does yield an algorithmic regularity lemma, though naively approximating the component functions f_1 and f_2 at each step, leads to a significantly lossy dependence between the final error, the splittability parameter τ, and k.

We follow a hybrid of the two approaches above. We give an inductive argument, which at step t, approximates g via h_t which is a sum of t-split functions. However, instead of simply applying another 2-split to each term in the decomposition h_t to compute h_{t+1}, we build an approximation for *all of* h_t using the regularity argument above from scratch. We rely on the special structure of h_t to solve the algorithmic problem $\max_{f_1,\ldots,f_{t+1}} \langle h_t - h_{t+1}, f_1 \otimes \cdots \otimes f_{t+1} \rangle$, reducing it to a matrix cut-norm computation[5]. This yields near-optimal dependence of the error on τ and k, needed for our coding applications.

Decoding direct-sum codes using regularity. We now consider the problem of decoding, from a received, possibly corrupted, $\tilde{y} \in \mathbb{F}_2^W$, to obtain the closest $y \in \mathrm{dsum}_W(\mathcal{C}_0)$ (or a list) i.e., finding $\mathrm{argmin}_{z_0 \in \mathcal{C}_0} \Delta(\tilde{y}, \mathrm{dsum}_W(z_0))$. Let $g : [n]^k \to \{-1,1\}$ be defined as $g(i_1,\ldots,i_k) = (-1)^{\tilde{y}(i_1,\ldots,i_k)}$ if $(i_1,\ldots,i_k) \in W$ and 0 otherwise. Also, for any $z \in \mathbb{F}_2^n$, define the function χ_z as $\chi_z(i) = (-1)^{z_i}$. As before, let μ_1 denote the uniform measure on

5. Strictly speaking, we also need to be careful about the bit-complexity of our matrix entries, to allow for near-linear time computation. However, all the entries in matrices we consider will have bit-complexity $O_{k,\delta}(\log n)$.

$[n]$. Using that g is 0 outside W, and that $|W| = d^{k-1} \cdot n$, we get

$$\begin{aligned} 1 - 2 \cdot \Delta(\tilde{y}, \mathrm{dsum}_W(z)) &= \mathop{\mathbb{E}}_{(i_1,\ldots,i_k)\in W} [g(i_1,\ldots,i_k) \cdot \chi_z(i_1) \cdots \chi_z(i_k)] \\ &= \left(\frac{n}{d}\right)^{k-1} \cdot \mathop{\mathbb{E}}_{(i_1,\ldots,i_k)\in [n]^k} [g(i_1,\ldots,i_k) \cdot \chi_z(i_1) \cdots \chi_z(i_k)] \\ &= \left(\frac{n}{d}\right)^{k-1} \cdot \left\langle g, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} . \end{aligned}$$

At this point, we modify the problem in three ways. First, instead of restricting the optimization to $z_0 \in \mathcal{C}_0$, we widen the search to all $z \in \mathbb{F}_2^n$. We will be able to show that because of the pseudorandom (distance amplification) properties of W, a good (random) solution z found by our algorithm, will be within the unique decoding radius of $\mathcal{C}_0$ (with high probability). Secondly, using the fact that for splittable W, the function g has an approximation $h = \sum_{\ell=1}^{p} c_\ell \cdot f_{\ell,1} \otimes \cdots \otimes f_{\ell,k}$ given by the regularity lemma, we can restrict our search to z which (approximately) maximize the objective

$$\left\langle h, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} = \sum_{\ell=1}^{p} c_\ell \cdot \prod_{t\in[k]} \left\langle f_{\ell,t}\, , \chi_z \right\rangle_{\mu_1}$$

Finally, instead of searching for $\chi_z : [n] \to \{-1, 1\}$, we further widen the search to $\overline{f} : [n] \to [-1, 1]$. A random "rounding" choosing each $\chi_z(i)$ independently so that $\mathbb{E}[\chi_z] = \overline{f}$ should preserve the objective value with high probability. We now claim that the resulting search for functions $\overline{f}$ maximizing $\left\langle h, \overline{f}^{\otimes k} \right\rangle_{\mu_1^{\otimes k}}$, can be solved via a simple brute-force search. Note that the objective only depends on the inner products with a finite number of functions $\{f_{\ell,t}\}_{\ell\in[p],t\in[k]}$ with range $\{-1, 1\}$. Partitioning the space $[n]$ in 2^{pk} "atoms" based on the values of these functions, we can check that it suffices to search over $\overline{f}$, which are constant on each atom. Moreover, it suffices to search the values in each atom, up to an appropriate discretization η, which can be done in time $O\left((1/\eta)^{2^{pk}}\right)$.

For the problem of list decoding $\tilde{y}$ up to radius $1/2 - \beta$, we show that each $z_0 \in$

$\mathcal{C}_0$, such that $\mathrm{dsum}_W(z_0)$ is in the list, there must be an $\overline{f}$ achieving a large value of $\left\langle h, \overline{f}^{\otimes k} \right\rangle_{\mu_1^{\otimes k}}$ which then yields a z within the unique decoding radius of z_0. Since we enumerate over all $\overline{f}$, this recovers the entire list. Details of the decoding algorithm are given in Section 5.5.

5.3 Preliminaries

We now introduce some notation. The asymptotic notation $\widetilde{O}(r(n))$ hides polylogarithmic factors in $r(n)$.

5.3.1 Codes

We briefly recall some standard code terminology. Given $z, z' \in \mathbb{F}_2^n$, recall that the relative Hamming distance between z and z' is $\Delta(z, z') := \left|\{i \mid z_i \neq z'_i\}\right| / n$. A binary code is any subset $\mathcal{C} \subseteq \mathbb{F}_2^n$. 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 if $\mathcal{C}$ is a linear subspace of $\mathbb{F}_2^n$. The rate of $\mathcal{C}$ is $\log_2(|\mathcal{C}|)/n$, or equivalently $\dim(\mathcal{C})/n$ if $\mathcal{C}$ is linear.

Definition 5.3.1 (Bias)**.** *The* bias *of a word $z \in \mathbb{F}_2^n$ is defined as* $\mathrm{bias}(z) := \left|\mathbb{E}_{i \in [n]}(-1)^{z_i}\right|$. *The bias of a code $\mathcal{C}$ is the maximum bias of any non-zero codeword in $\mathcal{C}$.*

Definition 5.3.2 (ε-balanced Code)**.** *A binary code $\mathcal{C}$ is* ε-balanced *if* $\mathrm{bias}(z + z') \leq \varepsilon$ *for every pair of distinct $z, z' \in \mathcal{C}$.*

Remark 5.3.3. *For linear binary code $\mathcal{C}$, the condition $\mathrm{bias}(\mathcal{C}) \leq \varepsilon$ is equivalent to $\mathcal{C}$ being an ε-balanced code.*

5.3.2 Direct Sum Lifts

Starting from a code $\mathcal{C} \subseteq \mathbb{F}_2^n$, we amplify its distance by considering the *direct sum lifting* operation based on a collection $W(k) \subseteq [n]^k$. The direct sum lifting maps each codeword of $\mathcal{C}$ to a new word in $\mathbb{F}_2^{|W(k)|}$ by taking the k-XOR of its entries on each element of $W(k)$.

Definition 5.3.4 (Direct Sum Lifting)**.** *Let $W(k) \subseteq [n]^k$. For $z \in \mathbb{F}_2^n$, we define the* direct sum lifting *as $\mathrm{dsum}_{W(k)}(z) = y$ such that $y_{(i_1,\ldots,i_k)} = \sum_{j=1}^{k} z_{i_j}$ for all $(i_1, \ldots, i_k) \in W(k)$. The direct sum lifting of a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ is*

$$\mathrm{dsum}_{W(k)}(\mathcal{C}) = \{\mathrm{dsum}_{W(k)}(z) \mid z \in \mathcal{C}\}.$$

We will omit $W(k)$ from this notation when it is clear from context.

Remark 5.3.5. *We will be concerned with collections $W(k) \subseteq [n]^k$ arising from length-$(k-1)$ walks on expanding structures (mostly arising from Ta-Shma's direct sum construction [TS17]).*

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 5.3.6 (Parity Sampler)**.** *A collection $W(k) \subseteq [n]^k$ is called an $(\varepsilon_0, \varepsilon)$*-parity sampler *if for all $z \in \mathbb{F}_2^n$ with $\mathrm{bias}(z) \leq \varepsilon_0$, we have $\mathrm{bias}(\mathrm{dsum}_{W(k)}(z)) \leq \varepsilon$.*

5.3.3 Splittable Tuples

We now formally define the *splittability* property for a collection of tuples $W(k) \subseteq [n]^k$. For $1 \leq a \leq b \leq k$, we define $W[a,b] \subseteq [n]^{(b-a+1)}$ as

$$W[a,b] := \{(i_a, i_{a+1}, \ldots, i_b) \mid (i_1, i_2, \ldots, i_k) \in W(k)\},$$

and use $W[a]$ to stand for $W[a,a]$. We will work with d-regular tuples in the following sense.

Definition 5.3.7 (Regular tuple collection)**.** *We say that $W(k) \subseteq [n]^k$ is d-regular if for every $1 \le a \le b \le k$, we have*

- $|W[a,b]| = d^{b-a} \cdot n$,
- $W[a] = [n]$.

A collection $W(k)$ being d-regular is analogous to a graph being d-regular.

Example 5.3.8. *The collection $W(k)$ of all length-$(k-1)$ walks on a d-regular connected graph $G = ([n], E)$ is a d-regular collection of tuples.*

The space of functions $\mathbb{R}^{W[a,b]}$ is endowed with an inner product associated to the uniform measure $\mu_{[a,b]}$ on $W[a,b]$. We use the shorthand μ_b for $\mu_{[1,b]}$.

Definition 5.3.9 (Splitable tuple collection)**.** *Let $\tau > 0$. We say that a collection $W(k) \subseteq [n]^k$ is* τ-splittable *if it is d-regular and either $k = 1$ or for every $1 \le a \le t < b \le k$ we have*

- *the* split *operator $\mathsf{S}_{W[a,t],W[t+1,b]} \in \mathbb{R}^{W[a,t] \times W[t+1,b]}$ defined as*

$$\left(\mathsf{S}_{W[a,t],W[t+1,b]}\right)_{(i_a,\ldots,i_t),(i_{t+1},\ldots,i_b)} := \frac{\mathbf{1}\left[(i_a,\ldots,i_t,i_{t+1},\ldots i_b) \in W[a,b]\right]}{d^{k-t}}$$

satisfy $\sigma_2(\mathsf{S}_{W[a,t],W[t+1,b]}) \le \tau$, where σ_2 denotes the second largest singular value.

Example 5.3.10. *The collection $W(k)$ of all length-$(k-1)$ walks on a d-regular a graph $G = ([n], E)$ whose normalized adjacency matrix has second largest singular value at most τ is a collection of τ-splittable tuples as shown in [AJQ+20].*

Example 5.3.11. *The collection $W(k)$ of tuples arising (from a slight modification) of the direct sum construction of Ta-Shma [TS17] is a τ-splittable as shown in [JQST20]. Precise parameters are recalled later as Theorem D.1.1 of Appendix D.1.*

5.3.4 Factors

It will be convenient to use the language of factors, to search the decompositions identified by regularity lemmas, for relevant codewords. This concept (from ergodic theory) takes a rather simple form in our finite settings: it is just a partition of base set $\mathcal{X}$, with an associated operation of averaging functions defined on $\mathcal{X}$, separately over each piece.

Definition 5.3.12 (Factors and measurable functions)**.** *Let $\mathcal{X}$ be a finite set. A* factor *$\mathcal{B}$ is a partition of the set $\mathcal{X}$, and the subsets of the partition are referred to as* atoms of the factor*.* A function $f : \mathcal{X} \to \mathcal{R}$ is said to measurable with respect to $\mathcal{B}$ ($\mathcal{B}$-measurable) if f is constant on each atom of $\mathcal{B}$.

Definition 5.3.13 (Conditional averages)**.** *If $f : \mathcal{X} \to \mathbb{R}$ is a function, μ is a measure on the space $\mathcal{X}$, and $\mathcal{B}$ is a factor, then we define the* conditional average function *$\mathbb{E}[f|\mathcal{B}]$ as*

$$\mathbb{E}\left[f|\mathcal{B}\right](x) \;:=\; \mathop{\mathbb{E}}_{y\sim\mu|\mathcal{B}(x)}\left[f(y)\right] ,$$

where $\mathcal{B}(x)$ denotes the atom containing x. Note that the function $\mathbb{E}[f|\mathcal{B}]$ is measurable with respect to $\mathcal{B}$.

We will need the following simple observation regarding conditional averages.

Proposition 5.3.14. *Let $h : \mathcal{X} \to \mathbb{R}$ be a $\mathcal{B}$-measurable function, and let $f : \mathcal{X} \to \mathbb{R}$ be any function. Then, for any measure μ over $\mathcal{X}$, we have*

$$\left\langle h, f\right\rangle_{\mu} \;=\; \left\langle h, \mathbb{E}\left[f|\mathcal{B}\right]\right\rangle_{\mu} .$$

Proof. By definition of the $\mathcal{B}$-measurability, h is constant on each atom, and thus we can

write $h(x)$ as $h(\mathcal{B}(x))$.

$$\begin{aligned}\langle h, f\rangle_\mu &= \mathop{\mathbb{E}}_{x\sim\mu}[h(x)\cdot f(x)] = \mathop{\mathbb{E}}_{x\sim\mu}\mathop{\mathbb{E}}_{y\sim\mu|\mathcal{B}(x)}[h(y)\cdot f(y)] \\ &= \mathop{\mathbb{E}}_{x\sim\mu}\left[h(\mathcal{B}(x))\cdot \mathop{\mathbb{E}}_{y\sim\mu|\mathcal{B}(x)}[f(y)]\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mu}[h(x)\cdot \mathbb{E}[f|\mathcal{B}](x)] = \langle h, \mathbb{E}[f|B]\rangle_\mu . \qquad \blacksquare\end{aligned}$$

The factors we will consider will be defined by a finite collection of functions appearing in a regularity decomposition.

Definition 5.3.15 (Function factors)**.** *Let $\mathcal{X}$ and $\mathcal{R}$ be finite sets, and let $\mathcal{F}_0 = \{f_1, \ldots, f_r : \mathcal{X} \to \mathcal{R}\}$ be a finite collection of functions. We consider the factor $\mathcal{B}_{\mathcal{F}_0}$ defined by the functions in $\mathcal{F}_0$, as the factor with atoms $\{x \mid f_1(x) = c_1, \ldots, f_r(x) = c_r\}$ for all $(c_1, \ldots, c_r) \in \mathcal{R}^r$.*

Remark 5.3.16. *Note that when the above function are indicators for sets i.e., each $f_j = \mathbf{1}_{S_j}$ for some $S_j \subseteq \mathcal{X}$, then the function factor $\mathcal{B}_{\mathcal{F}_0}$ is the same as the σ-algebra generated by these sets*[6]*. Also, given the functions $f_1, \ldots, f_r$ as above, the function factor $\mathcal{B}_{\mathcal{F}_0}$ can be computed in time $O(|\mathcal{X}| \cdot |\mathcal{R}|^r)$.*

5.3.5 Functions and Measures

We describe below some classes of functions, and spaces with associated measures, arising in our proof. The measures we consider are either uniform on the relevant space, or are products of measures on its component spaces.

6. For a finite $\mathcal{X}$, the σ-algebra generated by $S_1, \ldots, S_p \subseteq \mathcal{X}$ is the smallest subset of the power set of $\mathcal{X}$ containing $\mathcal{X}$, $S_1, \ldots, S_p$ that is closed under union, intersection and complement. This finite version will be enough for us in this work (see [Bil95] for the general definition).

Function classes. Let $S \subseteq [n]$. We define $\chi_S \colon [n] \to \{\pm 1\}$ as $\chi_S(i) \coloneqq (-1)^{\mathbf{1}_{i \in S}}$ (we observe that as defined χ_S is not a character[7]). We need the following two collection of functions for which algorithmic results will be obtained.

Definition 5.3.17 (CUT functions)**.** *We define the set of* $0/1$ CUT *cut functions as*

$$\mathrm{CUT}^{\otimes k} \coloneqq \{\pm \mathbf{1}_{S_1} \otimes \cdots \otimes \mathbf{1}_{S_k} \mid S_1, \ldots, S_k \subseteq [n]\},$$

and defined the set of ± 1 CUT *functions as*

$$\mathrm{CUT}_{\pm}^{\otimes k} \coloneqq \{\pm \chi_{S_1} \otimes \cdots \otimes \chi_{S_k} \mid S_1, \ldots, S_k \subseteq [n]\}.$$

We will use a higher-order version of cut norm.

Definition 5.3.18. *Let* $g \in \mathbb{R}^{[n]^k}$*, the k-tensor cut norm is*

$$\|g\|_{\square^{\otimes k}} \coloneqq \max_{f \in \mathrm{CUT}^{\otimes k}} \langle g, f \rangle ,$$

where the inner product is over the counting measure on $[n]^k$*.*

Some of our results hold for more general class of functions.

Definition 5.3.19 (t-split functions)**.** *Suppose* $W(k)$ *is a regular collection of k-tuples. For* $t \in \{0, \ldots, k-1\}$*, we define a generic class of tensor product functions* $\mathcal{F}_t$ *as*

$$\mathcal{F}_t \subseteq \left\{ \pm f_1 \otimes \cdots \otimes f_t \otimes f_{t+1} \mid f_j \in \mathbb{R}^{W[1]} \text{ for } j \leq t, f_{t+1} \in \mathbb{R}^{W[t+1,k]}, \left\|f_j\right\|_\infty \leq 1 \text{ for } j \leq t+1 \right\}.$$

To avoid technical issues, we assume that each $\mathcal{F}_t$ *is finite.*

7. Strictly speaking χ_S is not a character but by identifying the elements of $[n]$ with those of a canonical basis of $\mathbb{F}_2^n$ it becomes a character for $\mathbb{F}_2^n$.

Fixing some $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$, we define the set of functions that are linear combinations of function from $\mathcal{F}$ with coefficients of bounded support size and bounded ℓ_1-norm as follows

$$\mathcal{H}(R_0, R_1, \mathcal{F}) \coloneqq \left\{ \sum_{\ell=1}^{p} c_\ell \cdot f_\ell \mid p \leq R_0, \sum |c_\ell| \leq R_1, f_\ell \in \mathcal{F} \right\}.$$

Measures and inner products. Recall that $\mu_1 \coloneqq \mu_{[1,1]}$ is the uniform measure on $W[1]$ (equivalently uniform measure on $W[i]$ since $W(k)$ is regular) and $\mu_{[t+1,k]}$ is the uniform measure on $W[t+1,k]$. We define following measure ν_t as

$$\nu_t \coloneqq (\mu_1)^{\otimes t} \otimes \left(\mu_{[t+1,k]}\right).$$

Note that ν_0 is the equal to μ_k and ν_{k-1} is equal to $\mu_1^{\otimes k}$. We will need to consider inner products of functions according to various measures defined above, which we will denote as $\langle \cdot, \cdot \rangle_\mu$ for the measure μ. When a measure is not indicated, we take the inner product $\langle f, g \rangle$ to be according to the counting measure on the domains of the functions f and g.

5.4 Weak Regularity for Splittable Tuples

We will show how functions supported on a (possibly) sparse splittable collection of tuples $W(k) \subseteq [n]^k$ admit weak regular decompositions in the style of Frieze and Kannan [FK96]. In Section 5.4.1, we start by showing an abstract regularity lemma for functions that holds in some generality and does not require splittability. Next, in Section 5.4.2, we show that splittable collections of tuples satisfy suitable (simple) generalizations of the expander mixing lemma for graphs which we call splittable mixing lemma. By combining this abstract weak regularity decomposition with splittable mixing lemmas, we obtain *existential* decomposition results for splittable tuples in Section 5.4.3. Then, we proceed to make these existential results not only algorithmic but near-linear time computable

in Section 5.4.4. These algorithmic results will rely on fast cut norm like approximation algorithms tailored to our settings and this is done in Section 5.4.5. As mentioned previously, this last step borrows heavily from known results [AN04, AK07, LP20].

5.4.1 *Abstract Weak Regularity Lemma*

We now show a weak regularity decomposition lemma for functions that works in some generality and does not require splittability. We now fix some notation for this section. Let $\mathcal{X}$ be a finite set endowed with a probability measure μ. Let $\mathbb{R}^{\mathcal{X}}$ be a Hilbert space endowed with inner product $\langle f, g\rangle_\mu = \mathbb{E}_\mu[f \cdot g]$ and associated norm $\|\cdot\|_\mu = \sqrt{\langle \cdot, \cdot\rangle_\mu}$. Let $\mathcal{F} \subseteq \{f \colon \mathcal{X} \to \mathbb{R} \mid \|f\|_\mu \leq 1\}$ be a finite collection of functions such that $\mathcal{F} = -\mathcal{F}$.

In a nutshell, given any $g \in \mathbb{R}^{\mathcal{X}}$, the abstract weak regularity lemma will allow us to find an approximator h, with respect to the semi-norm[8] $g - h \mapsto \max_{f\in\mathcal{F}} \langle g - h, f\rangle$, which is a linear combinations of a certain *small* number of functions from $\mathcal{F}$ (where this number depends only on the approximation accuracy and the norm $\|g\|_\mu$). This means that g and h have approximately the same correlations with functions from $\mathcal{F}$. We will produce h in an iterative procedure, where at each step an oracle of the following kind (*cf.*, Definition 5.4.1) is invoked.

Definition 5.4.1 (Correlation Oracle)**.** *Let $1 \geq \delta \geq \delta' > 0$ be accuracy parameters and $B > 0$. We say that $\mathcal{O}_{\mu,B}$ is a (δ, δ')-correlation oracle for $\mathcal{F}$ if given $h \in \mathbb{R}^{\mathcal{X}}$ with $\|h\|_\mu^2 = O(B)$ if there exists $f \in \mathcal{F}$ with $\langle h, f\rangle \geq \delta$, then $\mathcal{O}_{\mu,B}$ returns some $f' \in \mathcal{F}$ with $\langle h, f'\rangle \geq \delta'$.*

More precisely, our abstract weak regularity decomposition is as follows.

Lemma 5.4.2 (Abstract Weak Regularity)**.** *Let $\mathcal{O}_{\mu,B}$ be a (δ, δ')-correlation oracle for $\mathcal{F}$ with $\delta \geq \delta' > 0$. Let $g \colon \mathcal{X} \to \mathbb{R}$ satisfy $\|g\|_\mu^2 \leq B$. Then, we can find $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell \in$*

8. See [Rud91, Chapter 1] for a definition of semi-norm.

$\mathcal{H}(B/(\delta')^2, B/\delta', \mathcal{F})$ *with* $f_\ell \in \mathcal{F}$, $c_\ell \in [\delta'/(1+\delta'/\sqrt{B})^p, \delta']$ *and* $\|h\|_\mu^2 \le B$ *such that*

$$\max_{f \in \mathcal{F}} \quad \langle g - h, f \rangle_\mu \;\le\; \delta.$$

Furthermore, if $\mathcal{O}_{\mu,B}$ *runs in time* $\mathcal{T}_{\mathcal{O}_{\mu,B}}$, *then* h *can be computed in*

$$\widetilde{O}\left(\mathrm{poly}(B, 1/\delta') \cdot (\mathcal{T}_{\mathcal{O}_{\mu,B}} + |\mathrm{Supp}(\mu)|)\right)$$

time, where $\mathrm{Supp}(\mu)$ *is the support of* μ. *The function* h *is constructed in Algorithm 5.4.3 as the final function in a sequence of approximating functions* $h^{(\ell)} \in \mathcal{H}(B/(\delta')^2, B/\delta', \mathcal{F})$.

The proof is based on the following algorithm.

Algorithm 5.4.3 (Regularity Decomposition Algorithm)**.**

Input $g\colon \mathcal{X} \to \mathbb{R}$

Output $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell$

- *Let* Π *be the projector onto the convex ball* $\{g' \in \mathbb{R}^{\mathcal{X}} \mid \|g'\|_\mu^2 \le B\}$.
- *Let* $\ell = 0$ *and* $h^{(\ell)} = 0$
- *While* $\max_{f \in \mathcal{F}} \left\langle g - h^{(\ell)}, f \right\rangle_\mu \ge \delta$:
  - $\ell = \ell + 1$
  - *Let* $f_\ell \in \mathcal{F}$ *be such that* $\left\langle g - h^{(\ell-1)}, f_\ell \right\rangle_\mu \ge \delta'$ (Correlation Oracle $\mathcal{O}_{\mu,B}$ Step)
  - *Let* $c_\ell = \delta'$
  - $h^{(\ell)} = \Pi(h^{(\ell-1)} + c_\ell \cdot f_\ell)$
- *Let* $p = \ell$
- *return* $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell$

We will need the following general fact about projections[9] onto a convex body.

Fact 5.4.4 (Implicit in Lemma 3.1 of [Bub15])**.** *Let $\mathcal{Y}$ be a compact convex body in a finite dimensional Hilbert space $\mathcal{V}$ equipped with inner product $\langle \cdot, \cdot \rangle_\nu$ and associated norm $\|\cdot\|_\nu$. Let $\Pi_{\mathcal{Y}}$ be projector onto $\mathcal{Y}$. Then, for $y \in \mathcal{Y}$ and $x \in \mathcal{V}$, we have*

$$\|y - x\|_\nu^2 \;\geq\; \|y - \Pi_{\mathcal{Y}}(x)\|_\nu^2 \;+\; \|\Pi_{\mathcal{Y}}(x) - x\|_\nu^2 .$$

Proof of Lemma 5.4.2. We will show that the norm of $\left\|g - h^{(\ell)}\right\|_\mu$ strictly decreases as the algorithm progresses. Computing we obtain

$$\begin{aligned}
\left\|g - h^{(\ell)}\right\|_\mu^2 &= \left\|g - \Pi(h^{(\ell-1)} + c_\ell \cdot f_\ell)\right\|_\mu^2 \\
&\leq \left\|g - (h^{(\ell-1)} + c_\ell \cdot f_\ell)\right\|_\mu^2 - \left\|(h^{(\ell-1)} + c_\ell \cdot f_\ell) - \Pi(h^{(\ell-1)} + c_\ell \cdot f_\ell)\right\|_\mu^2 \quad \text{(By Fact 5.4.4)} \\
&\leq \left\|g - (h^{(\ell-1)} + c_\ell \cdot f_\ell)\right\|_\mu^2 \\
&= \left\|g - h^{(\ell-1)}\right\|_\mu^2 + c_\ell^2 \cdot \|f_\ell\|_\mu^2 - 2c_\ell \cdot \left\langle g - h^{(\ell-1)}, f_\ell \right\rangle_\mu \\
&\leq \left\|g - h^{(\ell-1)}\right\|_\mu^2 - (\delta')^2
\end{aligned}$$

where the inequality follows from $c_\ell = \delta'$, the bound $\|f_\ell\|_\mu \leq 1$ and

$$\left\langle g - h^{(\ell-1)}, f_\ell \right\rangle_\mu \geq \delta'.$$

Since $\|g\|_\mu^2 \leq B$ and $\left\|g - h^{(\ell)}\right\|_\mu^2$ decreases by at least $(\delta')^2$ in each iteration, we conclude that the algorithm halts in at most $p \leq B/(\delta')^2$ steps.

By construction each c_ℓ is initialized to δ' and can not increase (it can only decrease due to projections). Thus, we obtain $\sum_{\ell=1}^{p} |c_\ell| \leq p \cdot \delta' \leq B/\delta'$. Also by construction at termination $\|h\|_\mu^2 \leq B$. It remains to show that $c_\ell \geq \delta'/(1 + \delta'/\sqrt{B})^p$. Note that

9. See [Bub15, Chapter 3] for a defintion of projector.

the projection $\Pi(h^{(\ell-1)} + c_\ell \cdot f_\ell)$ at each iteration either does nothing to the coefficients c_ℓ's or scales them by a factor of at most $(1 + \delta'/\sqrt{B})$ since $\left\|h^{(\ell-1)}\right\|_\mu + \|c_\ell \cdot f_\ell\|_\mu \leq \sqrt{B}(1 + \delta'/\sqrt{B})$. This readily implies the claimed lower bound on the coefficients c_ℓ's at termination. Moreover, we have $h^{(\ell)} \in \mathcal{H}(B/(\delta')^2, B/\delta', \mathcal{F})$ also by construction.

Running Time: The decomposition algorithm calls the correlation oracle at most $p+1$ times. Since the coefficients c_ℓ always lie in $[\delta'/(1+\delta'/\sqrt{B})^p, \delta'] \subseteq [\delta'/\exp(p\delta'/\sqrt{B}), \delta']$, the bit complexity is $C = O(p\delta'/\sqrt{B})$ and computing the projection (which amounts to computing $h^{(\ell)}/\left\|h^{(\ell)}\right\|_\mu$ if $\left\|h^{(\ell)}\right\|_\mu^2 > B$) takes at most $\widetilde{O}(p^2 \cdot \mathrm{poly}(C) \cdot |\mathrm{Supp}(\mu)|)$. Then the total running time is at most

$$\widetilde{O}(p(\mathcal{T}_{\mathcal{O}_{\mu,B}} + p^2 \cdot \mathrm{poly}(C) \cdot |\mathrm{Supp}(\mu)|)) \;=\; \widetilde{O}\left(\mathrm{poly}(B, 1/\delta') \cdot (\mathcal{T}_{\mathcal{O}_{\mu,B}} + |\mathrm{Supp}(\mu)|)\right),$$

concluding the proof. ∎

Remark 5.4.5. *If we are only interested in an existential version of Lemma 5.4.2, we can always use a trivial existential (δ, δ)-correlation oracle. However, to obtain weak regularity decompositions efficiently in our settings, we will later use efficient (δ, δ')-correlation oracle with $\delta' = \Omega(\delta)$.*

5.4.2 Splittable Mixing Lemma

A splittable collection of tuples gives rise to several expanding split operators (see Definition 5.3.9). This allows us to show that a splittable collection satisfies some higher-order analogues of the well known expander mixing lemmas for graphs (*cf.*,[HLW06][Section 2.4]) as we make precise next.

Lemma 5.4.6 (Splittable Mixing Lemma)**.** *Suppose $W(k) \subseteq [n]^k$ is a τ-splittable collection of tuples. For every $t \in \{0, \ldots, k-2\}$ and every $f, f' \in \mathcal{F}_{t+1}$, we have*

$$\left|\langle f', f\rangle_{\nu_{t+1}} \;-\; \langle f', f\rangle_{\nu_t}\right| \;\leq\; \tau.$$

Proof. Let $f = f_1 \otimes \cdots \otimes f_t \otimes f_{t+1} \otimes f_{t+2}$ and $f' = f'_1 \otimes \cdots \otimes f'_t \otimes f'_{t+1} \otimes f'_{t+2}$. We have

$$\begin{aligned}\left|\langle f', f\rangle_{\nu_{t+1}} - \langle f', f\rangle_{\nu_t}\right| &= \left|\prod_{i=1}^{t} \mathop{\mathbb{E}}_{\mu_1} f_i f'_i\right| \cdot \left|\mathop{\mathbb{E}}_{\mu_1 \otimes \mu_{[t+2,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2} - \mathop{\mathbb{E}}_{\mu_{[t+1,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2}\right| \\ &\leq \left|\mathop{\mathbb{E}}_{\mu_1 \otimes \mu_{[t+2,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2} - \mathop{\mathbb{E}}_{\mu_{[t+1,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2}\right|.\end{aligned}$$

Let $f''_{t+1} = f_{t+1}f'_{t+1}$ and $f''_{t+2} = f_{t+2}f'_{t+2}$. Note that

$$\mathop{\mathbb{E}}_{\mu_1 \otimes \mu_{[t+2,k]}} f''_{t+1} \otimes f''_{t+2} - \mathop{\mathbb{E}}_{\mu_{[t+1,k]}} f''_{t+1} \otimes f''_{t+2} = \left\langle f''_{t+1}, \left(\frac{\mathsf{J}_{\text{rec}}}{|W[t+2,k]|} - \mathsf{S}_{W[t+1],W[t+2,k]}\right) f''_{t+2}\right\rangle_{\mu_1},$$

where J_{rec} is the (rectangular) $|W[t+1]| \times |W[t+2,k]|$ all ones matrix. Using the τ-splittability assumption, we have the following bound on the largest singular value

$$\sigma\left(\frac{\mathsf{J}_{\text{rec}}}{|W[t+2,k]|} - \mathsf{S}_{W[t+1],W[t+2,k]}\right) \leq \sigma_2\left(\mathsf{S}_{W[t+1],W[t+2,k]}\right) \leq \tau.$$

Then

$$\left|\mathop{\mathbb{E}}_{\mu_1 \otimes \mu_{[t+2,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2} - \mathop{\mathbb{E}}_{\mu_{[t+1,k]}} f_{t+1}f'_{t+1} \otimes f_{t+2}f'_{t+2}\right| \leq \tau,$$

concluding the proof. ∎

We can iterate the preceding lemma to obtain the following.

Lemma 5.4.7 (Splittable Mixing Lemma Iterated)**.** *Suppose $W(k) \subseteq [n]^k$ is a τ-splittable collection of tuples. For every $f = f_1 \otimes \cdots \otimes f_k \in \mathcal{F}_{k-1}$, we have*

$$\left|\mathop{\mathbb{E}}_{\nu_0} f - \mathop{\mathbb{E}}_{\nu_{k-1}} f\right| \leq (k-1) \cdot \tau.$$

Proof. Let $1 \in \mathcal{F}_{k-1}$ be the constant 1 function. Note that for any $t \in \{0, \ldots, k-1\}$ the restriction of any $f' \in \mathcal{F}_{k-1}$ to the support of ν_t which we denote by $f'|_t$ belongs to $\mathcal{F}_t$. It

is immediate that $\langle f, 1\rangle_{\nu_t} = \langle f|_t, 1\rangle_{\nu_t}$. Computing we obtain

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{\nu_0} f \;-\; \mathop{\mathbb{E}}_{\nu_{k-1}} f\right| \;&=\; \left|\langle f,1\rangle_{\nu_0} \;-\; \langle f,1\rangle_{\nu_{k-1}}\right| \;\le\; \sum_{i=0}^{k-2}\left|\langle f,1\rangle_{\nu_i} \;-\; \langle f,1\rangle_{\nu_{i+1}}\right| \\ &= \sum_{i=0}^{k-2}\left|\langle f|_t,1|_t\rangle_{\nu_i} \;-\; \langle f|_{t+1},1|_{t+1}\rangle_{\nu_{i+1}}\right| \\ &\le \sum_{i=0}^{k-2}\tau, \qquad \text{(By Lemma 5.4.6)}\end{aligned}$$

finishing the proof. ∎

In Section 5.4.4, we will need two corollaries of the splittable mixing lemma which we prove now.

Claim 5.4.8. *Let $W(k) \subseteq [n]^k$ be a τ-splittable collection of tuples. Let $t \in \{0,\dots,k-2\}$ and $h_{t+1} \in \mathcal{H}(R_0, R_1, \mathcal{F}_{t+1})$. For every $f \in \mathcal{F}_{t+1}$, we have*

$$\left|\langle h_{t+1}, f\rangle_{\nu_{t+1}} \;-\; \langle h_{t+1}, f\rangle_{\nu_t}\right| \;\le\; \tau \cdot R_1.$$

Proof. Since $h_{t+1} \in \mathcal{H}(R_0, R_1, \mathcal{F}_{t+1})$, we can write $h_{t+1} = \sum_\ell c_\ell \cdot f_\ell$, where $f_\ell \in \mathcal{F}_{t+1}$ and $\sum_\ell |c_\ell| \le R_1$. By the splittable mixing lemma, *cf.*, Lemma 5.4.6, we have

$$\left|\langle h_{t+1}, f\rangle_{\nu_{t+1}} \;-\; \langle h_{t+1}, f\rangle_{\nu_t}\right| \;\le\; \sum_\ell |c_\ell| \cdot \left|\langle f_\ell, f\rangle_{\nu_{t+1}} \;-\; \langle f_\ell, f\rangle_{\nu_t}\right| \;\le\; \tau \cdot R_1. \qquad \blacksquare$$

Claim 5.4.9. *Let $W(k) \subseteq [n]^k$ be a τ-splittable collection of tuples. Let $t \in \{0,\dots,k-2\}$ and $h_{t+1} \in \mathcal{H}(R_0, R_1, \mathcal{F}_{t+1})$. Then*

$$\left|\|h_{t+1}\|^2_{\nu_{t+1}} \;-\; \|h_{t+1}\|^2_{\nu_t}\right| \;\le\; \tau \cdot R_1^2.$$

Proof. Since $h_{t+1} \in \mathcal{H}(R_0, R_1, \mathcal{F}_{t+1})$, we can write $h_{t+1} = \sum_\ell c_\ell \cdot f_\ell$, where $f_\ell \in \mathcal{F}_{t+1}$ and

$\sum_\ell |c_\ell| \le R_1$. By the splittable mixing lemma, *cf.*, Lemma 5.4.6, we have

$$\left|\langle h_{t+1}, h_{t+1}\rangle_{\nu_{t+1}} - \langle h_{t+1}, h_{t+1}\rangle_{\nu_t}\right| \le \sum_{\ell,\ell'} |c_\ell| \cdot |c_{\ell'}| \cdot \left|\langle f_\ell, f_{\ell'}\rangle_{\nu_{t+1}} - \langle f_\ell, f_{\ell'}\rangle_{\nu_t}\right| \le \tau \cdot R_1^2. \qquad \blacksquare$$

5.4.3 *Existential Weak Regularity Decomposition*

Using the abstract weak regularity lemma, Lemma 5.4.2, together splittable mixing lemmas of Section 5.4.2, we can obtain (non-constructive) existential weak regularity decompositions for splittable structures.

Lemma 5.4.10 (Existential Weak Regularity for Splittable Tuples)**.** *Let $W(k) \subseteq [n]^k$ be a τ-splittable structure. Let $g \in \mathbb{R}^{W[1]^k}$ be supported on $W(k)$ with $\|g\|_{\mu_k} \le 1$. Let $\mathcal{F} = \mathcal{F}_{k-1}$ (cf., Definition 5.3.19) be arbitrary. For every $\delta > 0$, if $\tau \le O(\delta^2/(k-1))$, then there exists $h \in \mathbb{R}^{W[1]^k}$ supported on $O(1/\delta^2)$ functions in $\mathcal{F}$ such that*

$$\max_{f \in \mathcal{F}} \langle g - h, f\rangle \le \delta \cdot |W(k)|,$$

where the inner product is over the counting measure on $W[1]^k$.

Proof. Apply the weak regularity Lemma 5.4.2, with parameters δ and δ' equal to δ, collection $\mathcal{F}$, input function g, measure $\mu = \mu_k$ (i.e., uniform measure on $W(k)$) and a non-explicit correlation oracle based on the existential guarantee. This yields $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell \in \mathcal{H}(1/\delta^2, 1/\delta, \mathcal{F})$ where

$$\max_{f \in \mathcal{F}} \langle g - h, f\rangle_{\mu_k} \le \delta.$$

Let $f \in \mathcal{F}$. We claim that $h' = h \cdot |W(k)| \,/\, |W[1]|^k$ satisfies the conclusion of the current

lemma. For this, we bound

$$\left||W(k)|\,\langle g-h,f\rangle_{\mu_k} \;-\; \langle g-h',f\rangle\right| \;\le\; \left||W(k)|\,\langle g,f\rangle_{\mu_k} - \langle g,f\rangle\right| \;+\; \sum_{\ell=1}^{p} |c_\ell| \cdot \left||W(k)|\,\langle f_\ell,f\rangle_{\mu_k} \;-\; \frac{|W(k)|}{|W[1]|^k}\,\langle f_\ell,f\rangle\right|.$$

The first term in the RHS above is zero since

$$|W(k)|\,\langle g,f\rangle_{\mu_k} \;=\; \sum_{\mathfrak{s}\in W(k)} g(\mathfrak{s})\cdot f(\mathfrak{s}) \;=\; \langle g,f\rangle\,,$$

where in the second equality we used that g is supported on $W(k)$. Suppose that $f = f_1 \otimes \cdots \otimes f_k$ and $f_\ell = f_{\ell,1} \otimes \cdots \otimes f_{\ell,k}$. Set $f'_\ell = (f_1 \cdot f_{\ell,1}) \otimes \cdots \otimes (f_k \cdot f_{k,1})$ where $(f_j \cdot f_{j,1})$ is the pointwise product of f_j and $f_{j,1}$. Note that

$$\langle f_\ell,f\rangle_{\mu_k} \;=\; \mathop{\mathbb{E}}_{\nu_0}\left[f'_\ell\right] \quad \text{and} \quad \frac{\langle f_\ell,f\rangle}{|W[1]|^k} \;=\; \mathop{\mathbb{E}}_{\nu_{k-1}}\left[f'_\ell\right],$$

where we recall that μ_k is equal to ν_0 and $\mu_1^{\otimes k}$ is equal to ν_{k-1}. Moreover, f'_ℓ is the tensor product of k functions in $\mathbb{R}^{X[1]}$ of ℓ_∞-norm at most 1. By the splittable mixing lemma (*cf.*, Lemma 5.4.7), we have

$$\left|\mathop{\mathbb{E}}_{\nu_0}\left[f'_\ell\right] \;-\; \mathop{\mathbb{E}}_{\nu_{k-1}}\left[f'_\ell\right]\right| \;\le\; (k-1)\cdot\tau.$$

Hence, we obtain

$$\begin{aligned}\left||W(k)|\,\langle g-h,f\rangle_{\mu_k} \;-\; \langle g-h',f\rangle\right| \;&\le\; \sum_{\ell=1}^{p} |c_\ell|\cdot|W(k)|\cdot\left|\mathop{\mathbb{E}}_{\nu_0}\left[f'_\ell\right] \;-\; \mathop{\mathbb{E}}_{\nu_{k-1}}\left[f'_\ell\right]\right| \\ &\le\; \sum_{\ell=1}^{p} |c_\ell|\cdot(k-1)\cdot\tau\cdot|W(k)| \;\le\; \delta\cdot|W(k)|\,,\end{aligned}$$

from which the lemma readily follows. ∎

5.4.4 *Efficient Weak Regularity Decomposition*

The goal of this section is to prove an efficient version of weak regularity that can be computed in near-linear time. We obtain parameters somewhat comparable to those parameters of the existential weak regularity in Lemma 5.4.10 above with a mild polynomial factor loss of $\Theta(1/k^2)$ on the splittability requirement.

Theorem 5.4.11. *[Efficient Weak Regularity] Let $W(k) \subseteq [n]^k$ be a τ-splittable collection of tuples. Let $g \in \mathbb{R}^{W[1]^k}$ be supported on $W(k)$ with $\|g\|_{\mu_k} \leq 1$. Suppose $\mathcal{F}$ is either $\mathrm{CUT}^{\otimes k}$ or $\mathrm{CUT}_{\pm}^{\otimes k}$. For every $\delta > 0$, if $\tau \leq \delta^2/(k^3 \cdot 2^{20})$, then we can find $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell$ with $p = O(k^2/\delta^2)$, $c_1, \ldots, c_p \in \mathbb{R}$ and functions $f_1, \ldots, f_p \in \mathcal{F}$, such that $\|h\|_{\mu_1^{\otimes k}} \leq 2$ and h is a good approximator to g in the following sense*

$$\max_{f \in \mathcal{F}} \quad \left\langle g - \left(\frac{d}{n}\right)^{k-1} h, f \right\rangle \;\leq\; \delta \cdot |W(k)|\,,$$

where the inner product is over the counting measure on $W[1]^k$. Furthermore, h can be found in $\widetilde{O}(2^{2^{\widetilde{O}(k^2/\delta^2)}} \cdot |W(k)|)$ time.

Warm-up: We first sketch a simpler naive algorithmic weak regularity decompoistion for $\mathrm{CUT}^{\otimes k}$ whose parameters are much worse than the existential parameters of Lemma 5.4.10, but it can be computed in near-linear time. The fast accumulation of errors will explain our motivation in designing the efficient algorithm underlying Theorem 5.4.11. The reader only interested in the latter is welcome to skip ahead.

Lemma 5.4.12 (Naive Efficient Weak Regularity)**.** *Let $W' \subseteq W(k)$ where $W(k)$ is τ-splittable. Let $\mathcal{F}$ be either $\mathrm{CUT}^{\otimes k}$ or $\mathrm{CUT}_{\pm}^{\otimes k}$. For every $\delta > 0$, if $\tau \leq (O(\delta))^{2^k}$, then we can find h supported on $(O(1/\delta))^{2^k}$ functions of $\mathcal{F}$ such that*

$$\max_{f \in \mathcal{F}} \quad \langle \mathbf{1}_{W'} - h, f \rangle \;\leq\; (k-1) \cdot \delta \cdot |W(k)|\,,$$

where the inner product is over the counting measure on $W[1]^k$. Furthermore, this can be done in time $\tilde{O}_\delta(|W(k)|)$.

Proof Sketch: In this sketch, our goal is to show the fast accumulation of errors when applying the weak regularity decomposition for matrices. For simplicity, we assume that this can be done in near-linear time on the number of non-zero entries of the matrix. Precise details and much better parameters are given in the proof of Theorem 5.4.11.

Applying the matrix regularity decomposition to $\mathbf{1}_{W'}$, viewed a matrix in $\mathbb{R}^{W[1,k-1]\times W[k]}$ supported on $W[1,k]$, with accuracy parameter $\delta_1 > 0$, we get in $\tilde{O}_{\delta_1}(|W[1,k]|)$ time

$$\left\| \mathbf{1}_{W'} - \frac{d}{n} \sum_{\ell_1=1}^{p_1} c_{\ell_1} \cdot \mathbf{1}_{S_{\ell_1}} \otimes \mathbf{1}_{T_{\ell_1}} \right\|_\square \leq \delta_1 \cdot |W[1,k]|,$$

where $p_1 = O(1/\delta_1^2)$ and $\sum_{\ell_1} \left| c_{\ell_1} \right| \leq O(1/\delta_1)$.

In turn, for each $\mathbf{1}_{S_{\ell_1}}$ viewed a matrix in $\mathbb{R}^{W[1,k-2]\times W[k-1]}$ supported on $W[1,k-1]$, we apply the matrix regularity decomposition with accuracy parameter $\delta_2 > 0$ getting in $\tilde{O}_{\delta_2}(|W[1,k-1]|)$ time

$$\left\| \mathbf{1}_{S_{\ell_1}} - \frac{d}{n} \sum_{\ell_2=1}^{p_2} c_{\ell_2,\ell_1} \cdot \mathbf{1}_{S_{\ell_2,\ell_1}} \otimes \mathbf{1}_{T_{\ell_2,\ell_1}} \right\|_\square \leq \delta_2 \cdot |W[1,k-1]|,$$

where $p_2 = O(1/\delta_2^2)$ and $\sum_{\ell_2} \left| c_{\ell_2,\ell_1} \right| \leq O(1/\delta_2)$. Continuing this process inductively with accuracy parameters $\delta_3, \ldots, \delta_{k-1}$, we obtain

$$h \coloneqq \left(\frac{d}{n}\right)^{k-1} \sum_{\ell_1}^{p_1} \cdots \sum_{\ell_{k-1}=1}^{p_{k-1}} c_{\ell_1} \ldots c_{\ell_1,\ldots,\ell_{k-1}} \cdot \mathbf{1}_{T_{\ell_1,\ldots,\ell_{k-1}}} \otimes \cdots \otimes \mathbf{1}_{T_{\ell_1}},$$

in time $\widetilde{O}_{\delta_1,\ldots,\delta_{k-1}}(|W(k)|)$. We show that h is close in k-tensor cut norm (*cf.*, Defini-

tion 5.3.18) to $\mathbf{1}_{W'}$. Computing we have

$$
\begin{aligned}
&\|\mathbf{1}_{W'} - h\|_{\square^{\otimes k}} \leq \\
&\quad \sum_{j=0}^{k-2} \sum_{\ell_1=1}^{p_1} \cdots \sum_{\ell_j=1}^{p_j} \left| c_{\ell_1} \ldots c_{\ell_1,\ldots,\ell_j} \right| \cdot \\
&\qquad \left\| \mathbf{1}_{S_{\ell_1,\ldots,\ell_j}} - \left(\frac{d}{n}\right)^{k-j-1} \sum_{\ell_{j+1}=1}^{p_{j+1}} c_{\ell_1,\ldots,\ell_{j+1}} \cdot \mathbf{1}_{S_{\ell_1,\ldots,\ell_{j+1}}} \otimes \mathbf{1}_{T_{\ell_1,\ldots,\ell_{j+1}}} \right\|_{\square^{\otimes k-j}} \cdot \\
&\qquad \left(\frac{d}{n}\right)^{j} \cdot \left\| \mathbf{1}_{T_{\ell_1,\ldots,\ell_j}} \otimes \cdots \otimes \mathbf{1}_{T_{\ell_1}} \right\|_{\square^{\otimes j}} \\
\leq \quad & \sum_{j=0}^{k-2} \sum_{\ell_1=1}^{p_1} \cdots \sum_{\ell_j=1}^{p_j} d^j \cdot \left| c_{\ell_1} \ldots c_{\ell_1,\ldots,\ell_j} \right| \cdot \\
&\qquad \left\| \mathbf{1}_{S_{\ell_1,\ldots,\ell_j}} - \left(\frac{d}{n}\right)^{k-j-1} \sum_{\ell_{j+1}=1}^{p} c_{\ell_1,\ldots,\ell_{j+1}} \cdot \mathbf{1}_{S_{\ell_1,\ldots,\ell_{j+1}}} \otimes \mathbf{1}_{T_{\ell_1,\ldots,\ell_{j+1}}} \right\|_{\square} \\
\leq \quad & \sum_{j=0}^{k-2} \sum_{\ell_1=1}^{p_1} \cdots \sum_{\ell_j=1}^{p_j} d^j \cdot \left| c_{\ell_1} \ldots c_{\ell_1,\ldots,\ell_j} \right| \cdot \delta_{j+1} \cdot |W[1, k-j]| \\
\leq \quad & |W(k)| \sum_{j=0}^{k-2} \delta_{j+1} \prod_{\ell=1}^{j} O(1/\delta_\ell).
\end{aligned}
$$

By setting $\delta_j = \Theta(\delta^{2^j})$, the LHS becomes at most $(k-1) \cdot \delta \cdot |W(k)|$. □

We now proceed to prove our main result in this section, namely Theorem 5.4.11. Fist, we establish some extra notation now. Let $W(k)$ be a d-regular collection of tuples. Most of our derivations which are existential hold for a generic $\mathcal{F}_t$ (*cf.*, Definition 5.3.19). However, we only derive near-linear time algorithmic results when $\mathcal{F}_t$ is either the CUT functions

$$
\mathcal{F}_t^{0/1} := \left\{ \pm \mathbf{1}_{S_1} \otimes \cdots \otimes \mathbf{1}_{S_t} \otimes \mathbf{1}_T \mid S_j \subseteq W[1], T \subseteq W[t+1, k] \right\},
$$

or "signed" CUT functions

$$\mathcal{F}_t^{\pm 1} := \left\{ \pm \chi_{S_1} \otimes \cdots \otimes \chi_{S_t} \otimes \chi_T \mid S_j \subseteq W[1], T \subseteq W[t+1,k] \right\},$$

where above we recall that for $S \subseteq [n]$, we have $\chi_S(i) = (-1)^{\mathbf{1}_{i \in S}}$ for $i \in [n]$. Observe that the condition $S_j \subseteq W[1]$ is equivalent to $S_j \subseteq W[i]$ since $W(k)$ is d-regular.

For quick reference, we collect the notation needed in our algorithmic weak regularity decomposition in the following table.

$\mathcal{F}_t := \left\{ \pm f_1 \otimes \cdots \otimes f_t \otimes f_{t+1} \mid f_j \subseteq \mathbb{R}^{W[1]} \text{ for } i \le t, f_{t+1} \subseteq \mathbb{R}^{W[t+1,k]}, \left\| f_j \right\|_\infty \le 1 \right\}$
$\mathcal{F}_t^{0/1} := \left\{ \pm \mathbf{1}_{S_1} \otimes \cdots \otimes \mathbf{1}_{S_t} \otimes \mathbf{1}_T \mid S_j \subseteq W[1], T \subseteq W[t+1,k] \right\} \subseteq \mathcal{F}_t$
$\mathcal{F}_t^{\pm 1} := \left\{ \pm \chi_{S_1} \otimes \cdots \otimes \chi_{S_t} \otimes \chi_T \mid S_j \subseteq W[1], T \subseteq W[t+1,k] \right\} \subseteq \mathcal{F}_t$
$\mathcal{H}(R_0, R_1, \mathcal{F}) := \left\{ \sum_{\ell=1}^{p} c_\ell \cdot f_\ell \mid p \le R_0, \sum \lvert c_\ell \rvert \le R_1, f_\ell \in \mathcal{F} \right\}$
μ_1 is the uniform distribution on $W[1]$ and $\mu_{[t+1,k]}$ is the uniform distribution on $W[t+1,k]$
$\nu_t := (\mu_1)^{\otimes t} \otimes \left(\mu_{[t+1,k]} \right)$

Our main result of this section, namely, the near-linear time weak regularity decomposition Theorem 5.4.11, can be readily deduced from Lemma 5.4.13 below.

Lemma 5.4.13 (Efficient Weak Regularity Induction). *Let $W(k) \subseteq [n]^k$ be a τ-splittable d-regular collection of tuples. Let $g \in \mathcal{F}_0$ and $t \in \{0, \ldots, k-1\}$ with $\|g\|_{\mu_k} \le 1$. For every $\delta > 0$, if $\tau \le \delta^2/(k \cdot 2^{18})$, then there exists $h_t \in \mathcal{H}(O(1/\delta^2), 2^8(1+1/k)^t/\delta, \mathcal{F}_t)$ with $\|h_t\|_{\nu_t}^2 \le (1+1/k)^t$ such that*

$$\max_{f \in \mathcal{F}_t} \left\langle g - \left(\frac{d}{n}\right)^t h_t, f \right\rangle_{\nu_t} \le 2 \cdot \left(\frac{d}{n}\right)^t \cdot t \cdot \delta.$$

Furthermore, the function h_t can be found in $\widetilde{O}((2t)^{2^{O(1/\delta^2)}} \cdot |W(k)|)$ time.

We restate Theorem 5.4.11 below and then prove it assuming Lemma 5.4.13.

Theorem 5.4.11. *[Efficient Weak Regularity] Let $W(k) \subseteq [n]^k$ be a τ-splittable collection of tuples. Let $g \in \mathbb{R}^{W[1]^k}$ be supported on $W(k)$ with $\|g\|_{\mu_k} \leq 1$. Suppose $\mathcal{F}$ is either $\mathrm{CUT}^{\otimes k}$ or $\mathrm{CUT}_{\pm}^{\otimes k}$. For every $\delta > 0$, if $\tau \leq \delta^2/(k^3 \cdot 2^{20})$, then we can find $h = \sum_{\ell=1}^{p} c_\ell \cdot f_\ell$ with $p = O(k^2/\delta^2)$, $c_1, \ldots, c_p \in \mathbb{R}$ and functions $f_1, \ldots, f_p \in \mathcal{F}$, such that $\|h\|_{\mu_1^{\otimes k}} \leq 2$ and h is a good approximator to g in the following sense*

$$\max_{f \in \mathcal{F}} \left\langle g - \left(\frac{d}{n}\right)^{k-1} h, f \right\rangle \leq \delta \cdot |W(k)|,$$

where the inner product is over the counting measure on $W[1]^k$. Furthermore, h can be found in $\widetilde{O}(2^{2^{\widetilde{O}(k^2/\delta^2)}} \cdot |W(k)|)$ time.

Proof. Set $\mathcal{F}_t = \mathcal{F}_t^{0/1}$ if $\mathcal{F} = \mathrm{CUT}^{\otimes k}$ or set $\mathcal{F}_t = \mathcal{F}_t^{\pm 1}$ if $\mathcal{F} = \mathrm{CUT}_{\pm}^{\otimes k}$. We apply Lemma 5.4.13 with $t = k-1$, accuracy δ as $\delta/(2k)$ and input function g. This gives $h_t = \sum_{\ell=1}^{p} c'_\ell \cdot f_\ell \in \mathcal{H}(O(k^2/\delta^2), O(k/\delta), \mathcal{F}_t)$ such that

$$\max_{f \in \mathcal{F}_t} \left\langle g - \left(\frac{d}{n}\right)^{t} h_t, f \right\rangle_{\nu_t} \leq 2 \cdot \left(\frac{d}{n}\right)^{t} \cdot t \cdot \delta. \tag{5.1}$$

Note that $\nu_t = \nu_{k-1} = \mu_1^{\otimes k}$ is the uniform measure on $W[1]^k$. Since $W(k)$ is d-regular, $|W(k)| = |W[1]|^k \cdot (d/n)^{k-1}$. Set $h = \cdot h_t$. Then the guarantee in Eq. (5.1) becomes

$$\max_{f \in \mathcal{F}} \left\langle g - \left(\frac{d}{n}\right)^{k-1} h, f \right\rangle \leq \delta \cdot |W(k)|,$$

where the inner product is under the counting measure. By Lemma 5.4.13, we have $\|h_t\|_{\nu_t}^2 \leq (1 + 1/k)^t \leq e$, so $\|h_t\|_{\nu_t} \leq 2$. Then $\|h\|_{\mu_1^{\otimes k}} \leq 2$. The running time follows from Lemma 5.4.13 completing the proof. ∎

We now prove Lemma 5.4.13 above assuming the following algorithmic result which we prove later.

Lemma 5.4.14. *[Algorithmic Weak Regularity Step] Let $\delta > 0$ and $t \in \{0, \ldots, k-2\}$. Let $h_t \in \mathcal{H}(O(B/\delta^2), O(B/\delta), \mathcal{F}_t)$ with $\|h_t\|_{\nu_t}^2 \leq B$. Then there exists $h_{t+1} \in \mathcal{H}(O(B/\delta^2), 2^8 B/\delta, \mathcal{F}_{t+1})$ with $\|h_{t+1}\|_{\nu_t}^2 \leq B$ such that*

$$\max_{f \in \mathcal{F}_{t+1}} \langle h_t - h_{t+1}, f \rangle_{\nu_t} \leq \delta.$$

Furthermore, each h_{t+1} can be found in time $\widetilde{O}((2t)^{2^{O(1/\delta^2)}} \cdot |W(k)|)$.

Proof of Lemma 5.4.13. We will prove the lemma with the following simple equivalent conclusion

$$\left\langle g - \left(\frac{d}{n}\right)^t h_t, f \right\rangle_{\nu_t} \leq 2 \cdot \left(\frac{d}{n}\right)^t \cdot t \cdot \delta \qquad \Leftrightarrow \qquad \left\langle \left(\frac{n}{d}\right)^t g - h_t, f \right\rangle_{\nu_t} \leq 2 \cdot t \cdot \delta,$$

which we will prove holds for every $f \in \mathcal{F}_t$. The base case $t = 0$ follows immediately by setting $h_0 = g$. Let $t \in \{0, \ldots, k-2\}$. Since $h_t \in \mathcal{H}(O(1/\delta^2), 2^8(1+1/k)^t/\delta, \mathcal{F}_t)$, invoking Lemma 5.4.14 with accuracy parameter δ and input function h_t, we obtain $h_{t+1} \in \mathcal{H}(O(1/\delta^2), 2^8(1+1/k)^{t+1}/\delta, \mathcal{F}_{t+1})$ satisfying

$$\max_{f \in \mathcal{F}_{t+1}} \langle h_t - h_{t+1}, f \rangle_{\nu_t} \leq \delta. \tag{5.2}$$

Let $f \in \mathcal{F}_{t+1}$. We will show that h_{t+1} satisfies the conclusion of the lemma. Expanding

we have

$$\left\langle \left(\frac{n}{d}\right)^{t+1} g - h_{t+1}, f \right\rangle_{\nu_{t+1}} = \underbrace{\left\langle \left(\frac{n}{d}\right)^{t} g - h_t, f \right\rangle_{\nu_t}}_{(i)} + \left(\frac{n}{d}\right)^{t} \cdot \underbrace{\left(\frac{n}{d}\langle g, f\rangle_{\nu_{t+1}} - \langle g, f\rangle_{\nu_t}\right)}_{(ii)}$$
$$+ \underbrace{\langle h_t - h_{t+1}, f\rangle_{\nu_t}}_{(iii)} + \underbrace{\langle h_{t+1}, f\rangle_{\nu_t} - \langle h_{t+1}, f\rangle_{\nu_{t+1}}}_{(iv)}.$$

We will bound each of the terms in RHS above.

Term (i): Suppose $f = f_1 \otimes \cdots \otimes f_{t+1} \otimes f_{t+2} \in \mathcal{F}_{t+1}$. Let $f' = f_1 \otimes \cdots \otimes f_t \otimes f'_{t+1}$, where $f'_{t+1} = (f_{t+1} \otimes f_{t+2})|_{W[t+2,k]}$, so that $f' \in \mathcal{F}_t$. Using the induction hypothesis, we have

$$\left\langle \left(\frac{n}{d}\right)^{t} g - h_t, f \right\rangle_{\nu_t} = \left\langle \left(\frac{n}{d}\right)^{t} g - h_t, f' \right\rangle_{\nu_t} \leq 2 \cdot t \cdot \delta.$$

Term (ii): Since $g \in \mathcal{F}_0$, it is supported on $W(k)$ and so we have

$$\begin{aligned}\langle g, f\rangle_{\nu_t} &= \frac{1}{|W[1]|^t \, |W[t+1,k]|} \sum_{\mathfrak{s} \in W(k)} g(\mathfrak{s}) \cdot f(\mathfrak{s}) \\ &= \frac{n}{d} \cdot \frac{1}{|W[1]|^{t+1} \, |W[t+2,k]|} \sum_{\mathfrak{s} \in W(k)} g(\mathfrak{s}) \cdot f(\mathfrak{s}) = \frac{n}{d} \cdot \langle g, f\rangle_{\nu_{t+1}}.\end{aligned}$$

where the second equality follows from $|W[t+1,k]| = d \cdot |W[t+2,k]|$ by the d-regular assumption.

Term (iii): By Eq. (5.2), we have $\langle h_t - h_{t+1}, f\rangle_{\nu_t} \leq \delta$.

Term (iv): For notional convenience, set $R_1 = 2^8(1+1/k)^{t+1}/\delta$. Since $h_{t+1} \in \mathcal{H}(\infty, R_1, \mathcal{F}_{t+1})$ and the splittability parameter τ satisfies $\tau \leq \delta^2/(k \cdot 2^{18})$, from Claim 5.4.8 we obtain

$$\langle h_{t+1}, f\rangle_{\nu_t} - \langle h_{t+1}, f\rangle_{\nu_{t+1}} \leq \tau \cdot R_1 \leq \delta.$$

Putting everything together yields

$$\left\langle \left(\frac{n}{d}\right)^{t+1} g - h_t, f \right\rangle_{\nu_{t+1}} \;\leq\; \underbrace{2 \cdot t \cdot \delta}_{(i)} + \left(\frac{n}{d}\right)^{t} \cdot \underbrace{0}_{(ii)} + \underbrace{\delta}_{(iii)} + \underbrace{\delta}_{(iv)} \;\leq\; 2 \cdot (t+1) \cdot \delta,$$

concluding the claimed inequality.

Now we use the bound $\|h_{t+1}\|_{\nu_t}^2 \leq \|h_t\|_{\nu_t}^2$ from Lemma 5.4.14 together with the splittability assumption $\tau \leq \delta^2/(k \cdot 2^{18})$ to bound the norm $\|h_{t+1}\|_{\nu_{t+1}}^2$ under the new measure ν_{t+1}. Under these assumptions and using Claim 5.4.9 we get

$$\begin{aligned} \left| \|h_{t+1}\|_{\nu_{t+1}}^2 - \|h_{t+1}\|_{\nu_t}^2 \right| \;\leq\; \tau \cdot R_1^2 \;&\leq\; \frac{\delta^2}{k \cdot 2^{18}} \cdot \frac{2^{16}(1+1/k)^{2(t+1)}}{\delta^2} \\ &\leq\; \frac{(1+1/k)^t}{k}. \end{aligned}$$

where we used the bounds on τ, R_1 and $(1+1/k)^{(t+2)} \leq 4$ for $0 \leq t \leq k-2$. From the previous inequality and the induction hypothesis $\|h_t\|_{\nu_t}^2 \leq (1+1/k)^t$, we finally get $\|h_{t+1}\|_{\nu_{t+1}}^2 \leq (1+1/k)^{t+1}$ as desired. ∎

We now show a near-linear time weak regularity decomposition for special functions of the form $h_t \in \mathcal{H}(O(1/\delta^2), O(1/\delta), \mathcal{F}_t)$ that admit a tensor product structure. The goal is to design a correlation oracle that exploits the special tensor product structure of the function $(h_t - h_{t+1}^{(\ell)})$, where $h_{t+1}^{(\ell)}$ is the ℓth approximator of h_t in the abstract weak regularity algorithm (*cf.*, Algorithm 5.4.3).

Lemma 5.4.14. *[Algorithmic Weak Regularity Step] Let $\delta > 0$ and $t \in \{0, \ldots, k-2\}$. Let $h_t \in \mathcal{H}(O(B/\delta^2), O(B/\delta), \mathcal{F}_t)$ with $\|h_t\|_{\nu_t}^2 \leq B$. Then there exists $h_{t+1} \in \mathcal{H}(O(B/\delta^2), 2^8 B/\delta, \mathcal{F}_{t+1})$ with $\|h_{t+1}\|_{\nu_t}^2 \leq B$ such that*

$$\max_{f \in \mathcal{F}_{t+1}} \langle h_t - h_{t+1}, f \rangle_{\nu_t} \;\leq\; \delta.$$

Furthermore, each h_{t+1} can be found in time $\widetilde{O}((2t)^{2^{O(1/\delta^2)}} \cdot |W(k)|)$.

Our correlation oracle for higher-order tensors will make calls to a correlation oracle for matrices Theorem 5.4.15 (i.e., 2-tensors) stated below. This matrix oracle is presented in Section 5.4.5 and it follows from a simple combination of a matrix cut norm approximation algorithm by Alon and Naor [AN04] with known fast SDP solvers for sparse matrices such as those by Lee and Padmanabhan [LP20] and Arora and Kale [AK07].

Theorem 5.4.15. *[Alon–Naor Correlation Oracle] Let $\mathcal{F}$ be either $\mathrm{CUT}^{\otimes 2}$ or $\mathrm{CUT}_{\pm}^{\otimes 2}$ and μ be the uniform measure supported on at most m elements of $[n'] \times [n']$. There exists an algorithmic $(\delta, \alpha_{\mathbf{AN}} \cdot \delta)$-correlation oracle $\mathcal{O}_{\mu,B}$ running in time $\mathcal{T}_{\mathcal{O}_{\mu,B}} = \tilde{O}\left(\mathrm{poly}(B/\delta) \cdot (m + n')\right)$, where $\alpha_{\mathbf{AN}} \geq 1/2^4$ is an approximation ratio constant.*

Proof. We will apply the abstract weak regularity lemma, *cf.*,Lemma 5.4.2, with $\mathcal{F} = \mathcal{F}_{t+1}$, $\delta, \delta' = \delta/2^8$ and $\mu = \nu_t$. This will result in a function from $\mathcal{H}(O(B/\delta^2), 2^8 B/\delta, \mathcal{F}_{t+1})$.

Correlation oracle task: To make this application take near-linear time, we need to specify a correlation oracle $\mathcal{O}_{\nu_t} = \mathcal{O}_{\nu_t, O(1)}$ and now we take advantage of the special tensor structure in our setting. We want an oracle that given

$$h_t = \sum_{\ell=1}^{p} c_\ell \cdot g_\ell, \quad g_\ell \in \mathcal{F}_t, \qquad g_\ell = g_{\ell,1} \otimes \cdots \otimes g_{\ell,t} \otimes \underbrace{g_{\ell,t+1}}_{\in \mathbb{R}^{W[t+1,k]}} \quad \text{and}$$

$$h_{t+1} = \sum_{\ell=1}^{p} c'_\ell \cdot g'_\ell, \quad g'_\ell \in \mathcal{F}_{t+1}, \quad g'_\ell = g'_{\ell,1} \otimes \cdots \otimes g'_{\ell,t} \otimes \underbrace{g'_{\ell,t+1}}_{\in \mathbb{R}^{W[1]}} \otimes \underbrace{g'_{\ell,t+2}}_{\in \mathbb{R}^{W[t+2,k]}},$$

if there exists

$$f = f_1 \otimes \cdots \otimes f_t \otimes \underbrace{f_{t+1}}_{\in \mathbb{R}^{W[1]}} \otimes \underbrace{f_{t+2}}_{\in \mathbb{R}^{W[t+2,k]}} \in \mathcal{F}_{t+1}$$

satisfying

$$\langle h_t - h_{t+1}, f \rangle_{\nu_t} \geq \delta,$$

for some $f \in \mathcal{F}_{t+1}$, finds $f' \in \mathcal{F}_{t+1}$ in near-linear time such that

$$\langle h_t - h_{t+1}, f' \rangle_{\nu_t} \;\geq\; \delta' \;=\; \frac{\delta}{2^8}.$$

Here, h_{t+1} is the current approximator of h_t in the abstract weak regularity algorithm and, by Lemma 5.4.2, $h_{t+1} \in \mathcal{H}(O(1/\delta^2), 2^8(1+1/k)^{t+1}/\delta, \mathcal{F}_{t+1})$. Expanding $\langle h_t - h_{t+1}, f \rangle_{\nu_t}$ we get

$$\begin{aligned}\langle h_t - h_{t+1}, f \rangle_{\nu_t} \;=\; & \sum_{\ell=1}^{p} c_\ell \underbrace{\prod_{j=1}^{t} \left\langle g_{\ell,j}, f_j \right\rangle_{\mu_1}}_{\gamma_\ell} \cdot \langle g_{\ell,t+1}, f_{t+1} \otimes f_{t+2} \rangle_{\mu_{[t+1,k]}} \;- \\ & \sum_{\ell=1}^{p} c'_\ell \underbrace{\prod_{j=1}^{t} \left\langle g'_{\ell,j}, f_j \right\rangle_{\mu_1}}_{\gamma'_\ell} \cdot \left\langle g'_{\ell,t+1} \otimes g'_{\ell,t+2}, f_{t+1} \otimes f_{t+2} \right\rangle_{\mu_{[t+1,k]}},\end{aligned}$$

where we define $\gamma_\ell \coloneqq \prod_{j=1}^{t} \left\langle g_{\ell,j}, f_j \right\rangle_{\mu_1}$ and $\gamma'_\ell \coloneqq \prod_{j=1}^{t} \left\langle g'_{\ell,j}, f_j \right\rangle_{\mu_1}$ for $\ell \in [p]$, $j \in [t]$. Suppose $g_{\ell,j} = f_{S_{\ell,j}}$ and $g'_{\ell,j} = f_{S'_{\ell,j}}$ for $\ell \in [p]$, $j \in [t]$, where $f_{S_{\ell,j}}, f_{S'_{\ell,j}}$ are either $\mathbf{1}_{S_{\ell,j}}, \mathbf{1}_{S'_{\ell,j}}$ or $\chi_{S_{\ell,j}}, \chi_{S'_{\ell,j}}$ depending on $\mathcal{F}_t$ being $\mathcal{F}_t^{0/1}$ or $\mathcal{F}_t^{\pm 1}$, respectively.

Sigma-algebra brute force: Now for each $j \in [t]$, we form the σ-algebra Σ_j generated by $\{S_{\ell,j}, S'_{\ell,j}\}_{\ell \in [p]}$ which can be done in $2^p \cdot \widetilde{O}(|W[1]|)$ time by Remark 5.3.16 and yields at most 2^p atoms. Hence, the generation of all these σ-algebras takes at most $t \cdot 2^p \cdot \widetilde{O}(|W[1]|)$ time. Suppose $f_j = f_{S_j}$ for some $S_j \subseteq W[1]$. Let $\eta > 0$ be an approximation parameter to be specified shortly. For each atom $\sigma_{j'} \in \Sigma_j$, we enumerate over all possible values for the ratio $\left|\sigma_{j'} \cap S_j\right| / \left|\sigma_{j'}\right|$ up to accuracy η. More precisely, if $\left|\sigma_{j'}\right| \geq 1/\eta$, we consider the values

$$0, 1 \cdot \eta, 2 \cdot \eta, \ldots, \lfloor 1/\eta \rfloor \cdot \eta,$$

and we consider $0, 1/\left|\sigma_{j'}\right|, 2/\left|\sigma_{j'}\right|, \ldots, \left|\sigma_{j'}\right| / \left|\sigma_{j'}\right|$ otherwise. Let $\left|\Sigma_j\right|$ denote the number

of atoms in Σ_j. This enumeration results in $\prod_{j=1}^{t}(1/\eta)^{\left|\Sigma_j\right|}$ configurations which allows us to approximate any realizable values for $\left\langle g_{\ell,j}, f_j \right\rangle_{\mu_1}$ within additive error at most $4 \cdot \eta$ since either

$$\left\langle g_{\ell,j}, f_j \right\rangle_{\mu_1} = \mathbb{E}_{\mu_1}\left[\mathbf{1}_{S_{\ell,j}} \cdot \mathbf{1}_{S_j}\right] = \frac{\left|S_{\ell,j} \cap S_j\right|}{|W[1]|} = \frac{1}{|W[1]|} \sum_{\sigma_{j'} \subseteq S_{\ell,j}} \left|\sigma_{j'} \cap S_j\right| \quad \text{or}$$

$$\begin{aligned} \left\langle g_{\ell,j}, f_j \right\rangle_{\mu_1} = \mathbb{E}_{\mu_1}\left[\chi_{S_{\ell,j}} \cdot \chi_{S_j}\right] &= \frac{|W[1]| - 2(\left|S_{\ell,j}\right| + \left|S_j\right| - 2\left|S_{\ell,j} \cap S_j\right|)}{|W[1]|} \\ &= \frac{|W[1]| - 2(\left|S_{\ell,j}\right| + \sum_{\sigma_{j'}} \left|\sigma_{j'} \cap S_j\right| - 2\sum_{\sigma_{j'} \subseteq S_{\ell,j}} \left|\sigma_{j'} \cap S_j\right|)}{|W[1]|}, \end{aligned}$$

according to $\mathcal{F}_{t+1}$. We can approximate $\left\langle g'_{\ell,j}, f_j \right\rangle_{\mu_1}$ similarly. In turn, we can approximate each of the realizable values in $\{\gamma_\ell, \gamma'_\ell\}_{\ell \in [p]}$ within additive error $4 \cdot t \cdot \eta$ by some configuration of fractional value assignment to the atoms of each σ-algebra.

Invoking the matrix correlation oracle: Let $\mathsf{A} \coloneqq \sum_\ell \left(c_\ell \cdot \gamma_\ell \cdot g_{\ell,t+1} + c'_\ell \cdot \gamma'_\ell \cdot g'_{\ell,t+1} \otimes g'_{\ell,t+2}\right)$. We conveniently view A as a *sparse* matrix of dimension $|W[t+1]| \times |W[t+2,k]|$ with at most $|W[t+1,k]|$ non-zeros entries. Define $\varphi_{\mathsf{A}}(f_{t+1}, f_{t+2}) \coloneqq \langle \mathsf{A}, f_{t+1} \otimes f_{t+2} \rangle_{\mu_{[t+1,k]}}$. Define

$$\mathsf{OPT}(\mathsf{A}) \coloneqq \max_{f_{t+1}, f_{t+2}} \varphi_{\mathsf{A}}(f_{t+1}, f_{t+2}), \tag{5.3}$$

where f_{t+1}, f_{t+2} range over valid $f_{S_{t+1}}, f_{S_{t+2}}$ (again according to kind of $\mathcal{F}_{t+1}$ we have). In the computation of $\mathsf{OPT}(\mathsf{A})$, we have incurred so far an additive error of at most

$$4 \cdot t \cdot \eta \cdot \sum_\ell (|c_\ell| + |c'_\ell|).$$

Let $\widetilde{\mathsf{A}}$ be obtained from A by zeroing out all entries of absolute value smaller than $\delta/8$. Note that $\mathsf{OPT}(\widetilde{\mathsf{A}}) \geq \mathsf{OPT}(\mathsf{A}) - \delta/8$ and the absolute value of the entries of $\widetilde{\mathsf{A}}$ lie $[\delta/8, O(1/\delta)]$.

For each entry of A, we compute a rational approximation $\pm P/Q$ where $Q = \Theta(1/\delta)$ and $P \in [1, O(1/\delta)]$ obtaining $\widetilde{\mathsf{A}}'$ such that

$$\mathsf{OPT}(\widetilde{\mathsf{A}}') \;\geq\; \mathsf{OPT}(\widetilde{\mathsf{A}}) - \delta/8 \;\geq\; \mathsf{OPT}(\widetilde{\mathsf{A}}) \;\geq\; \mathsf{OPT}(\mathsf{A}) - \delta/4.$$

Using Theorem 5.4.15 with accuracy parameter $\delta/4$ and input matrix $\widetilde{\mathsf{A}}'$, we obtain in $\mathcal{T}_{\mathsf{A}} := \widetilde{O}(\mathrm{poly}(1/\delta) \cdot |W[t+1,k]|)$ time, with an extra additive error of $\delta/4$ and a multiplicative guarantee of $\alpha_{\mathbf{AN}}$, a 2-tensor $\tilde{f}_{t+1} \otimes \tilde{f}_{t+2}$ satisfying

$$\varphi_{\widetilde{\mathsf{A}}}(\tilde{f}_{t+1}, \tilde{f}_{t+2}) \;\geq\; \alpha_{\mathbf{AN}} \cdot \left(\mathsf{OPT}(\mathsf{A}) \;-\; 2 \cdot \frac{\delta}{4} \;-\; 4 \cdot t \cdot \eta \cdot \sum_{\ell} (|c_\ell| + |c'_\ell|) \right).$$

Since $h_t \in \mathcal{H}(O(1/\delta^2), 2^8 \cdot (1+1/k)^t/\delta, \mathcal{F}_t)$ and $h_{t+1} \in \mathcal{H}(O(1/\delta^2), 2^8 \cdot (1+1/k)^{t+1}/\delta, \mathcal{F}_{t+1})$, we have $\sum_\ell (|c_\ell| + |c'_\ell|) \leq 2^{10}/\delta$ and $p = O(1/\delta^2)$. By choosing $\eta \leq O(\delta^2/t)$ appropriately, we can bound

$$4 \cdot t \cdot \eta \cdot \sum_{\ell} (|c_\ell| + |c'_\ell|) \;\leq\; 4 \cdot t \cdot \frac{2^{10}}{\delta} \cdot \eta \;\leq\; \frac{\delta}{4}.$$

Hence, $\varphi_{\widetilde{\mathsf{A}}}(\tilde{f}_{t+1}, \tilde{f}_{t+2}) \geq \alpha_{\mathbf{AN}} \cdot \delta/4$ since we are under the assumption that $\mathsf{OPT}(\mathsf{A}) \geq \delta$.

Running Time: First, observe that with our choices of parameters the total number of configurations m_{config} is at most

$$m_{\text{config}} \;\leq\; \prod_{j=1}^{t} (1/\eta)^{\left|\Sigma_j\right|} \;\leq\; \left(\frac{t}{\delta^2}\right)^{2^p} \;\leq\; (2t)^{2^{O(1/\delta^2)}},$$

so that the correlation oracle $\mathcal{O}_{\nu_t}$ takes time at most

$$m_{\text{config}} \cdot \mathcal{T}_{\mathsf{A}} \;\leq\; (2t)^{2^{O(1/\delta^2)}} \cdot \widetilde{O}(\mathrm{poly}(1/\delta) \cdot |W[t+1,k]|) \;=\; \widetilde{O}((2t)^{2^{O(1/\delta^2)}} \cdot |W[t+1,k]|).$$

Using the running time of the oracle $\mathcal{O}_{\nu_t}$, the total running time of the weak regularity decomposition follows from Lemma 5.4.2 which concludes the proof. ∎

5.4.5 *Near-linear Time Matrix Correlation Oracles*

The main result of this section, Theorem 5.4.15 below, is a near-linear time correlation oracle for $\mathrm{CUT}^{\otimes 2}$ and $\mathrm{CUT}_{\pm}^{\otimes 2}$. We combine the constant factor approximation algorithms of Alon–Naor [AN04] for $\|\mathsf{A}\|_{\infty\to 1}$ and $\|\mathsf{A}\|_{\square}$ based on semi-definite programming (SDP) with the faster SDP solvers for sparse matrices such as those by Lee and Padmanabhan [LP20] and by Arora and Kale [AK07]. We point out that these SDP solvers provide additive approximation guarantees which are sufficient for approximating several CSPs, e.g., MaxCut, but they do not seem to provide non-trivial multiplicative approximation guarantees for $\|\mathsf{A}\|_{\infty\to 1}$ or $\|\mathsf{A}\|_{\square}$ in general. Since in our applications of computing regularity decomposition we are only interested in additive approximations, those solvers provide non-trivial sufficient approximation guarantees for $\|\mathsf{A}\|_{\infty\to 1}$ or $\|\mathsf{A}\|_{\square}$ in our settings.

Theorem 5.4.15. *[Alon–Naor Correlation Oracle] Let $\mathcal{F}$ be either $\mathrm{CUT}^{\otimes 2}$ or $\mathrm{CUT}_{\pm}^{\otimes 2}$ and μ be the uniform measure supported on at most m elements of $[n'] \times [n']$. There exists an algorithmic $(\delta, \alpha_{\mathbf{AN}} \cdot \delta)$-correlation oracle $\mathcal{O}_{\mu,B}$ running in time $\mathcal{T}_{\mathcal{O}_{\mu,B}} = \tilde{O}\left(\mathrm{poly}(B/\delta) \cdot (m + n')\right)$, where $\alpha_{\mathbf{AN}} \geq 1/2^4$ is an approximation ratio constant.*

Theorem 5.4.15 is a simple consequence of the following theorem.

Theorem 5.4.16. *Let $\mathsf{A} \in \mathbb{R}^{n\times n}$ be a matrix of integers with at most m non-zero entries. Let $\delta \in (0, 2^{-5}]$ be an accuracy parameter. Suppose that*

$$\mathsf{OPT} := \max_{x_i, y_i \in \{\pm 1\}} \sum_{i,j=1}^{n} \mathsf{A}_{i,j} x_i y_j \;\geq\; \delta \cdot m.$$

Then, with high probability,i.e., $o_n(1)$, we we can find in $\tilde{O}\left(\mathrm{poly}(\|\mathsf{A}\|_{\infty}/\delta) \cdot (m+n)\right)$ time

vectors $\tilde{x}, \tilde{y} \in \{\pm 1\}^n$ *such that*

$$\sum_{i,j=1}^{n} \mathsf{A}_{i,j} \tilde{x}_i \tilde{y}_j \;\geq\; \frac{1}{4} \cdot \mathsf{OPT},$$

and find sets $\tilde{S}, \tilde{T} \subseteq [n]$ *such that*

$$\left| \sum_{i \in \tilde{S}, j \in \tilde{T}} \mathsf{A}_{i,j} \right| \;\geq\; \frac{1}{2^4} \cdot \|\mathsf{A}\|_{\square},$$

where $\|\mathsf{A}\|_{\square}$ *is the cut norm of* A.

The proof of the preceding theorem will rely on the following result which encapsulates the known sparse SDP solvers [AK07, LP20]. For concreteness, we will rely on [LP20] although the guarantee from [AK07] also suffice for us.

Lemma 5.4.17. *[Sparse SDP Solver Wrapper based on [LP20] and partially on [AK07]] Let* $\mathsf{C} \in \mathbb{R}^{n \times n}$ *be a matrix with at most* m *non-zero entries. For every accuracy* $\gamma > 0$*, with high probability we can find in time* $\widetilde{O}((m+n)/\mathrm{poly}(\gamma))$ *vectors* $u_1, \ldots, u_n \in \mathbb{R}^n$ *in the unit ball (i.e.,* $\|u_i\| \leq 1$*) such that that the matrix* $\widetilde{\mathsf{X}}_{i,j} := \left\langle u_i, u_j \right\rangle$ *satisfies*

$$\mathrm{Tr}\left(\mathsf{C} \cdot \widetilde{\mathsf{X}}\right) \;\geq\; \max_{\mathsf{X} \succeq 0, X_{i,i} \leq 1} \mathrm{Tr}\left(\mathsf{C} \cdot \mathsf{X}\right) - \gamma \sum_{i,j} \left|\mathsf{C}_{i,j}\right|.$$

Proof of Theorem 5.4.16. We now implement the strategy mentioned above of combing the approximation algorithms of Alon–Naor [AN04] with the near-linear time sparse SDP solvers. We still need to argue that this indeed leads to the claimed approximation guarantees while being computable in near-linear time overall. We point out that Alon–Naor actually give a constant factor SDP based approximation algorithm for $\|\mathsf{A}\|_{\infty \to 1}$ from which a constant factor approximation algorithm for $\|\mathsf{A}\|_{\square}$ can be readily deduced from

in near-linear time incurring an extra $1/4$ factor approximation loss[10]. Using the matrix A, we set

$$\mathsf{C} := \frac{1}{2}\begin{pmatrix} 0 & \mathsf{A} \\ \mathsf{A}^\dagger & 0 \end{pmatrix}.$$

The SDP relaxation of Alon–Naor for $\|\mathsf{A}\|_{\infty\to 1}$ becomes

$$\begin{aligned} \max \quad & \mathrm{Tr}(\mathsf{C}\cdot\mathsf{X}) && =: \mathsf{SDP}^* \\ \text{s.t.} \quad & \mathsf{X}_{i,i} \leq 1 && \forall i \in [2n] \\ & \mathsf{X} \succeq 0, \end{aligned}$$

except for the constraints $X_{i,i} \leq 1$ which they instead take to be $X_{i,i} = 1$. This technical difference will play a (small) role in the rounding of this SDP since Alon–Naor analysis relies on Gram vectors of X being on the unit sphere. Moreover, we will be solving this SDP within only a weak additive approximation guarantee[11]. Although these technical differences need to be handled, this will be simple to do.

Applying the solver of Lemma 5.4.17 with accuracy parameter $\gamma = \delta^2/\|\mathsf{A}\|_\infty$ to the above SDP, we obtain in $\widetilde{O}(\mathrm{poly}(\|\mathsf{A}\|_\infty/\delta)\cdot(m+n))$ time vectors $u_1,\ldots,u_{2n} \in \mathbb{R}^{2n}$ in the unit ball so that the matrix $\widetilde{\mathsf{X}}_{i,j} := \left\langle u_i, u_j \right\rangle$ satisfy

$$\mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}\right) \geq \max_{\mathsf{X}\succeq 0, X_{i,i}\leq 1} \mathrm{Tr}\left(\mathsf{C}\cdot\mathsf{X}\right) - \delta^2\cdot m.$$

By assumption, we have $\mathsf{SDP}^* := \max_{\mathsf{X}\succeq 0, X_{i,i}\leq 1} \mathrm{Tr}\left(\mathsf{C}\cdot\mathsf{X}\right) \geq \mathsf{OPT} \geq \delta\cdot m$, in which case

10. In Section 5.4 of Alon–Naor [AN04], there is a transformation avoiding any loss in the approximation ratio. Since constant factors are not asymptotically important for us, we rely on the simpler transformation which loses a factor of $1/4$. It simply consists in choosing $\widetilde{S} \in \{\{i \mid \widetilde{x}_i = 1\}, \{i \mid \widetilde{x}_i = -1\}\}$ and $\widetilde{T} \in \{\{j \mid \widetilde{y}_j = 1\}, \{j \mid \widetilde{y}_j = -1\}\}$ maximizing $\mathbf{1}_{\widetilde{S}}^t \mathsf{A} \mathbf{1}_{\widetilde{T}}$, which can be done in near-linear time given as input $\widetilde{x}, \widetilde{y}$.

11. This may not be sufficient to obtain $X_{i,i} \approx 1$ by an extremality argument

the above guarantee becomes

$$\mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}\right) \;\geq\; (1-\delta)\cdot\mathsf{SDP}^*.$$

To obtain diagonal entries equal to 1 in our SDP solution we simply consider the new SDP solution $\widetilde{\mathsf{X}}' = \widetilde{\mathsf{X}} + \Lambda$, where Λ is the diagonal matrix defined as $\Lambda_{i,i} \coloneqq 1 - \widetilde{\mathsf{X}}_{i,i}$. Gram vectors $u'_1, \ldots, u'_{2n}$ of $\widetilde{\mathsf{X}}'$ can be obtained in near-linear time from $u_1, \ldots, u_{2n}$ and Λ by setting

$$u'_i \coloneqq u_i \oplus \sqrt{\Lambda_{i,i}}\cdot e_i \;\in\; \mathbb{R}^{2m}\oplus\mathbb{R}^{2m},$$

where $e_i \in \mathbb{R}^{2m}$ has a one at the ith position and zero everywhere else. Observe that for our particular C, we have

$$\mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}'\right) \;=\; \mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}\right).$$

We now proceed to round $\widetilde{\mathsf{X}}'$ according to the rounding scheme of Alon–Naor [AN04] (*cf.*,Section 5.1) which was chosen because it is simple enough to easily afford a near-linear time computation while providing a $\approx 0.27 \geq 1/4$ approximation guarantee [12] This rounding consists in sampling a Gaussian vector $g \sim N(0, \mathsf{I}_d)$ and setting $\widetilde{x}_i \coloneqq \operatorname{sgn}\langle u'_i, g\rangle$ and $\widetilde{y}_{i+n} \coloneqq \operatorname{sgn}\left\langle u'_{i+n}, g\right\rangle$ for $i \in [n]$. To analyze the approximation guarantee, the following identity is used.

Fact 5.4.18 (Alon–Naor [AN04], *cf.*,Eq. 5)**.** *Let $u, w \in \mathbb{R}^d$ be unit vectors in ℓ_2-norm. Then*

$$\frac{\pi}{2}\cdot\mathbb{E}\left[\operatorname{sgn}\langle u,g\rangle\operatorname{sgn}\langle w,g\rangle\right] = \langle u,w\rangle + \mathbb{E}\left[\left(\langle u,g\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\langle u,g\rangle\right)\left(\langle w,g\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\langle w,g\rangle\right)\right],$$

where the expectations are taken with respect to a random Gaussian vector $g \sim N(0, \mathsf{I}_d)$.

12. Alon–Naor [AN04] have a more sophisticated rounding scheme that achieves $0.56 \geq 1/2$ approximation. In our applications, it is important to have a constant factor approximation, but the distinction between $1/2$ and the weaker $1/4$ factor approximation guarantee is not asymptotically relevant.

Using Fact 5.4.18, the expected value of the rounding, i.e.,

$$\mathbb{E}\left[\sum_{i,j} \mathsf{A}_{i,j} \operatorname{sgn}\left\langle u'_i, g\right\rangle \operatorname{sgn}\left\langle u'_{j+n}, g\right\rangle\right],$$

becomes

$$\frac{2}{\pi}\cdot\sum_{i,j}\mathsf{A}_{i,j}\left\langle u'_i, u'_{j+n}\right\rangle + \frac{2}{\pi}\cdot\sum_{i,j}\mathsf{A}_{i,j}\mathbb{E}\left[\left(\left\langle u'_i, g\right\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\left\langle u'_i, g\right\rangle\right)\left(\left\langle u'_{j+n}, g\right\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\left\langle u'_{j+n}, g\right\rangle\right)\right],$$

As in Alon–Naor [AN04], we will use the fact that $\left\langle u'_i, g\right\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\left\langle u'_i, g\right\rangle$ and $\left\langle u'_{j+n}, g\right\rangle - \sqrt{\frac{\pi}{2}}\operatorname{sgn}\left\langle u'_{j+n}, g\right\rangle$ are themselves vectors on a Hilbert space with norm squared $\pi/2 - 1$. Then, in our setting we obtain

$$\begin{aligned}
\mathbb{E}\left[\sum_{i,j}\mathsf{A}_{i,j}\operatorname{sgn}\left\langle u'_i, g\right\rangle\operatorname{sgn}\left\langle u'_{j+n}, g\right\rangle\right] &\geq \frac{2}{\pi}(1-\delta)\cdot\mathsf{SDP}^* - \left(1-\frac{2}{\pi}\right)\cdot\mathsf{SDP}^* \\
&\geq \frac{2}{\pi}\left(2-\frac{\pi}{2}-\delta\right)\cdot\mathsf{SDP}^* \\
&\geq \left(\frac{1}{4}+\Omega(1)\right)\cdot\mathsf{SDP}^* && \text{(Since } \delta \leq 2^{-5}\text{)} \\
&\geq \left(\frac{1}{4}+\Omega(1)\right)\cdot\mathsf{OPT},
\end{aligned}$$

as claimed. By standard techniques, this guarantee on the expected value of the rounded solution can be used to give with high probability a guarantee of $1/4\cdot\mathsf{OPT}$ (namely, by repeating this rounding scheme $O(\mathrm{poly}(1/\gamma)\cdot\log(n))$ times). ∎

We now proceed to establish the sparse SDP solver wrapper claimed in Lemma 5.4.17. For concreteness, we will use the following sparse SDP solver result of Lee–Padmanabhan [LP20]. The analogous result of Arora–Kale [AK07] with slightly worse parameters also suffices for our purposes, but the main result of [LP20] is stated in more convenient form.

Theorem 5.4.19 (Adapted from Theorem 1.1 of [LP20])**.** *Given a matrix $\mathsf{C} \in \mathbb{R}^{n\times n}$ with m non-zero entries, parameter $\gamma \in (0, 1/2]$, with high probability, in time $\widetilde{O}((m+n)/\gamma^{3.5})$, it is*

possible to find a symmetric matrix $\mathsf{Y} \in \mathbb{R}^{n\times n}$ *with* $O(m)$ *non-zero entries and diagonal matrix* $\mathsf{S} \in \mathbb{R}^{n\times n}$ *so that* $\widetilde{X} = \mathsf{S} \cdot \exp \mathsf{Y} \cdot \mathsf{S}$ *satisfies*

- $\widetilde{X} \succeq 0$,
- $\widetilde{X}_{i,i} \leq 1$ *for every* $1 \leq i \leq n$, *and*
- $\mathrm{Tr}(\mathsf{C} \cdot \widetilde{X}) \geq \max_{\mathsf{X}\succeq 0, X_{i,i}\leq 1} \mathrm{Tr}\left(\mathsf{C} \cdot \mathsf{X}\right) - \gamma \sum_{i,j} \left|\mathsf{C}_{i,j}\right|$.

Furthermore, we have $\|\mathsf{Y}\|_{\mathrm{op}} \leq O(\log(n)/\gamma)$ *(cf.,Lemma C.2.3 of [LP20]).*

Remark 5.4.20. *We observe that Theorem 5.4.19 differs from Theorem 1.1 of [LP20] only by an additional bound on* $\|\mathsf{Y}\|_{\mathrm{op}}$. *This bound is important in analyzing the error when approximating (matrix) exponential of* Y.

We now show how we can approximate the Gram vectors of the SDP solution of Theorem 5.4.19. We rely on part of the analysis in Arora–Kale [AK07].

Claim 5.4.21. *Let* $\mathsf{C} \in \mathbb{R}^{n\times n}$ *be a matrix with at most* m *non-zero entries and* $\gamma \in (0, 1/2]$. *Suppose* $\widetilde{X} = \mathsf{S} \cdot \exp \mathsf{Y} \cdot \mathsf{S}$ *satisfy the conclusions of Theorem 5.4.19 given* $\mathsf{C} \in \mathbb{R}^{n\times n}$ *and accuracy* γ. *Then with high probability we can find in* $\widetilde{O}(\mathrm{poly}(1/\gamma) \cdot (m+n))$ *time approximate Gram vectors* $u_1, \ldots, u_n \in \mathbb{R}^n$ *such that* $\widetilde{X}'_{i,j} := \left\langle u_i, u_j \right\rangle$ *satisfy*

- $\widetilde{X}'_{i,i} \leq 1$ *for every* $1 \leq i \leq n$, *and*
- $\mathrm{Tr}(\mathsf{C} \cdot \widetilde{X}') \geq \mathrm{Tr}\left(\mathsf{C} \cdot \widetilde{\mathsf{X}}\right) - \gamma \sum_{i,j} \left|\mathsf{C}_{i,j}\right|$.

Proof. Since $\widetilde{X} = (\mathsf{S} \cdot \exp(\mathsf{Y}/2))(\mathsf{S} \cdot \exp(\mathsf{Y}/2))^t$, the rows of $\mathsf{S} \cdot \exp(\mathsf{Y}/2)$ can be taken as Gram vectors $u_1, \ldots, u_n \in \mathbb{R}^n$ of $\widetilde{X}$. If we knew the rows of $\exp(\mathsf{Y}/2)$, we could readily recover these Gram vectors since S is diagonal. As observed in Arora–Kale [AK07], computing $\exp(\mathsf{Y}/2)$ may be computationally expensive, so instead one can approximate the matrix-vector product $\exp(\mathsf{Y}/2)u$ using $d = O(\log(n)/\gamma^2)$ random Gaussian vectors

$u \sim N(0, I_n)$. By the Johnson–Lindenstrauss Lemma and scaling by $\sqrt{n/d}$, with high probability we obtain vectors $\widetilde{u}_1, \ldots, \widetilde{u}_n$ satisfying for every $i, j \in [n]$ say

$$\left|\left\langle u_i, u_j \right\rangle - \left\langle \widetilde{u}_i, \widetilde{u}_j \right\rangle\right| \;\leq\; \frac{\gamma}{6}.$$

In particular, whp $\|\widetilde{u}_i\|_2^2 \leq 1 + \gamma/6$. Thus, by normalizing the vectors $\widetilde{u}_i$ with $\|\widetilde{u}_i\|_2 > 1$ to have ℓ_2-norm one the preceding approximation deteriorates to

$$\left|\left\langle u_i, u_j \right\rangle - \left\langle \widetilde{u}_i, \widetilde{u}_j \right\rangle\right| \;\leq\; \gamma/2.$$

To compute each the matrix-vector product $\exp(\mathsf{Y}/2)u$ in $\widetilde{O}(\mathrm{poly}(1/\gamma) \cdot (m+n))$, we rely on the following lemma.

Lemma 5.4.22 (Arora–Kale [AK07], *cf.*,Lemma 6)**.** *Let $\mathcal{T}_{\mathsf{Y}}$ be the time needed to compute the matrix-vector product $\mathsf{Y}u$. Then the vector $v := \sum_{i=0}^{k} \mathsf{Y}^i u/(i!)$ can be computed in $O(k \cdot \mathcal{T}_{\mathsf{Y}})$ time and if $k \geq \max\{e^2 \cdot \|\mathsf{Y}\|_{\mathrm{op}}, \ln(1/\delta)\}$, then $\|\exp(\mathsf{Y})u - v\|_2 \leq \delta$.*

By noting that $\|\mathsf{Y}\|_{\mathrm{op}} \leq O(\log(n)/\gamma)$ and the time $\mathcal{T}_{\mathsf{Y}}$ (*cf.*, Lemma 5.4.22) $\mathsf{Y}u$ is $\widetilde{O}((m+n)/\gamma)$, applying Lemma 5.4.22 with say $\delta \leq \mathrm{poly}(\gamma/n)$ we can approximate each $\exp(\mathsf{Y}/2)u$ in time $\widetilde{O}((m+n)/\gamma)$. Therefore, the total running is $\widetilde{O}(\mathrm{poly}(1/\gamma) \cdot (m+n))$ as claimed. Then the actual Gram vectors still satisfy

$$\left|\left\langle u_i, u_j \right\rangle - \left\langle \widetilde{u}_i, \widetilde{u}_j \right\rangle\right| \;\leq\; \gamma.$$

Hence, we get

$$\mathrm{Tr}(\mathsf{C} \cdot \widetilde{X}') \;\geq\; \mathrm{Tr}\left(\mathsf{C} \cdot \widetilde{\mathsf{X}}\right) - \gamma \sum_{i,j} \left|\mathsf{C}_{i,j}\right|,$$

concluding the proof. ∎

We are ready to prove Lemma 5.4.17 which is restated below for convenience.

Lemma 5.4.17. *[Sparse SDP Solver Wrapper based on [LP20] and partially on [AK07]] Let $\mathsf{C} \in \mathbb{R}^{n\times n}$ be a matrix with at most m non-zero entries. For every accuracy $\gamma > 0$, with high probability we can find in time $\widetilde{O}((m+n)/\mathrm{poly}(\gamma))$ vectors $u_1, \ldots, u_n \in \mathbb{R}^n$ in the unit ball (i.e., $\|u_i\| \leq 1$) such that that the matrix $\widetilde{\mathsf{X}}_{i,j} := \left\langle u_i, u_j \right\rangle$ satisfies*

$$\mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}\right) \;\geq\; \max_{\mathsf{X}\succeq 0, X_{i,i}\leq 1} \mathrm{Tr}\left(\mathsf{C}\cdot\mathsf{X}\right) - \gamma\sum_{i,j}\left|\mathsf{C}_{i,j}\right|.$$

Proof of Lemma 5.4.17. Follows by combining the SDP solution $\widetilde{X}$ of Theorem 5.4.19 with the fast approximate Gram vector computation of Claim 5.4.21, the latter yielding another approximated SDP solution $\widetilde{X}'$. In both of these computations, we use accuracy parameter $\gamma/2$ so that

$$\begin{aligned}\mathrm{Tr}(\mathsf{C}\cdot\widetilde{X}') \;&\geq\; \mathrm{Tr}\left(\mathsf{C}\cdot\widetilde{\mathsf{X}}\right) - \frac{\gamma}{2}\sum_{i,j}\left|\mathsf{C}_{i,j}\right| \\ &\geq\; \max_{\mathsf{X}\succeq 0, X_{i,i}\leq 1} \mathrm{Tr}\left(\mathsf{C}\cdot\mathsf{X}\right) - \frac{\gamma}{2}\sum_{i,j}\left|\mathsf{C}_{i,j}\right| - \frac{\gamma}{2}\sum_{i,j}\left|\mathsf{C}_{i,j}\right|.\end{aligned}$$

Moreover, each step takes $\widetilde{O}(\mathrm{poly}(1/\gamma)\cdot(m+n))$ which concludes the proof. ∎

5.5 Regularity Based Decoding for Direct-Sum Codes

We now develop list-decoding algorithms for direct-sum codes, using the regularity lemmas obtained in the previous section. We will prove the following theorem.

Theorem 5.5.1. *Let $\mathcal{C}_0 \subset \mathbb{F}_2^n$ be a code with $\mathrm{bias}(\mathcal{C}_0) \leq \varepsilon_0$, which is unique-decodable to distance $(1-\varepsilon_0)/4$ in time $\mathcal{T}_0$. Let $W \subseteq [n]^k$ be a d-regular, τ-splittable collection of tuples, and let $\mathcal{C} = \mathrm{dsum}_W(\mathcal{C}_0)$ be the corresponding direct-sum lifting of $\mathcal{C}_0$ with $\mathrm{bias}(\mathcal{C}) \leq \varepsilon$. Let β be such that*

$$\beta \;\geq\; \max\left\{\sqrt{\varepsilon},\; \left(2^{20}\cdot\tau\cdot k^3\right)^{1/2},\; 2\cdot\left(\frac{1}{2}+2\varepsilon_0\right)^{k/2}\right\}.$$

Then, there exists a randomized algorithm, which given $\tilde{y} \in \mathbb{F}_2^W$, recovers the list $\mathcal{L}_\beta(\tilde{y}) := \{y \in \mathcal{C} \mid \Delta(\tilde{y}, y) \leq 1/2 - \beta\}$ with probability $1 - o(1)$, in time $\tilde{O}(C_{\beta,k,\varepsilon_0} \cdot (|W| + \mathcal{T}_0))$, where $C_{k,\beta,\varepsilon_0} = (6/\varepsilon_0)^{2^{O(k^3/\beta^2)}}$.

To obtain the decoding algorithm, we first define a function $g : [n]^k \to \{-1, 1\}$ supported on W as

$$g(i_1, \ldots, i_k) := \begin{cases} (-1)^{\tilde{y}_{(i_1,\ldots,i_k)}} & \text{if } (i_1, \ldots, i_k) \in W \\ 0 & \text{otherwise} \end{cases}$$

For each $z \in \mathbb{F}_2^n$, we also consider the similar function $\chi_z : [n] \to \{-1, 1\}$ defined as $\chi_z(i) = (-1)^{z_i}$. We first re-state the decoding problem in terms of the functions g and χ_z.

Claim 5.5.2. *Let $z \in \mathbb{F}_2^n$, and let the functions g and χ_z be as above. Then,*

$$\Delta(\tilde{y}, \mathrm{dsum}_W(z)) \leq \frac{1}{2} - \beta \quad \Leftrightarrow \quad \left\langle g, \chi_z^{\otimes k} \right\rangle_{\mu_k} = \left(\frac{n}{d}\right)^{k-1} \cdot \left\langle g, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} \geq 2\beta\,.$$

Proof. We have

$$\begin{aligned} \Delta(\tilde{y}, \mathrm{dsum}_W(z)) &= \mathop{\mathbb{E}}_{(i_1,\ldots,i_k)\sim W} \left[\mathbb{1}_{\{\tilde{y}_{(i_1,\ldots,i_k)} \neq z_{i_1} + \cdots + z_{i_k} \mod 2\}} \right] \\ &= \mathop{\mathbb{E}}_{(i_1,\ldots,i_k)\sim \mu_k} \left[\frac{1 - g(i_1, \ldots, i_k) \cdot \prod_{t\in[k]} \chi_z(i_t)}{2} \right] = \frac{1}{2} - \frac{1}{2} \cdot \left\langle g, \chi_z^{\otimes k} \right\rangle_{\mu_k}. \end{aligned}$$

Finally, using the fact that g is only supported on W, and $|W| = d^{k-1} \cdot n$ by d-regularity, we have $\langle g, f \rangle_{\mu_k} = (n/d)^{k-1} \cdot \langle g, f \rangle_{\mu_1^{\otimes k}}$ for any function $f : [n]^k \to \mathbb{R}$. ∎

Note that each element of the list $\mathcal{L}_\beta(\tilde{y})$ must be equal to $\mathrm{dsum}_W(z)$ for some $z \in \mathcal{C}_0$. Thus, to search for all such z, we will consider the decomposition h of the function g,

given by Theorem 5.4.11 with respect to the class of functions $\mathcal{F} = \mathrm{CUT}_{\pm}^{\otimes k}$. Since the functions $\chi_z^{\otimes k}$ belong to $\mathcal{F}$, it will suffice to only consider the inner product $\left\langle h, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}}$.

Also, since the approximating function h is determined by a small number of functions, say $\{f_1, \ldots, f_r : [n] \to \{-1, 1\}\}$, it will suffice to (essentially) consider only the functions measurable in the factor $\mathcal{B}$ determined by $f_1, \ldots, f_r$. Recall that the factor $\mathcal{B}$ is simply a partition of $[n]$ in 2^r pieces according to the values of $f_1, \ldots, f_r$. Also, since any $\mathcal{B}$-measurable function is constant on each piece, it is completely specified by $|\mathcal{B}|$ real values. We will only consider functions taking values in $[-1, 1]$, and discretize this space to an appropriate accuracy η, to identify all relevant $\mathcal{B}$-measurable functions with the set $\{0, \pm\eta, \pm 2\eta, \ldots, \pm 1\}^{|\mathcal{B}|}$. The decoding procedure is described in the following algorithm.

Algorithm 5.5.3 (List Decoding).

Input $\tilde{y} \in \mathbb{F}_2^W$

Output *List* $\mathcal{L} \subseteq \mathcal{C}$

- *Obtain the approximator* h *given by Theorem 5.4.11 for* $\mathcal{F} = \mathrm{CUT}_{\pm}^{\otimes k}$, $\delta = \beta$, *and the function* $g : [n]^k \to \{-1, 1\}$ *defined as*

$$g(i_1, \ldots, i_k) := \begin{cases} (-1)^{\tilde{y}_{(i_1,\ldots,i_k)}} & \text{if } (i_1, \ldots, i_k) \in W \\ 0 & \text{otherwise} \end{cases}$$

- *Let* h *be of the form* $h = \sum_{j=1}^{p} c_j \cdot f_{j_1} \otimes \cdots \otimes f_{j_k}$, *with each* $f_{j_t} : [n] \to \{-1, 1\}$. *Let* $\mathcal{B}$ *be the factor determined by the functions* $\left\{ f_{j_t} \right\}_{j \in [p], t \in [k]}$.

- *Let* $\mathcal{L} = \emptyset$ *and let* $\eta = 1/\lceil (2/\varepsilon_0) \rceil$. *For each* $\mathcal{B}$*-measurable function* $\overline{f}$ *given by a value in* $D_\eta := \{0, \pm\eta, \pm 2\eta, \ldots, \pm 1\}$ *for every atom of* $\mathcal{B}$*:*

  – *Sample a random function* $\chi : [n] \to \{-1, 1\}$ *by independently sampling* $\chi(i) \in \{-1, 1\}$ *for each* i, *such that* $\mathbb{E}[\chi(i)] = \overline{f}(i)$. *Take* $\tilde{z} \in \mathbb{F}_2^n$ *to be such that* $\chi = \chi_{\tilde{z}}$.

  – *If there exists* $z \in \mathcal{C}_0$ *such that*

$$\Delta(\tilde{z}, z) \leq \frac{(1 - \varepsilon_0)}{4} \quad \text{and} \quad \Delta(\tilde{y}, \mathrm{dsum}_W(z)) \leq \frac{1}{2} - \beta\,,$$

  then $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathrm{dsum}_W(z)\}$.

- *Return* $\mathcal{L}$.

Note that by our choice of the β in Theorem 5.5.1, we have that $\tau \leq \beta^2/(2^{20}k^3)$. Thus, we can indeed apply Theorem 5.4.11 to obtain the function h as required by the algorithm. To show that the algorithm can recover the list, we will need to show that for each z such that $\mathrm{dsum}_W(z) \in \mathcal{L}_\beta$, the sampling procedure finds a $\tilde{z}$ close to z with

significant probability. To analyze this probability, we first prove the following claim.

Claim 5.5.4. *Let $z \in \mathbb{F}_2^n$ and let $\overline{f} : [n] \to D_\eta$ be a minimizer of $\|\mathbb{E}[\chi_z|\mathcal{B}] - \overline{f}\|_\infty$ among all $\mathcal{B}$-measurable functions in $D_\eta^{|\mathcal{B}|}$. Then, over the random choice of χ such that $\mathbb{E}[\chi] = \overline{f}$, we have*

$$\mathop{\mathbb{E}}_{\chi}\left[\langle \chi, \chi_z\rangle_{\mu_1}\right] \;=\; \left\langle \overline{f}, \chi_z\right\rangle_{\mu_1} \;\geq\; \|\mathbb{E}\left[\chi_z|\mathcal{B}\right]\|_{\mu_1}^2 - \eta\,.$$

Proof. By linearity of the inner product, we have

$$\mathop{\mathbb{E}}_{\chi}\left[\langle \chi, \chi_z\rangle_{\mu_1}\right] \;=\; \langle \mathbb{E}\left[\chi\right], \chi_z\rangle_{\mu_1} \;=\; \left\langle \overline{f}, \chi_z\right\rangle_{\mu_1} \;=\; \left\langle \overline{f}, \mathbb{E}\left[\chi_z|\mathcal{B}\right]\right\rangle_{\mu_1},$$

where the last equality used Proposition 5.3.14 and the fact that $\overline{f}$ is $\mathcal{B}$-measurable. Since $\mathbb{E}[\chi_z|\mathcal{B}]$ takes values in $[-1,1]$ and $\overline{f}$ is the minimizer over all functions in $D_\eta^{|\mathcal{B}|}$, we must have $\|\mathbb{E}[\chi_z|\mathcal{B}] - \overline{f}\|_\infty \leq \eta$. Using this pointwise bound, we get

$$\begin{aligned}\left\langle \overline{f}, \mathbb{E}\left[\chi_z|\mathcal{B}\right]\right\rangle_{\mu_1} &= \mathop{\mathbb{E}}_{i\sim\mu_1}\left[\overline{f}(i)\cdot \mathbb{E}\left[\chi_z|\mathcal{B}\right](i)\right] \\ &\geq \mathop{\mathbb{E}}_{i\sim\mu_1}\left[\left(\mathbb{E}\left[\chi_z|\mathcal{B}\right](i)\right)^2 - \eta\cdot\left|\mathbb{E}\left[\chi_z|\mathcal{B}\right](i)\right|\right] \;\geq\; \|\mathbb{E}\left[\chi_z|\mathcal{B}\right]\|_{\mu_1}^2 - \eta\,. \qquad \blacksquare\end{aligned}$$

We next show that when $z \in \mathbb{F}_2^n$ is such that $\left\langle g, \chi_z^{\otimes k}\right\rangle$ is large, then the norm of the conditional expectation $\mathbb{E}[\chi_z|\mathcal{B}]$ is also large, and hence the sampling procedure finds a $\tilde{z}$ close to z. When we have a $z \in \mathcal{C}_0$ with such a property, we can use $\tilde{z}$ to recover z using the unique decoding algorithm for $\mathcal{C}_0$.

Lemma 5.5.5. *Let $z \in \mathbb{F}_2^n$ be such that*

$$\left\langle g, \chi_z^{\otimes k}\right\rangle_{\mu_k} = \left(\frac{n}{d}\right)^{k-1}\cdot\left\langle g, \chi_z^{\otimes k}\right\rangle_{\mu_1^{\otimes k}} \geq\; 2\beta\,.$$

Then, we have $\|\mathbb{E}[\chi_z|\mathcal{B}]\|_{\mu_1}^2 \geq (\beta/2)^{2/k}$.

Proof. Let h be the approximating function obtained by applying Theorem 5.4.11 to g with approximation error $\delta = \beta$. Note that we have $\|h\|_{\mu_1^{\otimes k}} \leq 2$, and for any $f \in \mathrm{CUT}_{\pm}^{\otimes k}$,

$$\left(\frac{n}{d}\right)^{k-1} \cdot \left\langle g - \left(\frac{d}{n}\right)^{k-1} \cdot h \,, f \right\rangle_{\mu_1^{\otimes k}} \;\leq\; \delta \,.$$

Using $f = \chi_z^{\otimes k}$ and $\delta = \beta$, we get

$$\left\langle h, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} \;\geq\; 2\beta - \delta \;\geq\; \beta \,.$$

Using Proposition 5.3.14, and the fact that $\mathcal{B}$ is defined so that all functions in the decomposition of h are (by definition) $\mathcal{B}$-measurable, we have

$$\left\langle h, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} \;=\; \sum_{j=1}^{p} c_j \prod_{t=1}^{k} \langle f_{j_t}, \chi_z \rangle_{\mu_1} \;=\; \sum_{j=1}^{p} c_j \prod_{t=1}^{k} \langle f_{j_t}, \mathbb{E}\left[\chi_z | \mathcal{B}\right] \rangle_{\mu_1} \;=\; \left\langle h, \left(\mathbb{E}\left[\chi_z | \mathcal{B}\right]\right)^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} .$$

Combining the above with Cauchy-Schwarz, we get

$$\beta \;\leq\; \left\langle h, \chi_z^{\otimes k} \right\rangle_{\mu_1^{\otimes k}} \;\leq\; \|h\|_{\mu_1^{\otimes k}} \cdot \left\| \left(\mathbb{E}\left[\chi_z | \mathcal{B}\right]\right)^{\otimes k} \right\|_{\mu_1^{\otimes k}} \;=\; \|h\|_{\mu_1^{\otimes k}} \cdot \|\mathbb{E}\left[\chi_z | \mathcal{B}\right]\|_{\mu_1}^{k} \,.$$

Using $\|h\|_{\mu_1^{\otimes k}} \leq 2$ then gives $\|\mathbb{E}[\chi_z|\mathcal{B}]\|_{\mu_1}^2 \geq (\beta/2)^{2/k}$. ∎

Using the above results, we can now complete the analysis of the algorithm.

Proof of Theorem 5.5.1. We first argue that for any codeword $z \in \mathcal{C}_0$ such that $\mathrm{dsum}_W(z) \in \mathcal{L}_\beta$, sampling a random function χ (with $\mathbb{E}[\chi] = \overline{f}$ for an appropriate $\overline{f}$) finds a $\tilde{z}$ close to z with significant probability. Let $\overline{f} \in D_\eta^{\mathcal{B}}$ be the minimizer of $\|\chi_z - \overline{f}\|_\infty$, for such a $z \in \mathcal{C}_0$. We have by Claim 5.5.4 that $\mathbb{E}_\chi[\langle \chi, \chi_z \rangle_{\mu_1}] \geq \|\mathbb{E}[\chi_z|\mathcal{B}]\|_{\mu_1}^2 - \eta$. Since $\Delta(\tilde{y}, \mathrm{dsum}_W(z)) \leq 1/2 - \beta$, we have by Claim 5.5.2 that $\left\langle g, \chi_z^{\otimes k} \right\rangle_{\mu_k} \geq 2\beta$. Thus, by Lemma 5.5.5, we have that $\|\mathbb{E}[\chi_z|\mathcal{B}]\|_{\mu_1}^2 \geq (\beta/2)^{2/k}$. Combining these, and using the lower bound on β, we get

that

$$\mathop{\mathbb{E}}_{\chi}\left[\langle \chi, \chi_z\rangle_{\mu_1}\right] \;\geq\; \left(\frac{\beta}{2}\right)^{2/k} - \eta \;\geq\; \frac{1}{2} + 2\varepsilon_0 - \eta \;\geq\; \frac{1}{2} + \frac{3\varepsilon_0}{2}\,.$$

Since $\langle \chi, \chi_z\rangle_{\mu_1}$ is the average of n independent (not necessarily identical) random variables $\{\chi(i)\cdot\chi_z(i)\}_{i\in[n]}$ in the range $[-1,1]$, we get by Hoeffding's inequality that

$$\mathop{\mathbb{P}}_{\chi}\left[\langle \chi, \chi_z\rangle_{\mu_1} \leq \frac{1}{2} + \varepsilon_0\right] \;\leq\; \mathop{\mathbb{P}}_{\chi}\left[\left|\langle \chi, \chi_z\rangle_{\mu_1} - \mathop{\mathbb{E}}_{\chi}\left[\langle \chi, \chi_z\rangle_{\mu_1}\right]\right| \geq \frac{\varepsilon_0}{2}\right] \;\leq\; 2\cdot\exp\left(-\varepsilon_0^2\cdot n/8\right)\,.$$

Thus, given a good sample χ satisfying $\langle \chi, \chi_z\rangle_{\mu_1} \geq 1/2 + \varepsilon_0$, we can recover the above $z \in \mathcal{C}_0$ such that $\mathrm{dsum}_W(z) \in \mathcal{L}_\beta$, via the unique decoding algorithm for $\mathcal{C}_0$. Also, given the right $\overline{f}$, we sample a good χ with probability at least $1 - 2\cdot\exp(-\varepsilon_0^2\cdot n/8)$. A union bound then gives

$$\mathbb{P}\left[\mathcal{L} = \mathcal{L}_\beta\right] \;\geq\; 1 \;-\; \left|\mathcal{L}_\beta\right|\cdot 2\cdot\exp(-\varepsilon_0^2\cdot n/8)\,.$$

Using $\beta \geq \sqrt{\varepsilon}$, we get that $\left|\mathcal{L}_\beta\right| \leq (1/\varepsilon)$ by the Johnson bound, which yields the desired probability bound[13].

Running time. Using Theorem 5.4.11, the regularity decomposition h can be computed in time $\tilde{O}(C_{\beta,k,\varepsilon_0}\cdot|W|)$. Given the functions $f_1,\ldots,f_r$ forming the decomposition h, the factor $\mathcal{B}$ can be computed in time $O(2^r\cdot n)$. For a chosen $\overline{f}$ in the sampling step, a sample χ can be computed in time $O(n)$, and the decoding problem for the corresponding $\tilde{z}$ can be solved in time $\mathcal{T}_0$. Also, the distance $\Delta(\tilde{y}, \mathrm{dsum}_W(z))$ can be computed in time $O(|W|)$. Since the total number of sampling steps is at most $(3/\eta)^{|B|}$ and the number of functions in the decomposition h is $O(k^3/\beta^2)$ from Theorem 5.4.11, we get that the total number of

13. We thank the announymous reviewer for pointing out that this last part of the proof works whenever the choice of β ensures that $\left|\mathcal{L}_\beta\right| = O_{\beta,\varepsilon_0}(1)$. Thus, Theorem 5.5.1 can be adpated to this more general condition, the original Johnson bound regime condition (i.e., $\beta \geq \sqrt{\varepsilon}$) providing just a sufficient condition.

sampling steps is $(6/\varepsilon_0)^{2^{O(k^3/\beta^2)}}$. Thus, the total running time is bounded by $\tilde{O}(C_{\beta,k,\varepsilon_0} \cdot (|W| + \mathcal{T}_0))$, where $C_{\beta,k,\varepsilon_0} = (6/\varepsilon_0)^{2^{O(k^3/\beta^2)}}$. ∎

5.6 Near-linear Time Decoding of Ta-Shma's Codes

We now proceed to prove our main result, namely Theorem 5.1.1, which establishes a near-linear time *unique* decoding algorithm for Ta-Shma's codes [TS17]. It will follow from the regularity based list decoding algorithm for direct sum codes, Theorem 5.5.1, applied to the decoding of a slight modification of Ta-Shma's construction from [JQST20] that yields a splittable collection of tuples for the direct sum.

Theorem 5.1.1 (Near-linear Time Unique Decoding). *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\alpha} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\alpha})$ where $\alpha = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) an $r(\varepsilon) \cdot \tilde{O}(N)$ time unique decoding algorithm that that decodes within radius $1/4 - \varepsilon/4$ and works with high probability,

where $r(\varepsilon) = \exp(\exp(\text{polylog}(1/\varepsilon)))$.

We now state the properties and guarantees needed in our work of this slightly modified version of Ta-Shma's direct sum construction of near optimal ε-balanced codes. To make the decoding task more transparent, we will additionally require the base code in Ta-Shma's construction have the following technical property.

Definition 5.6.1. *We say that a code has symbol multiplicity $m \in \mathbb{N}$ if it can be obtained from another code by repeating each symbol of its codeword m times.*

Theorem D.1.1. *[Ta-Shma's Codes (implicit in [TS17])] Let $c > 0$ be an universal constant. For every $\varepsilon > 0$ sufficiently small, there exists $k = k(\varepsilon)$ satisfying $\Omega(\log(1/\varepsilon)^{1/3}) \leq k \leq O(\log(1/\varepsilon))$, $\varepsilon_0 = \varepsilon_0(\varepsilon) > 0$, and positive integer $m = m(\varepsilon) \leq (1/\varepsilon)^{o(1)}$ such that Ta-Shma's construction yields a collection of τ-splittable tuples $W = W(k) \subseteq [n]^k$ satisfying:*

(i) For every linear ε_0-balanced code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ with symbol multiplicity m, the direct sum code $\mathrm{dsum}_W(\mathcal{C}_0)$ is:

(i.1) ε-balanced (parity sampling).

(i.2) if $\mathcal{C}_0$ has rate $\Omega(\varepsilon_0^c/m)$, then $\mathrm{dsum}_W(\mathcal{C}_0)$ has rate $\Omega(\varepsilon^{2+o(1)})$ (near optimal rate)

(ii) $\tau \leq \exp(-\Theta(\log(1/\varepsilon)^{1/6}))$ (splittability).

(iii) W is constructible in $\mathrm{poly}(|W|)$ time (explicit construction).

Ta-Shma's construction is based on a generalization of the zig-zag product of Reingold, Vadhan and Wigderson [RVW00]. To make the exposition more self-contained, we recall the slight modification from [JQST20] in Appendix D.1, but it is not exhaustive exposition. The interested reader is referred to Ta-Shma [TS17] for the original construction for aspects not covered here.

Ta-Shma's code construction requires an ε_0-balanced base code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ whose distance will be amplified by taking the direct sum with a carefully chosen collection of tuples W yielding an ε-balanced code $\mathcal{C} = \mathrm{dsum}_W(\mathcal{C}_0)$. Since our goal is to achieve near-linear time encoding and decoding of $\mathcal{C}$, we take an "off-the-shelf" base code $\mathcal{C}_0$ that is linear time encodable and decodable (near-linear time also suffices). A convenient choice is the linear binary code family of Guruswami–Indyk [GI05] that can be encoded and decoded in linear time. The rate versus distance trade-off is at the so-called Zyablov bound. In particular, it yields codes of distance $1/2 - \varepsilon_0$ with rate $\Omega(\varepsilon_0^3)$, but for our applications rate $\mathrm{poly}(\varepsilon_0)$ suffices (or with some extra steps even any rate depending only on ε_0

suffices, see Remark 5.6.5). We will use Corollary 5.6.2 implicit in [GI05].

Corollary 5.6.2. *[Implicit in Guruswami–Indyk [GI05]] For every $\varepsilon_0 > 0$, there exists a family of ε_0-balanced binary linear codes $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ of rate $\Omega(\varepsilon_0^3)$ which can be encoded in $O_{\varepsilon_0}(n)$ time and can be decoded in $O(\exp(\mathrm{poly}(1/\varepsilon_0)) \cdot n)$ time from up to a fraction $1/4 - \varepsilon_0$ of errors. Furthermore, every code in the family is explicitly specified given a binary linear code of block-length $\mathrm{poly}(1/\varepsilon_0)$ which can be constructed in probabilistic $O(\mathrm{poly}(1/\varepsilon_0))$ or deterministic $2^{O(\mathrm{poly}(1/\varepsilon_0))}$ time.*

We first prove the (gentle) *list* decoding result of Ta-Shma's codes.

Theorem 5.1.2 (Near-linear Time Gentle List Decoding)**.** *For every $\varepsilon > 0$ sufficiently small, there are explicit binary linear Ta-Shma codes $\mathcal{C}_{N,\varepsilon,\alpha} \subseteq \mathbb{F}_2^N$ for infinitely many values $N \in \mathbb{N}$ with*

(i) distance at least $1/2 - \varepsilon/2$ (actually ε-balanced),

(ii) rate $\Omega(\varepsilon^{2+\alpha})$ where $\alpha = O(1/(\log_2(1/\varepsilon))^{1/6})$, and

(iii) an $r(\varepsilon) \cdot \tilde{O}(N)$ time list decoding algorithm that decodes within radius $1/2 - 2^{-\Theta((\log_2(1/\varepsilon))^{1/6})}$ and works with high probability,

where $r(\varepsilon) = \exp(\exp(\mathrm{poly}(1/\varepsilon)))$.

Proof. We start by dealing with a simple technical issue of making the base code in Ta-Shma's construction have the required symbol multiplicity. Let $\mathcal{C}_0' \subseteq \mathbb{F}_2^{n'}$ be an ε_0-balanced code from Corollary 5.6.2 which we will use to obtain a base code in Ta-Shma's construction where $\varepsilon_0 > 0$ is a suitable value prescribed by this construction.

Ta-Shma's construction then takes $\mathcal{C}_0' \subseteq \mathbb{F}_2^{n'}$ and forms a new code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ by repeating each codeword symbol $m \leq (1/\varepsilon)^{o(1)}$ times. By Claim 5.6.6, $\mathcal{C}_0$ is an ε_0-balanced code that can be unique decoded within the same (fractional) radius of $\mathcal{C}_0'$ in

time $\mathcal{T}_0(n) = r \cdot \mathcal{T}_0(n') + \widetilde{O}(r^2 \cdot n')$, where $\mathcal{T}_0(n')$ is the running time of an unique decoder for $\mathcal{C}_0'$. Since by Corollary 5.6.2 $\mathcal{T}_0(n') = O(\exp(\mathrm{poly}(1/\varepsilon_0)) \cdot n')$ and $\varepsilon_0 \gg \varepsilon$, the decoding time of $\mathcal{C}_0$ becomes $\mathcal{T}_0(n) = O(\exp(\mathrm{poly}(1/\varepsilon)) \cdot n)$.

Let $W = W(k)$ be a collection of tuples from Ta-Shma's construction Theorem D.1.1 so that $\mathcal{C} = \mathrm{dsum}_W(\mathcal{C}_0)$ is ε-balanced, $\tau \leq \exp(-\Theta(\log(1/\varepsilon)^{1/6}))$ and $k = \Omega(\log(1/\varepsilon)^{1/3})$. We will invoke our list decoding algorithm Theorem 5.5.1 whose list decoding radius $1/2 - \beta$ has to satisfy

$$\beta \;\geq\; \max\left\{\sqrt{\varepsilon},\; \left(2^{20} \cdot \tau \cdot k^3\right)^{1/2},\, 2 \cdot \left(\frac{1}{2} + 2\varepsilon_0\right)^{k/2}\right\}.$$

Using our values of τ and k together with the fact that $\varepsilon_0 < 1$ is bounded away from 1 by a constant amount gives

$$\beta \;\geq\; \max\left\{\sqrt{\varepsilon},\, ,\; \exp(-\Theta((\log(1/\varepsilon))^{1/6})), \exp(-\Theta((\log(1/\varepsilon))^{1/3}))\right\}.$$

Hence, we can take $\beta = \exp(-\Theta(\log(1/\varepsilon)^{1/6}))$. Now, we compute the list decoding running proving a (crude) upper bound on its dependence on ε. By Theorem 5.5.1, the list decoding time

$$\tilde{O}(C_{\beta,k,\varepsilon_0} \cdot (|W| + \mathcal{T}_0(n))),$$

where $C_{k,\beta,\varepsilon_0} = (6/\varepsilon_0)^{2^{O(k^3/\beta^2)}}$. For our choices of parameters, this decoding time can be (crudely) bounded by $\tilde{O}(\exp(\exp(\mathrm{poly}(1/\varepsilon))) \cdot N)$. ∎

The gentle *list* decoding theorem above readily implies our main result for *unique* decoding if we are only interested in $\widetilde{O}_\varepsilon(N)$ decoding time without a more precise dependence on ε. We prove our main result, Theorem 5.1.1, for *unique* decoding making more precise the dependence of the running time on ε.

Proof. Proof of Theorem 5.1.1 We proceed as in the proof of Theorem 5.1.2 expect that we take $\beta = 1/4$ in the list decoding radius $1/2 - \beta$ so that by performing list decoding we can recover all codewords in the unique decoding radius of the corrupted codeword regardless of the bias of the code $\mathcal{C}_{N,\varepsilon,\alpha}$.

We now recompute the running time. By Theorem 5.5.1, the list decoding time

$$\tilde{O}(C_{\beta,k,\varepsilon_0} \cdot (|W| + \mathcal{T}_0(n))),$$

where $C_{k,\beta,\varepsilon_0} = (6/\varepsilon_0)^{2^{O(k^3/\beta^2)}}$. For our choices of parameters, this decoding time can be (crudely) bounded by $\tilde{O}(\exp(\exp(\text{polylog}(1/\varepsilon))) \cdot N)$. ∎

5.6.1 Choosing the Base Code

We now describe the (essentially) "off-the-shelf" base codes from Guruswami and Indyk [GI05] which we use in Ta-Shma's construction. We will need to prove that balanced codes can be easily obtained from [GI05]. The argument is quite simple and borrows from standard considerations related to the Zyablov and Gilbert–Varshamov bounds.

Corollary 5.6.2. *[Implicit in Guruswami–Indyk [GI05]] For every $\varepsilon_0 > 0$, there exists a family of ε_0-balanced binary linear codes $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ of rate $\Omega(\varepsilon_0^3)$ which can be encoded in $O_{\varepsilon_0}(n)$ time and can be decoded in $O(\exp(\text{poly}(1/\varepsilon_0)) \cdot n)$ time from up to a fraction $1/4 - \varepsilon_0$ of errors. Furthermore, every code in the family is explicitly specified given a binary linear code of block-length $\text{poly}(1/\varepsilon_0)$ which can be constructed in probabilistic $O(\text{poly}(1/\varepsilon_0))$ or deterministic $2^{O(\text{poly}(1/\varepsilon_0))}$ time.*

Theorem 5.6.3 (Guruswami–Indyk [GI05], *cf.*,Theorem 5)**.** *For every $\gamma > 0$ and for every $0 < R < 1$, there exists a family of binary linear* concatenated *codes of rate R, which can be*

encoded in linear time and can be decoded in linear time from up to a fraction e of errors, where

$$e \geq \max_{R<r<1} \frac{(1-r-\gamma)\cdot H_2^{-1}(1-R/r)}{2}. \tag{5.4}$$

$H_2^{-1}(x)$ is defined as the unique ρ in the range $0 \leq \rho \leq 1/2$ satisfying $H_2(\rho) = x$. Every code in the family is explicitly specified given a constant sized binary linear code which can be constructed in probabilistic $O(\log(1/\gamma)R^{-1}/\gamma^4)$ or deterministic $2^{O(\log(1/\gamma)R^{-1}/\gamma^4)}$ time [14].

As stated the codes in Theorem 5.6.3 are not necessarily balanced. We will see shortly that this can be easily achieved by choosing balanced inner codes in the concatenated code construction of Guruswami–Indyk [GI05]. To compute bounds on the parameters, we will use the following property about binary entropy.

Fact 5.6.4 ([GRS19],*cf.*,Lemma 3.3.7 abridged)**.** *Let H_2^{-1} be the inverse of the restriction of H_2 to $[0,1/2]$ (where H_2 is bijective). For every small enough $\varepsilon > 0$,*

$$H_2^{-1}(x-\varepsilon^2/C_2) \geq H_2^{-1}(x) - \varepsilon,$$

where C_2 is a constant.

Proof of Corollary 5.6.2. To achieve a final binary code of rate R, Guruswami and Indyk [GI05] concatenate an outer code of rate $r > R$ and distance $1-r-\gamma$ (over a non-binary alphabet of size $O_\gamma(1)$) with an inner binary linear code of rate R/r at the GV bound whose distance $\rho \in [0,1/2]$ satisfy $R/r = 1 - H_2(\rho)$ (since it is at the GV bound), or equivalently $\rho = H_2^{-1}(1-R/r)$. By choosing $\gamma = \Theta(\varepsilon_0)$ and $R = \Theta(\varepsilon_0^3)$ in Theorem 5.6.3, the decoding error e can be lower bounded by letting $r = \Theta(\varepsilon_0)$ so that Fact 5.6.4 implies

14. Note that dependence $\log(1/\gamma)R^{-1}/\gamma^4$ is slightly worse than that claimed in [GI05], but not qualitatively relevant here nor in [GI05].

that Eq. (5.4) becomes

$$e \;\geq\; \max_{R<r<1} \frac{(1-r-\gamma)\cdot H_2^{-1}(1-R/r)}{2} \;\geq\; \frac{1}{4}-\varepsilon_0.$$

To obtain codes that are ε_0-balanced, we require that the inner codes used in this code concatenation not only lie on the Gilbert–Varshamov bound but are also balanced. It is well known that with high probability a random binary linear code at the GV bound designed to have minimum distance $1/2-\gamma/2$ also has maximum distance at most $1/2+\gamma/2$, i.e., the code is γ-balanced. Therefore, we assume that our inner codes are balanced.

For our concrete choices of parameters, $\rho = 1/2-\Theta(\varepsilon_0)$ and we also require the inner code to be $\Theta(\varepsilon_0)$-balanced. Note that any non-zero codeword of the concatenated is obtained as follows: each of the $\geq (1-r-\gamma)$ non-zero symbols of the outer codeword is replaced by an inner codeword of bias bias $\Theta(\varepsilon_0)$ and the remaining $\leq r+\gamma$ zero symbols are mapped to zero (since the inner code is linear). Hence, the bias of the concatenated codeword is at most

$$(1-r-\gamma)\cdot\Theta(\varepsilon_0) \;+\; 1\cdot(r+\gamma),$$

which can be taken to be ε_0 by suitable choices of hidden constants. ∎

Remark 5.6.5. *Guruswami–Indyk [GI05] codes have several nice properties making them a convenient choice for base codes in Ta-Shma's construction, but they are not crucial here. We observe that for our purposes we could have started with any family of good binary linear codes admitting near-linear time encoding and decoding. From this family, we could boost its distance using a simpler version of Ta-Shma's construction (rounds I and II of [JQST20][Section 8]) and our near-linear time decoder Theorem 5.5.1 for direct sum. This would result in an alternative family of linear binary ε_0-balanced codes of rate $\Omega(\varepsilon_0^{2+\alpha})$, for some arbitrarily small constant $\alpha > 0$, that can be encoded and decoded in near-linear time. We also point out that for these base codes any rate* $\mathrm{poly}(\varepsilon_0)$ *suffices our purposes.*

To handle the technical requirement of a base code in Ta-Shma's construction having a symbol multiplicity property (*cf.*, Definition 5.6.1), we use the following observation.

Claim 5.6.6. *Let $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ be an ε_0-balanced linear code of dimension D_0. Suppose that $\mathcal{C}_0$ is uniquely decodable within (fractional) radius $\delta_0 \in (0,1]$ in time $\mathcal{T}_0(n)$. Let $m \in \mathbb{N}$ and $\mathcal{C} \subseteq \mathbb{F}_2^{m \cdot n}$ be the code formed by replicating m times each codeword from $\mathcal{C}_0$, i.e.,*

$$\mathcal{C} \coloneqq \{z_1 \cdots z_m \in \mathbb{F}_2^{m \cdot n} \mid z_1 = \cdots = z_m \in \mathcal{C}_0\}.$$

Then, $\mathcal{C}$ is an ε_0-balanced linear code of dimension D_0 that can be uniquely decoded within (fractional) radius δ_0 in time $m \cdot \mathcal{T}_0(n) + \widetilde{O}(m^2 \cdot n)$.

Proof. The only non-immediate property is the unique decoding guarantees of $\mathcal{C}$. Given $\tilde{y} \in \mathbb{F}_2^{m \cdot n}$ within δ_0 (relative) distance of $\mathcal{C}$. Let β_i be the fraction of errors in the ith $\mathbb{F}_2^n$ component $\tilde{y}$. By assumption $\mathbb{E}_{i \in [m]} \beta_i \leq \delta_0$, so there is at least one of such component that can be correctly uniquely decoded. We issue unique decoding calls for $\mathcal{C}_o$ on each component $i \in [m]$. For each successful decoding say $z \in \mathcal{C}_0$, we let $y = z \ldots z \in \mathbb{F}_2^{m \cdot n}$ and check whether $\Delta(\tilde{y}, y) \leq \delta_0$ returning y if this succeeds. Finally, observe that this procedure indeed takes at most the claimed running time. ∎

REFERENCES

[AAV13] Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: The quantum pcp conjecture. *SIGACT News*, 44(2), June 2013. 85, 88

[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. 91, 92, 100, 159, 186, 187, 270

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992. 184, 253, 376

[AJQ+20] Vedat Levi Alev, Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. List decoding of direct sum codes. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms*, pages 1412–1425. SIAM, 2020. 3, 7, 14, 187, 188, 193, 194, 195, 198, 199, 211, 213, 217, 228, 241, 245, 252, 253, 263, 264, 267, 268, 272, 274, 284

[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, 7, 93, 96, 106, 108, 109, 110, 121, 122, 129, 152, 153, 168, 169, 170, 173, 175, 178, 195, 245, 250, 276

[AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, STOC '07, pages 227–236, 2007. 279, 289, 305, 309, 310, 313, 314, 315, 316

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008. 9

[ALGV18] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: High-dimensional walks and an FPRAS for counting bases of a matroid. *arXiv preprint arXiv:1811.01816*, 2018. 14

[AN04] Noga Alon and Assaf Naor. Approximating the cut-norm via grothendieck's inequality. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 72–80, 2004. 279, 289, 305, 309, 310, 311, 312, 313

[Ari09] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009. 190

[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. 91, 186, 270

[BATS08] Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-ramanujan graphs using the zig-zag product. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 325–334, 2008. 5

[BH13] Fernando G. S. L. Brandão and Aram Wettroth Harrow. Product-state approximations to quantum ground states. In *Proceedings of the 45th ACM Symposium on Theory of Computing*, pages 871–880, 2013. 14, 84, 86, 87, 88, 89

[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. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer Science*, pages 428–437, 2016. 103

[Bil95] Patrick Billingsley. *Probability and Measure*. J. Wiley and Sons, 1995. 286

[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. 103

[BL18] A. Bhowmick and S. Lovett. The list decoding radius for Reed–Muller codes over small fields. *IEEE Transactions on Information Theory*, 64(6):4382–4391, 2018. 275

[Bog12] Andrej Bogdanov. A different way to improve the bias via expanders. Lecture notes, April 2012. URL: `http://www.cse.cuhk.edu.hk/~andrejb/csc5060/notes/12L12.pdf`. 187, 193, 212, 270

[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. 1, 2, 6, 9, 11, 12, 13, 14, 19, 22, 23, 24, 25, 27, 64, 65, 66, 70, 71, 72, 79, 82, 86, 88, 101, 108, 129, 152, 153, 169, 175, 178, 196, 238, 245, 247, 249, 250, 275, 348, 349, 350, 352

[BS02] M. Bernstein and N. J. A. Sloane. Some Canonical Sequences of Integers. *arXiv Mathematics e-prints*, page math/0205301, May 2002. `arXiv:math/0205301`. 37

[BSHR05] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005. 9

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, November 2015. 291

[BV20] Greg Bodwin and Santosh Vempala. A unified view of graph regularity via matrix decompositions, 2020. arXiv:1911.11868. 273, 274, 275

[Cha16] Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3), August 2016. 91, 186, 270

[COCF09] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, page 207–216, 2009. 273

[CTZ18] David Conlon, Jonathan Tidor, and Yufei Zhao. Hypergraph expanders of all uniformities from Cayley graphs. *arXiv e-prints*, page arXiv:1809.06342, September 2018. arXiv:1809.06342. 11

[DD19] Yotam Dikstein and Irit Dinur. Agreement testing theorems on layered set systems. In *Proceedings of the 60th IEEE Symposium on Foundations of Computer Science*, 2019. 2, 14, 276

[DDFH18] Yotam Dikstein, Irit Dinur, Yuval Filmus, and Prahladh Harsha. Boolean function analysis on high-dimensional expanders. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, pages 38:1–38:20, 2018. 10, 11, 13, 14, 16, 26, 41, 42, 43, 44, 45, 47, 48, 49, 53, 54, 353, 354

[DDG+15] Roee David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. In *ITCS '15*, pages 327–336, New York, NY, USA, 2015. ACM. 91, 186, 270

[DDHRZ20] Yotam Dikstein, Irit Dinur, Prahladh Harsha, and Noga Ron-Zewi. Locally testable codes via high-dimensional expanders. *arXiv preprint arXiv:2005.01045*, 2020. 276

[Del75] P. Delsarte. The association schemes of coding theory. In *Combinatorics*, pages 143–161. Springer Netherlands, 1975. 184

[DFHT21] Irit Dinur, Yuval Filmus, Prahladh Harsha, and Madhur Tulsiani. Explicit SoS Lower Bounds from High-Dimensional Expanders. In *ITCS*, 2021. 3

[DHK+18] Irit Dinur, Prahladh Harsha, Tali Kaufman, Inbal Livni Navon, and Amnon Ta-Shma. List decoding with double samplers. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:136, 2018. 10, 11

[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. 92, 94, 96, 112, 114, 159, 165, 187, 276

[DK12] Irit Dinur and Tali Kaufman. Locally testable codes and expanders. Manuscript, 2012. 10

[DK17] Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 974–985, 2017. 1, 6, 7, 10, 11, 13, 16, 18, 25, 93, 111, 112, 113, 186, 270, 276

[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. 91, 186, 270

[EK16a] Shai Evra and Tali Kaufman. Bounded degree cosystolic expanders of every dimension. In *Proceedings of the 48th ACM Symposium on Theory of Computing*, page 36–48, 2016. 1

[EK16b] Shai Evra and Tali Kaufman. Bounded degree cosystolic expanders of every dimension. In *Proceedings of the 48th ACM Symposium on Theory of Computing*, pages 36–48. ACM, 2016. 276

[Eli57] Peter Elias. List decoding for noisy channels. Technical Report 335, Research Laboratory of Electronics, MIT, 1957. 3

[Fil16] Yuval Filmus. An orthogonal basis for functions over a slice of the boolean hypercube. *Electr. J. Comb.*, 23(1):P1.23, 2016. 61

[FK96] A. Frieze and R. Kannan. The regularity lemma and approximation schemes for dense problems. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996. 8, 13, 273, 275, 288

[FK98] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998. 6

[Fri91] Joel Friedman. On the second eigenvalue and random walks in randomd-regular graphs. 11, 1991. 1

[Gal62] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962. 3, 190

[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. 91, 186, 270, 275

[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. 97, 188

[GI04] Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting Gilbert-Varshamov bound for low rates. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pages 756–757, 2004. 188, 189, 190

[GI05] V. Guruswami and P. Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005. 3, 275, 324, 325, 327, 328, 329

[Gil52] E.N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952. 4, 7, 184, 190, 268

[GKO$^+$17] Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the Gilbert-Varshamov bound. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2073–2091, 2017. 189, 190

[GM12] Bernd Gärtner and Jiri Matousek. *Approximation Algorithms and Semidefinite Programming*. Applications of Mathematics. Springer-Verlag Berlin Heidelberg, 2012. 126

[GM15] Christopher Godsil and Karen Meagher. *Erdős-Ko-Rado Theorems: Algebraic Approaches*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2015. 40, 61

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95-50, Electronic Colloquium on Computational Complexity, 1995. 100

[GR06] Venkatesan Guruswami and Atri Rudra. Explicit capacity-achieving list-decodable codes. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, pages 1–10, 2006. 184, 189, 190, 270

[GR08] Venkatesan Guruswami and Atri Rudra. Concatenated codes can achieve list-decoding capacity. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 258–267, 2008. 190

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001. 2, 108, 131, 194

[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. 134, 184, 328

[GRY19] Venkatesan Guruswami, Andrii Riazanov, and Min Ye. Arikan meets Shannon: Polar codes with near-optimal convergence to channel capacity. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:154, 2019. 190

[GS98] O. Goldreich and M. Sudan. Computational indistinguishibility: k samples versus $k+1$ samples. In *Proceedings of the 13th IEEE Conference on Computational Complexity*, 1998. 3

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *FOCS*, pages 482–491, 2011. 1, 2, 6, 9, 14, 196, 275

[GS12] Venkatesan Guruswami and Ali Kemal Sinop. Faster SDP hierarchy solvers for local rounding algorithms. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, pages 197–206. IEEE, 2012. 12

[Gur01] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. PhD thesis, MIT, 2001. 91

[Gur04] Venkatesan Guruswami. Guest column: Error-correcting codes and expander graphs. *SIGACT News*, 35(3):25–41, September 2004. 3, 275

[Gur05] Venkatesan Guruswami. Algebraic-geometric generalizations of the Parvaresh-Vardy codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(132), 2005. 190

[Gur09] Venkatesan Guruswami. List decoding of binary codes–a brief survey of some recent results. In *Coding and Cryptology*, pages 97–106. Springer Berlin Heidelberg, 2009. 8, 184, 186, 189, 190, 268

[Gur10] Venkatesan Guruswami. Bridging Shannon and Hamming: List error-correction with optimal rate. In *ICM*, 2010. 8, 184, 190, 268

[Ham50] Richard Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29:147–160, 1950. 7, 190

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 1–10, 1997. 187

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 194

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, August 2006. 1, 91, 292

[HRW17] B. Hemenway, N. Ron-Zewi, and M. Wootters. Local list recovery of high-rate tensor codes applications. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 204–215, Oct 2017. 189, 190

[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. 91, 186, 270

[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. 91, 186, 270

[JQST20] Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit ε-balanced codes near the Gilbert–Varshamov bound. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science*, 2020. 5, 268, 269, 272, 274, 275, 276, 284, 323, 324, 329, 365

[JST21] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of ta-shma's codes via splittable regularity. 2021. 5

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 767–775, 2002. 9

[KKK19] Sushrut Karmalkar, Adam Klivans, and Pravesh Kothari. List-decodable linear regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 4, 96, 97, 125

[KKL14] Tali Kaufman, David Kazhdan, and Alexander Lubotzky. Ramanujan complexes and bounded degree topological expanders. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*, 2014. 1

[KKL16] Tali Kaufman, David Kazhdan, and Alexander Lubotzky. Isoperimetric inequalities for ramanujan complexes and topological expanders. *Geometric and Functional Analysis*, 26(1):250–287, Feb 2016. 11

[KM17] Tali Kaufman and David Mass. High dimensional random walks and colorful expansion. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 4:1–4:27, 2017. 11, 13

[KM18] Tali Kaufman and David Mass. Good distance lattices from high dimensional expanders. *CoRR*, abs/1803.02849, 2018. URL: http://arxiv.org/abs/1803.02849, arXiv:1803.02849. 10

[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. 2, 9, 108, 131

[KO18a] Tali Kaufman and Izhar Oppenheim. Construction of new local spectral high dimensional expanders. In *Proceedings of the 50th ACM Symposium on Theory of Computing*, STOC 2018, pages 773–786. ACM, 2018. 11

[KO18b] Tali Kaufman and Izhar Oppenheim. High order random walks: Beyond spectral gap. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, pages 47:1–47:17, 2018. 11, 13, 14, 16

[KR02] Y. Kohayakawa and V. Rödl. Szemerédi's regularity lemma and quasi-randomness. In *Recent advances in algorithms and combinatorics*. Springer, Berlin, 2002. URL: citeseer.ist.psu.edu/kohayakawa02szemeredis.html. 275

[KV09] Ravindran Kannan and Santosh Vempala. *Spectral algorithms*. Now Publishers Inc, 2009. 275

[LLP17] Eyal Lubetzky, Alex Lubotzky, and Ori Parzanchevski. Random walks on Ramanujan complexes and digraphs. *arXiv e-prints*, page arXiv:1702.05452, Feb 2017. arXiv:1702.05452. 13

[LM06] Nathan Linial and Roy Meshulam. Homological connectivity of random 2-complexes. 26, 2006. 1

[LP20] Yin Tat Lee and Swati Padmanabhan. An $\widetilde{O}(m/\varepsilon^{3.5})$-cost algorithm for semidefinite programs with diagonal constraints. In *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125, pages 3069–3119, 2020. 279, 289, 305, 309, 310, 313, 314, 316

[LPS88] Alexander Lubotzky, R. Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988. 95, 167

[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. 11, 95, 112

[LSV05b] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Ramanujan complexes of typeãd. *Israel Journal of Mathematics*, 149(1):267–299, Dec 2005. 11, 95, 112

[Lub18] Alexander Lubotzky. High dimensional expanders. In *ICM*, 2018. 1, 91

[MM11] Konstantin Makarychev and Yury Makarychev. How to play unique games on expanders. In *Approximation and Online Algorithms*, pages 190–200. Springer Berlin Heidelberg, 2011. 9

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In *Advances in Cryptology – ASIACRYPT 2011*, pages 107–124, 2011. 189

[Mos10] Elchanan Mossel. Gaussian bounds for noise correlation of functions. *Geometric and Functional Analysis*, 19(6), 2010. 2, 276

[MR17] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 14

[MRRW77] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977. 4, 184, 268

[MRRZ+19] Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. LDPC codes achieve list decoding capacity, 2019. arXiv:1909.06430. 190

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2017. 355, 360

[NN90] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. In *Proceedings of the 22nd ACM Symposium on Theory of Computing*, pages 213–223, 1990. 184

[NS09] Michael Navon and Alex Samorodnitsky. Linear programming bounds for codes via a covering argument. *Discrete Comput. Geom.*, 41(2):199–207, March 2009. 184

[OGT15] Shayan Oveis Gharan and Luca Trevisan. A new regularity lemma and faster approximation algorithms for low threshold rank graphs. *Theory of Computing*, 11(9):241–256, 2015. URL: http://www.theoryofcomputing.org/articles/v011a009, doi:10.4086/toc.2015.v011a009. 1, 9, 14, 275

[PRT16] Ori Parzanchevski, Ron Rosenthal, and Ran J. Tessler. Isoperimetric inequalities in simplicial complexes. *Combinatorica*, 36(2):195–227, Apr 2016. 11

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Dense subsets of pseudorandom sets. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, 2008. 273, 275

[Rud91] W. Rudin. *Functional Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1991. 289

[RVW00] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, 2000. 5, 187, 199, 201, 205, 324, 367, 370

[RW17] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. In *Proceedings of the 44th International Colloquium on Au-*

tomata, Languages and Programming. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 12, 126

[RWZ20] N. Ron-Zewi, M. Wootters, and G. Zémor. Linear-time erasure list-decoding of expander codes. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 379–383, 2020. 275

[RY20] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms*, 2020. 4, 96, 97, 125

[Sag13] B.E. Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Graduate Texts in Mathematics. Springer New York, 2013. 61

[Sch08] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, 2008. 2

[Sha48] Claude Shannon. A mathematical theory of communications. *Bell System Technical Journal*, 27:379–423, 623–656, 1948. 190

[SKS19] Ronen Shaltiel, Swastik Kopparty, and Jad Silbak. Quasilinear time list-decodable codes for space bounded channels. In *Proceedings of the 60th IEEE Symposium on Foundations of Computer Science*, 2019. 191

[SS96] M. Sipser and D. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. Preliminary version in *Proc. of FOCS'94*. 3, 275

[Sti08] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Publishing Company, Incorporated, 2nd edition, 2008. 189

[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. 91, 100

[Sud97] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997. 3

[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. 91

[Tho83] C. Thommesen. The existence of binary linear concatenated codes with Reed- Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 29(6):850–853, November 1983. 189, 190

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004. arXiv:cs.CC/0409044. 91

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th ACM Symposium on Theory of Computing*, Proceedings of the 49th ACM Symposium on Theory of Computing, pages 238–251, New York, NY, USA, 2017. ACM. 3, 4, 7, 91, 100, 109, 111, 118, 167, 180, 184, 186, 190, 192, 193, 199, 200, 208, 210, 212, 252, 268, 283, 284, 323, 324, 365, 366, 372, 377

[TTV09] L. Trevisan, M. Tulsiani, and S. Vadhan. Boosting, regularity and efficiently simulating every high-entropy distribution. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, 2009. 273, 274, 275, 278

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Now Publishers Inc., 2012. 268

[Var57] R.R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, 1957. 4, 7, 184, 190, 268

[vL99] Jacobus H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1999. 189

[WJ04] Martin J Wainwright and Michael I Jordan. Treewidth-based conditions for exactness of the Sherali-Adams and Lasserre relaxations. Technical report, Technical Report 671, University of California, Berkeley, 2004. 14

[YZ14] Yuichi Yoshida and Yuan Zhou. Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 423–438. ACM, 2014. 14

APPENDIX A

APPENDIX TO CHAPTER 2

A.1 From Local to Global Correlation

We include the key result we use from [BRS11], namely, their Lemma 5.4 (below). While they proved the lemma for regular graphs, we include the details in the proof for general weighted graphs, since even for HDXs regular at the top level, the swap graphs are not necesarily regular. The extension to general graphs is straighforward (and [BRS11] indicated the same) but we include the details for the sake of completeness [1].

Lemma A.1.1 (Lemma 5.4 from [BRS11] (restatement of Lemma 2.7.5)). *Let $G = (V, E, \Pi_2)$ be a weighted graph, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a local PSD ensemble, where we have $\mathrm{Supp}(\mathbf{Y}_i) \leq q$ for every $i \in V$, and $q \geq 0$. If $\varepsilon \geq 0$ is a lower bound on the expected statistical difference between independent and correlated sampling along the edges,i.e.,*

$$\varepsilon \leq \mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2} \left[\left\| \{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\} \right\|_1 \right].$$

Then, conditioning on a random vertex decreases the variances,

$$\mathop{\mathbb{E}}_{i,j\sim\Pi_1} \left[\mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}} \left[\mathrm{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j \right] \right] \right] \leq \mathop{\mathbb{E}}_{i\sim\Pi_1} \left[\mathrm{Var}\left[\mathbf{Y}_i \right] \right] - \frac{\varepsilon^4}{4q^4 \cdot \mathrm{rank}_{\varepsilon^2/(4q^2)}(G)}.$$

The key ingredient in proving Lemma 5.4 is a "local to global" argument generalizing the expander case to low threshold rank graphs. This new argument is proven in two steps with Lemma A.1.2 being the first one.

1. For expander graphs it is possible to obtain an improved bound of $\Omega((\varepsilon/q)^2)$ instead of $\Omega((\varepsilon/q)^4)$ given by Lemma A.1.1, simply by using the definition of the second smallest eigenvalue of the Laplacian. While BRS analyzed $\mathbb{E}_{i,j}[\langle v_i, v_j\rangle^2]$ for low-threshold rank graphs, it is possible to directly analyze the quantity $\mathbb{E}_{i,j}[\langle v_i, v_j\rangle]$ for expanders, leading to the improved bound.

Lemma A.1.2 (Adapted from Lemma 6.1 of [BRS11]). *Let G be an undirected weighted graph. Suppose $v_1, \ldots, v_n \in \mathbb{R}^n$ are such that*

$$\mathop{\mathbb{E}}_{i \sim V(G)} [\langle v_i, v_i \rangle] = 1, \qquad \mathop{\mathbb{E}}_{ij \sim E(G)} \left[\left\langle v_i, v_j \right\rangle \right] \geq 1 - \varepsilon,$$

but

$$\mathop{\mathbb{E}}_{i,j \sim V(G)} \left[\left\langle v_i, v_j \right\rangle^2 \right] \leq \frac{1}{m}.$$

Then for $c > 1$, we have

$$\lambda_{\left(1 - \frac{1}{c}\right)^2 m} \geq 1 - c \cdot \varepsilon.$$

In particular, $\lambda_{m/4} \geq 1 - 2\varepsilon$.

Proof. Let Y be the Gram matrix defined as $\mathsf{Y}_{i,j} = \left\langle v_i, v_j \right\rangle$. Clearly, Y is positive semi-definite. Without loss of generality suppose that the edge weights $\{w(\{i,j\}) \mid ij \in E(G)\}$ form a probability distribution. Set $w(i) = \sum_{j \sim i} w(\{i,j\})$. Let D to be the diagonal matrix such that $\mathsf{D}(i,i) = w(i)$, i.e., the matrix of generalized degrees. Let A be such that $\mathsf{A}_{i,j} = w(\{i,j\})/2$ and $A_G = D^{-1/2} A D^{-1/2}$ be its normalized adjacency matrix.

Suppose $\mathsf{A}_G = \sum_{i=1}^n \lambda_i u_i u_i^\top$ is a spectral decomposition of A_G. Set $\mathsf{Y}' = \mathsf{D}^{1/2} \mathsf{Y} \mathsf{D}^{1/2}$. For convenience, define the matrix X as $\mathsf{X}(i,j) = \left\langle u_i, \mathsf{Y}' u_j \right\rangle$ and set $p(i) = \mathsf{X}(i,i)$. We claim that p is a probability distribution. Since Y' is positive semi-definite, we have that $p(i) \geq 0$. Moreover, $\sum_{i=1}^n p(i) = 1$ as

$$1 = \mathop{\mathbb{E}}_{i \sim V(G)} [\langle v_i, v_i \rangle] = \mathrm{Tr}(\mathsf{Y}') = \mathrm{Tr}(\mathsf{X}) = \sum_{i=1}^n \mathsf{X}(i,i) = \sum_{i=1}^n p(i).$$

Let m' be the largest value in $[n]$ satisfying $\lambda_{m'} \geq 1 - c \cdot \varepsilon$. By Cauchy-Schwarz[2],

$$q = \sum_{i=1}^{m'} p(i) \leq \sqrt{m'} \sqrt{\sum_{i=1}^{m'} p(i)^2} \leq \sqrt{m'} \sqrt{\sum_{i,j} (\mathsf{X}(i,j))^2} \leq \sqrt{\frac{m'}{m}},$$

where the last inequality follows from our assumption that

$$\frac{1}{m} \geq \mathop{\mathbb{E}}_{i,j \sim V(G)} \left[\left\langle v_i, v_j \right\rangle^2 \right] = \langle \mathsf{Y}', \mathsf{Y}' \rangle = \langle \mathsf{X}, \mathsf{X} \rangle = \sum_{i,j} \mathsf{X}(i,j)^2.$$

Then

$$1 - \varepsilon \leq \mathop{\mathbb{E}}_{ij \sim E(G)} \left[\left\langle v_i, v_j \right\rangle \right] = \langle \mathsf{A}, \mathsf{Y} \rangle = \langle \mathsf{A}_G, \mathsf{X} \rangle = \sum_{i=1}^{n} \lambda_i \mathsf{X}(i,i),$$

implies that

$$1 - \varepsilon \leq \sum_{i=1}^{n} \lambda_i \cdot \mathsf{X}(i,i) \leq \sum_{i=1}^{m'} p(i) + (1 - c \cdot \varepsilon) \sum_{i=m'+1}^{n} p(i) = 1 - c \cdot \varepsilon \, (1 - q) \, .$$

Finally, using the bound on q we obtain

$$\left(1 - \frac{1}{c}\right) \sqrt{m} \leq \sqrt{m'},$$

from which the lemma readily follows. ∎

As a corollary it follows that local correlation implies global correlation.

Corollary A.1.3 (Adapted from Lemma 4.1 of [BRS11])**.** *Let G be an undirected weighted graph. Suppose $v_1, \ldots, v_n \in \mathbb{R}^n$ are vectors in the unit ball such that*

$$\mathop{\mathbb{E}}_{ij \sim E(G)} \left[\left\langle v_i, v_j \right\rangle \right] \geq \rho,$$

2. In [BRS11], there was a minor bug in the application this Cauchy-Schwarz, which led to a bound of $(1 - 1/c)$ instead of $(1 - 1/c)^2$ in the lemma, leading to a global correlation bound of $\Omega(\rho)$ instead of $\Omega(\rho^2)$ as indicated in Corollary A.1.3.

then

$$\mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\left\langle v_i, v_j\right\rangle^2\right] \geq \frac{\rho^2}{4\cdot \mathrm{rank}_{\rho/4}(G)}.$$

In particular, we have

$$\mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\left|\left\langle v_i, v_j\right\rangle\right|\right] \geq \frac{\rho^2}{4\cdot \mathrm{rank}_{\rho/4}(G)}.$$

Proof. If all $v_1,\ldots,v_n$ are zero, the result trivially follows so assume that this is not the case. Then $\alpha = \mathbb{E}_{i\sim V(G)}[\langle v_i, v_i\rangle] > 0$. Also, $\alpha \leq 1$ since the vectors lie in the unit ball. Let $v_i' = v_i/\sqrt{\alpha}$. By construction

$$\mathop{\mathbb{E}}_{i\sim V(G)}\left[\left\langle v_i', v_i'\right\rangle\right] = 1, \qquad \mathop{\mathbb{E}}_{ij\sim E(G)}\left[\left\langle v_i', v_j'\right\rangle\right] \geq \frac{\rho}{\alpha}. \tag{A.1}$$

Under these assumptions we want to apply Lemma A.1.2 in the contra-positive, but first we set some parameters. Let $\rho' = \rho/(2\alpha)$, $\varepsilon = 1-\rho'$ and $c = (1-\rho'/2)/(1-\rho')$. Then

$$1 - \frac{1}{c} = \frac{\rho'/2}{1-\rho'/2} \leq \rho',$$

and

$$1 - c\cdot\varepsilon = \frac{\rho'}{2}.$$

Now, considering the contra-positive of the Lemma A.1.2 under the Eq. (A.1) we obtain

$$\mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\left\langle v_i', v_j'\right\rangle^2\right] > \frac{1}{m} \geq \frac{(\rho')^2}{\mathrm{rank}_{\rho'/2}(G)},$$

since $\mathrm{rank}_{\rho'/2}(G) < (\rho')^2 m$ as $\lambda_{(\rho')^2 m} < \rho'/2$. Or equivalently

$$\mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\frac{\left\langle v_i, v_j\right\rangle^2}{\alpha^2}\right] = \mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\left\langle v_i', v_j'\right\rangle^2\right] \geq \frac{\rho^2}{4\alpha^2\cdot \mathrm{rank}_{\rho/(4\alpha)}(G)} \geq \frac{\rho^2}{4\alpha^2\cdot \mathrm{rank}_{\rho/4}(G)},$$

where the last inequality follows form the fact that $\alpha \leq 1$. ∎

To finish the proof of Lemma 5.4, we state the following Fact A.1.4 extracted from [BRS11].

Fact A.1.4 (Adapted from [BRS11]). *Let $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ be a 2-local PSD ensemble where each $\mathbf{Y}_i$ can take at most q values. Suppose*

$$\varepsilon = \mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2}\left[\left\|\{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\}\right\|_1\right].$$

Then there exist vectors $v_1, \ldots, v_n$ in the unit ball such that

$$\mathop{\mathbb{E}}_{ij\sim E(G)}\left[\left\langle v_i, v_j\right\rangle\right] \geq \frac{1}{q^2}\cdot \mathop{\mathbb{E}}_{ij\sim\Pi_2}\left[\left\|\{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\}\right\|_1^2\right] \geq \frac{\varepsilon^2}{q^2}, \tag{A.2}$$

and

$$\mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\mathrm{Var}\left[\mathbf{Y}_i\right] - \mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}}\left[\mathrm{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j\right]\right]\right] \geq \mathop{\mathbb{E}}_{i,j\sim V(G)}\left[\left|\left\langle v_i, v_j\right\rangle\right|\right]. \tag{A.3}$$

Now we are ready to prove the key result from [BRS11] used in our proof.

Lemma A.1.5 (Lemma 5.4 from [BRS11] (restatement of Lemma 2.7.5)). *Let $G = (V, E, \Pi_2)$ be a weighted graph, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a local PSD ensemble, where we have $\mathrm{Supp}(\mathbf{Y}_i) \leq q$ for every $i \in V$, and $q \geq 0$. If $\varepsilon \geq 0$ is a lower bound on the expected statistical difference between independent and correlated sampling along the edges,i.e.,*

$$\varepsilon \leq \mathop{\mathbb{E}}_{\{i,j\}\sim\Pi_2}\left[\left\|\{\mathbf{Y}_{ij}\} - \{\mathbf{Y}_i\}\{\mathbf{Y}_j\}\right\|_1\right].$$

Then, conditioning on a random vertex decreases the variances,

$$\mathop{\mathbb{E}}_{i,j\sim\Pi_1}\left[\mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}}\left[\mathrm{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j\right]\right]\right] \leq \mathop{\mathbb{E}}_{i\sim\Pi_1}\left[\mathrm{Var}\left[\mathbf{Y}_i\right]\right] - \frac{\varepsilon^4}{4q^4\cdot \mathrm{rank}_{\varepsilon^2/(4q^2)}(G)}.$$

Proof. Using Eq. (A.2) there exist vectors $v_1, \ldots, v_n$ such that Fact A.1.4 implies

$$\mathop{\mathbb{E}}_{ij \sim E(G)} \left[\left\langle v_i, v_j \right\rangle \right] \geq \frac{\varepsilon^2}{q^2}.$$

From Corollary A.1.3 we obtain

$$\mathop{\mathbb{E}}_{i,j \sim V(G)} \left[\left| \left\langle v_i, v_j \right\rangle \right| \right] \geq \frac{\varepsilon^4}{4q^4 \cdot \operatorname{rank}_{\varepsilon^2/(4q^2)}(G)}.$$

Finally, using Eq. (A.3) we get

$$\mathop{\mathbb{E}}_{i,j \sim V(G)} \left[\operatorname{Var}\left[\mathbf{Y}_i\right] - \mathop{\mathbb{E}}_{\{\mathbf{Y}_j\}} \left[\operatorname{Var}\left[\mathbf{Y}_i \mid \mathbf{Y}_j \right] \right] \right] \geq \mathop{\mathbb{E}}_{i,j \sim V(G)} \left[\left| \left\langle v_i, v_j \right\rangle \right| \right] \geq \frac{\varepsilon^4}{4q^4 \cdot \operatorname{rank}_{\varepsilon^2/(4q^2)}(G)},$$

as claimed. ∎

A.2 Harmonic Analysis on HDXs

We provide the proofs of known facts used in Section 2.5.2.

Definition A.2.1 (From [DDFH18])**.** *We say that d-sized complex X is proper provided* $\ker(\mathsf{U}_i)$ *is trivial for* $1 \leq i < d$.

We will need the following decomposition.

Claim A.2.2. *Let* $\mathsf{A} \colon V \to W$ *where V and W are finite dimensional inner product spaces. Then*

$$V = \ker \mathsf{A} \oplus \operatorname{im} \mathsf{A}^\dagger.$$

Proof. We show that $\ker \mathsf{A} = \left(\operatorname{im} \mathsf{A}^\dagger\right)^\perp$. Recall that $v \in \left(\operatorname{im} \mathsf{A}^\dagger\right)^\perp$ if and only if $\left\langle \mathsf{A}^\dagger w, v \right\rangle = 0$ for every $w \in W$. This is equivalent to $\langle w, \mathsf{A}v \rangle = 0$ for every $w \in W$, implying $\mathsf{A}v = 0$.

∎

Lemma A.2.3 (From [DDFH18]). *We have*

$$C^k = \sum_{i=0}^{k} C_i^k.$$

Moreover, if X is proper then

$$C^k = \bigoplus_{i=0}^{k} C_i^k,$$

and $\dim C_i^k = |X(i)| - |X(i-1)|$.

Proof. We induct on k. For $k = 0$, $X(0) = \{\emptyset\}$ and $C^0 = C_0^0$. Now suppose $k > 0$. Since D_k and U_{k-1} are adjoints, we have $C^k = \ker \mathsf{D}_k \oplus \operatorname{im} \mathsf{U}_{k-1}$ or equivalently

$$C^k = \ker \mathsf{D}_k \oplus \mathsf{U}_{k-1} C^{k-1}. \tag{A.4}$$

Using the induction hypothesis $C^{k-1} = \sum_{i=0}^{k-1} C_i^{k-1}$. Note that

$$\mathsf{U}_{k-1} C_i^{k-1} = \left\{ \mathsf{U}_{k-1} \mathsf{U}^{k-1-i} h_i \mid h_i \in H_i \right\} = C_i^k.$$

Thus $C^k = C_k^k + \sum_{i=0}^{k-1} C_i^k$. Assuming $\ker(\mathsf{U}_i)$ is trivial for $0 \leq i < k$ we obtain

$$\dim C_i^k = \dim H_i = \dim C^i - \dim C^{i-1} = |X(i)| - |X(i-1)|,$$

where the second equality follows from Eq. (A.4). Hence $\dim C^k = \sum_{i=0}^{k} \dim C_i^k$. This implies that each $C_i^k \cap \sum_{j \neq i} C_j^k$ is trivial and so we have a direct sum as claimed. ∎

Corollary A.2.4 (From [DDFH18]). *Let $f \in C^k$. If X is proper, then f can be written uniquely as*

$$f = f_0 + \cdots + f_k,$$

where $f_i \in C_i^k$.

APPENDIX B

APPENDIX TO CHAPTER 3

B.1 Auxiliary Basic Facts of Probability

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

Fact B.1.1 (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 B.1.2 (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 B.1.3 (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.2 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 frame-

work. In fact, SOS abstracts the specific combinatorial properties of the lifting function being able to handle them in a unified way.

Lemma B.2.1. *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 C.1.3 yields

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

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 C.1.3 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. ∎

B.3 Derandomization

We show how to derandomize the list decoding framework (which amounts to derandomize Algorithm 3.6.29) when the lifting function is a parity sampler and it satisfies a bounded degree condition (cf Eq. (C.1)). 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 B.3.1 (Majority Word). *Let $z^* \in \{\pm 1\}^{X(1)}$ where $X(1) = [n]$. Suppose that $y^* = \mathrm{dsum}_{X(k)}(z^*)$ satisfy*

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

and

$$\mathbb{P}_{\mathfrak{s} \sim \Pi_k} [\mathfrak{s} \ni i] \leq \frac{g(\varepsilon)}{n}. \tag{B.1}$$

If also $\mathrm{dsum}_{X(k)}$ 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 $z'_i = \mathrm{argmax}_{b \in \{\pm 1\}} \mathbb{P}_{\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}[\mathbf{Z}_i = b]$.

Proof. Define $f(z) := \left| \mathbb{E}_{\mathfrak{s} \sim \Pi_k} y^*_{\mathfrak{s}} \cdot \mathrm{dsum}(z)_{\mathfrak{s}} \right|$. Then, using Eq. (C.1) 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 $\{\pm 1\}^n$ 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 dsum is a $(1 - \xi, 2\varepsilon)$-parity sampler, we obtain

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

By Jensen's inequality,

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

Using indepdence, we get

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

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

which implies

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

Since

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

Markov's inequality yields

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

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 B.3.2. *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 $\xi > 0$.*

APPENDIX C

APPENDIX TO CHAPTER 4

C.1 Auxiliary Basic Facts of Probability

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

Fact C.1.1 (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 C.1.2 (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 C.1.3 (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* $\beta > 0$,

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

C.2 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 frame-

work. In fact, SOS abstracts the specific combinatorial properties of the lifting function being able to handle them in a unified way.

Lemma C.2.1. *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 C.1.3 yields

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

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 C.1.3 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.3 Derandomization

We show how to derandomize the list decoding framework (which amounts to derandomize Algorithm 3.6.29) when the lifting function is a parity sampler and it satisfies a bounded degree condition (cf Eq. (C.1)). 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.3.1 (Majority Word)**.** *Let $z^* \in \{\pm 1\}^{X(1)}$ where $X(1) = [n]$. Suppose that $y^* = \mathrm{dsum}_{X(k)}(z^*)$ satisfy*

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

and

$$\mathbb{P}_{\mathfrak{s} \sim \Pi_k} [\mathfrak{s} \ni i] \leq \frac{g(\varepsilon)}{n}. \tag{C.1}$$

If also $\mathrm{dsum}_{X(k)}$ 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 $z'_i = \mathrm{argmax}_{b \in \{\pm 1\}} \mathbb{P}_{\{\mathbf{Z}^{\otimes}|_{(S,\sigma)}\}}[\mathbf{Z}_i = b]$.

Proof. Define $f(z) := \left| \mathbb{E}_{\mathfrak{s} \sim \Pi_k} y^*_{\mathfrak{s}} \cdot \mathrm{dsum}(z)_{\mathfrak{s}} \right|$. Then, using Eq. (C.1) 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 $\{\pm 1\}^n$ 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 dsum is a $(1 - \xi, 2\varepsilon)$-parity sampler, we obtain

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

By Jensen's inequality,

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

Using indepdence, we get

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

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

which implies

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

Since

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

Markov's inequality yields

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

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.3.2. *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 $\xi > 0$.*

APPENDIX D

APPENDIX TO CHAPTER 5

D.1 Properties of Ta-Shma's Construction

The goal of this section is to provide a reasonably self-contained compilation of the properties of the slightly modified version of Ta-Shma code construction [TS17] from [JQST20]. The properties we need are collected in Theorem D.1.1.

Theorem D.1.1. *[Ta-Shma's Codes (implicit in [TS17])] Let $c > 0$ be an universal constant. For every $\varepsilon > 0$ sufficiently small, there exists $k = k(\varepsilon)$ satisfying $\Omega(\log(1/\varepsilon)^{1/3}) \leq k \leq O(\log(1/\varepsilon))$, $\varepsilon_0 = \varepsilon_0(\varepsilon) > 0$, and positive integer $m = m(\varepsilon) \leq (1/\varepsilon)^{o(1)}$ such that Ta-Shma's construction yields a collection of τ-splittable tuples $W = W(k) \subseteq [n]^k$ satisfying:*

(i) For every linear ε_0-balanced code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ with symbol multiplicity m, the direct sum code $\mathrm{dsum}_W(\mathcal{C}_0)$ is:

(i.1) ε-balanced (parity sampling).

(i.2) if $\mathcal{C}_0$ has rate $\Omega(\varepsilon_0^c/m)$, then $\mathrm{dsum}_W(\mathcal{C}_0)$ has rate $\Omega(\varepsilon^{2+o(1)})$ (near optimal rate)

(ii) $\tau \leq \exp(-\Theta(\log(1/\varepsilon)^{1/6}))$ (splittability).

(iii) W is constructible in $\mathrm{poly}(|W|)$ time (explicit construction).

We first recall the s-wide replacement product in Appendix D.1.1, then describe Ta-Shma's original construction based on it in Appendix D.1.2, describe our modification to obtain splittability in Appendix D.1.3, derive the splittability property in Appendix D.1.4, and finally choose parameters in terms of desired bias ε of the code we construct in Appendix D.1.5. We refer the reader to [TS17] for formal details beyond those we actually need here.

D.1.1 The s-wide Replacement Product

Ta-Shma's code construction is based on the so-called s-wide replacement product [TS17]. This is a derandomization of random walks on a graph G that will be defined via a product operation of G with another graph H (see Definition D.1.3 for a formal definition). We will refer to G as the *outer* graph and H as the *inner* graph in this construction.

Let G be a d_1-regular graph on vertex set $[n]$ and H be a d_2-regular graph on vertex set $[d_1]^s$, where s is any positive integer. Suppose the neighbors of each vertex of G are labeled $1, 2, \ldots, d_1$. For $v \in V(G)$, let $v_G[j]$ be the j-th neighbor of v. The s-wide replacement product is defined by replacing each vertex of G with a copy of H, called a "cloud". While the edges within each cloud are determined by H, the edges between clouds are based on the edges of G, which we will define via operators $\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_{s-1}$. The i-th operator G_i specifies one inter-cloud edge for each vertex $(v, (a_0, \ldots, a_{s-1})) \in V(G) \times V(H)$, which goes to the cloud whose G component is $v_G[a_i]$, the neighbor of v in G indexed by the i-th coordinate of the H component. (We will resolve the question of what happens to the H component after taking such a step momentarily.)

Walks on the s-wide replacement product consist of steps with two different parts: an intra-cloud part followed by an inter-cloud part. All of the intra-cloud substeps simply move to a random neighbor in the current cloud, which corresponds to applying the operator $\mathsf{I} \otimes \mathsf{A}_H$, where A_H is the normalized adjacency matrix of H. The inter-cloud substeps are all deterministic, with the first moving according to G_0, the second according to G_1, and so on, returning to G_0 for step number $s+1$. The operator for such a walk taking $k-1$ steps on the s-wide replacement product is

$$\prod_{i=0}^{k-2} \mathsf{G}_{i \bmod s}(\mathsf{I} \otimes \mathsf{A}_H).$$

Observe that a walk on the s-wide replacement product yields a walk on the outer

graph G by recording the G component after each step of the walk. The number of $(k-1)$-step walks on the s-wide replacement product is

$$|V(G)| \cdot |V(H)| \cdot d_2^{k-1} = n \cdot d_1^s \cdot d_2^{k-1},$$

since a walk is completely determined by its intra-cloud steps. If d_2 is much smaller than d_1 and k is large compared to s, this is less than nd_1^{k-1}, the number of $(k-1)$-step walks on G itself. Thus the s-wide replacement product will be used to simulate random walks on G while requiring a reduced amount of randomness (of course this simulation is only possible under special conditions, namely, when we are uniformly distributed on each cloud).

To formally define the s-wide replacement product, we must consider the labeling of neighbors in G more carefully.

Definition D.1.2 (Rotation Map)**.** *Suppose G is a d_1-regular graph on $[n]$. For each $v \in [n]$ and $j \in [d_1]$, let $v_G[j]$ be the j-th neighbor of v in G. Based on the indexing of the neighbors of each vertex, we define the rotation map* [1] $\mathrm{rot}_G \colon [n] \times [d_1] \to [n] \times [d_1]$ *such that for every $(v, j) \in [n] \times [d_1]$,*

$$\mathrm{rot}_G((v, j)) = (v', j') \Leftrightarrow v_G[j] = v' \text{ and } v'_G[j'] = v.$$

Furthermore, if there exists a bijection $\varphi \colon [d_1] \to [d_1]$ such that for every $(v, j) \in [n] \times [d_1]$,

$$\mathrm{rot}_G((v, j)) = (v_G[j], \varphi(j)),$$

then we call rot_G locally invertible*.*

If G has a locally invertible rotation map, the cloud label after applying the rotation

1. This kind of map is denoted rotation map in the zig-zag terminology [RVW00].

map only depends on the current cloud label, not the vertex of G. In the s-wide replacement product, this corresponds to the H component of the rotation map only depending on a vertex's H component, not its G component. We define the s-wide replacement product as described before, with the inter-cloud operator G_i using the i-th coordinate of the H component, which is a value in $[d_1]$, to determine the inter-cloud step.

Definition D.1.3 (s-wide replacement product)**.** *Suppose we are given the following:*

- *A d_1-regular graph $G = ([n'], E)$ together with a locally invertible rotation map* $\mathrm{rot}_G \colon [n'] \times [d_1] \to [n'] \times [d_1]$.
- *A d_2-regular graph $H = ([d_1]^s, E')$.*

And we define:

- *For $i \in \{0, 1, \ldots, s-1\}$, we define* $\mathrm{Rot}_i \colon [n'] \times [d_1]^s \to [n'] \times [d_1]^s$ *as, for every $v \in [n']$ and $(a_0, \ldots, a_{s-1}) \in [d_1]^s$,*

$$\mathrm{Rot}_i((v, (a_0, \ldots, a_{s-1}))) \coloneqq (v', (a_0, \ldots, a_{i-1}, a_i', a_{i+1}, \ldots, a_{s-1})),$$

  where $(v', a_i') = \mathrm{rot}_G(v, a_i)$.

- *Denote by G_i the operator realizing* Rot_i *and let A_H be the normalized random walk operator of H. Note that G_i is a permutation operator corresponding to a product of transpositions.*

Then $k-1$ steps of the s-wide replacement product are given by the operator

$$\prod_{i=0}^{k-2} \mathsf{G}_{i \bmod s}(\mathsf{I} \otimes \mathsf{A}_H).$$

Ta-Shma instantiates the s-wide replacement product with an outer graph G that is a Cayley graph, for which locally invertible rotation maps exist generically.

Remark D.1.4. *Let R be a group and $A \subseteq R$ where the set A is closed under inversion. For every Cayley graph $\mathrm{Cay}(R, A)$, the map $\varphi\colon A \to A$ defined as $\varphi(g) = g^{-1}$ gives rise to the locally invertible rotation map*

$$\mathrm{rot}_{\mathrm{Cay}(R,A)}((r, a)) = (r \cdot a, a^{-1}),$$

for every $r \in R$, $a \in A$.

D.1.2 The Construction

Let $n' = |V(G)|, m = d_1^s = |V(H)|$ and $n = n' \cdot m = |V(G) \times V(H)|$. Ta-Shma's code construction works by starting with a constant bias code $\mathcal{C}_0'$ in $\mathbb{F}_2^{n'}$, repeating each codeword $m = d_1^s$ times to get a new ε_0-biased code $\mathcal{C}_0$ in $\mathbb{F}_2^n$, and boosting $\mathcal{C}_0$ to arbitrarily small bias using direct sum liftings. Recall that the direct sum lifting is based on a collection $W(k) \subseteq [n]^k$, which Ta-Shma obtains using $k-1$ steps of random walk on the s-wide replacement product of two regular expander graphs G and H. The graph G is on n' vertices and other parameters like degrees d_1 and d_2 of G and H respectively are chosen based on target code parameters.

To elaborate, every $k-1$ length walk on the replacement product gives a sequence of k vertices in the replacement product graph, which can be seen as an element of $[n]^k$. This gives the collection $W(k)$ with $|W(k)| = n' \cdot d_1^s \cdot d_2^{k-1}$ which means the rate of lifted code is smaller than the rate of $\mathcal{C}_0'$ by a factor of $d_1^s d_2^{k-1}$. However, the collection $W(k)$ is a parity sampler and this means that the bias decreases (or the distance increases) from that of $\mathcal{C}_0$. The relationship between this decrease in bias and decrease in rate with some careful parameter choices allows Ta-Shma to obtain nearly optimal ε-balanced codes.

D.1.3 Tweaking the Construction

Recall the first s steps in Ta-Shma's construction are given by the operator

$$\mathsf{G}_{s-1}(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{s-2} \cdots G_1(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_0(\mathsf{I} \otimes \mathsf{A}_H).$$

Naively decomposing the above operator into the product of operators $\prod_{i=0}^{s-1} \mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ is not good enough to obtain the *splittability* property which would hold provided $\sigma_2(\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H))$ was small for every i in $\{0, \ldots, s-1\}$. However, each $\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ has $|V(G)|$ singular values equal to 1 since G_i is an orthogonal operator and $(\mathsf{I} \otimes \mathsf{A}_H)$ has $|V(G)|$ singular values equal to 1. To avoid this issue we will tweak the construction to be the following product

$$\prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H).$$

The operator $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ is exactly the walk operator of the zig-zag product $G \textcircled{z} H$ of G and H with a rotation map given by the (rotation map) operator G_i. This tweaked construction is slightly simpler in the sense that $G \textcircled{z} H$ is an undirected graph. We know by the zig-zag analysis that $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H)$ is expanding as long G and H are themselves expanders. More precisely, we have a bound that follows from [RVW00].

Fact D.1.5. *Let G be an outer graph and H be an inner graph used in the s-wide replacement product. For any integer $0 \leq i \leq s-1$,*

$$\sigma_2((I \otimes \mathsf{A}_H)G_i(I \otimes \mathsf{A}_H)) \leq \sigma_2(G) + 2 \cdot \sigma_2(H) + \sigma_2(H)^2.$$

This bound will imply *splittability* as shown in Appendix D.1.4. We will need to argue that this modification still preserves the correctness of the parity sampling and that it can be achieved with similar parameter trade-offs.

The formal definition of a length-t walk on this slightly modified construction is given below.

Definition D.1.6. *Let $k \in \mathbb{N}$, G be a d_1-regular graph and H be a d_2-regular graph on d_1^s vertices. Given a starting vertex $(v,u) \in V(G) \times V(H)$, a $(k-1)$-step walk on the tweaked s-wide replacement product of G and H is a tuple $((v_1,u_1),\ldots,(v_k,u_k)) \in (V(G) \times V(H))^k$ such that*

- *$(v_1,u_1) = (v,u)$, and*
- *for every $1 \leq i < k$, we have (v_i,u_i) adjacent to (v_{i+1},u_{i+1}) in $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{(i-1) \bmod s}(\mathsf{I} \otimes \mathsf{A}_H)$.*

Note that each $(\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{(i-1) \bmod s}(\mathsf{I} \otimes \mathsf{A}_H)$ is a walk operator of a d_2^2-regular graph. Therefore, the starting vertex (v,u) together with a degree sequence $(m_1,\ldots,m_k) \in [d_2^2]^{k-1}$ uniquely defines a $(k-1)$-step walk.

Parity Sampling

We argue informally why parity sampling still holds with similar parameter trade-offs. In particular, we formalize a key result underlying parity sampling and, in Appendix D.1.5, we compute the new trade-off between bias and rate in some regimes. In Appendix D.1.1, the definition of the original s-wide replacement product as a purely graph theoretic operation was given. Now, we explain how Ta-Shma used this construction for parity sampling obtaining codes near the GV bound.

For a word $z \in \mathbb{F}_2^{V(G)}$ in the base code, let P_z be the diagonal matrix, whose rows and columns are indexed by $V(G) \times V(H)$, with $(\mathsf{P}_z)_{(v,u),(v,u)} = (-1)^{z_v}$. Note that P_z commutes with $I \otimes \mathsf{A}_H$.

Proving parity sampling requires analyzing the operator norm of the following product

$$\mathsf{P}_z \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H), \tag{D.1}$$

when $\mathrm{bias}(z) \leq \varepsilon_0$. Let $\mathbf{1} \in \mathbb{R}^{V(G)\times V(H)}$ be the all-ones vector, scaled to be of unit length under the ℓ_2 norm, and W be the collection of all $(t-1)$-step walks on the tweaked s-wide replacement product. Ta-Shma showed (and it is not difficult to verify) that

$$\mathrm{bias}\left(\mathrm{dsum}_W(z)\right) = \left|\left\langle \mathbf{1}, \mathsf{P}_z \prod_{i=0}^{k-2} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_{i \bmod s}\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H)\mathbf{1} \right\rangle\right|.$$

The measure used in this inner product is the usual counting measure over $\mathbb{R}^{V(G)\times V(H)}$. From the previous equation, one readily deduces that

$$\mathrm{bias}\left(\mathrm{dsum}_W(z)\right) \leq \sigma_1\left(\mathsf{P}_z \prod_{i=0}^{s-1} (\mathsf{I} \otimes \mathsf{A}_H)\mathsf{G}_i\mathsf{P}_z(\mathsf{I} \otimes \mathsf{A}_H)\right)^{\lfloor (k-1)/s \rfloor}.$$

The key technical result obtained by Ta-Shma is the following, which is used to analyze the bias reduction as a function of the total number walk steps $k-1$. Here θ is a parameter used in obtaining explicit Ramanujan graphs.

Fact D.1.7 (Theorem 24 abridged [TS17])**.** *If H is a Cayley graph on $\mathbb{F}_2^{s \log d_1}$ and $\varepsilon_0 + 2\cdot\theta + 2\cdot\sigma_2(G) \leq \sigma_2(H)^2$, then*

$$\left\| \prod_{i=0}^{s-1} \mathsf{P}_z\mathsf{G}_i(\mathsf{I} \otimes \mathsf{A}_H) \right\|_{\mathrm{op}} \leq \sigma_2(H)^s + s\cdot\sigma_2(H)^{s-1} + s^2\cdot\sigma_2(H)^{s-3},$$

where $\mathsf{P}_z \in \mathbb{R}^{(V(G)\times V(H))\times(V(G)\times V(H))}$ is the sign operator *of a ε_0 biased word $z \in \mathbb{F}_2^{V(G)}$ defined as a diagonal matrix with $(P_z)_{(v,u),(v,u)} = (-1)^{z_v}$ for every $(v,u) \in V(G)\times V(H)$.*

We reduce the analysis of Ta-Shma's tweaked construction to an analog of Fact D.1.7.

In doing so, we only lose one extra step as shown below.

Corollary D.1.8. *If H^2 is a Cayley graph on $\mathbb{F}_2^{s\log d_1}$ and $\varepsilon_0 + 2\cdot\theta + 2\cdot\sigma_2(G) \le \sigma_2(H)^4$, then*

$$\left\|\prod_{i=0}^{s-1}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{P}_z\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)\right\|_{\mathrm{op}} \le \sigma_2(H^2)^{s-1} + (s-1)\cdot\sigma_2(H^2)^{s-2} + (s-1)^2\cdot\sigma_2(H^2)^{s-4},$$

where P_z is the sign operator *of an ε_0-biased word $z \in \mathbb{F}_2^{V(G)}$ as in Fact D.1.7.*

Proof. We have

$$\begin{aligned}\left\|\prod_{i=0}^{s-1}(\mathsf{I}\otimes\mathsf{A}_H)\mathsf{P}_z\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H)\right\|_{\mathrm{op}} &\le \|(\mathsf{I}\otimes\mathsf{A}_H)\|_{\mathrm{op}}\left\|\prod_{i=1}^{s-1}\mathsf{P}_z\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H^2)\right\|_{\mathrm{op}}\|\mathsf{P}_z\mathsf{G}_0(\mathsf{I}\otimes\mathsf{A}_H)\|_{\mathrm{op}}\\ &\le \left\|\prod_{i=1}^{s-1}\mathsf{P}_z\mathsf{G}_i(\mathsf{I}\otimes\mathsf{A}_H^2)\right\|_{\mathrm{op}}\\ &\le \sigma_2(H^2)^{s-1} + (s-1)\cdot\sigma_2(H^2)^{s-2} + (s-1)^2\cdot\sigma_2(H^2)^{s-4},\end{aligned}$$

where the last inequality follows from Fact D.1.7. ∎

Remark D.1.9. *We know that in the modified construction H^2 is a Cayley graph since H is a Cayley graph.*

D.1.4 Splittability

In this subsection, we focus on the splittability parameters arising out of the construction described above. The collection $W(k) \subseteq [n]^k$ is obtained from taking $k-1$ step walks on s-wide replacement as described above, which is d_2^2-regular. Recall from Definition 5.3.9 that we need to show $\sigma_2(\mathsf{S}_{W[a,t],W[t+1,b]}) \le \tau$ for all $1 \le a < t < b \le k$, where,

$$\left(\mathsf{S}_{W[a,t],W[t+1,b]}\right)_{(i_a,\cdots,i_t),(i_{t+1},\cdots,i_b)} := \frac{\mathbf{1}[(i_a,\cdots,i_t,i_{t+1},\cdots,i_b) \in W[a,b]]}{d_2^{2(b-s)}}$$

Lemma D.1.10. *Let $1 \leq a < t < b \leq k$. Suppose G is a d_1-regular outer graph on vertex set $[n]$ with walk operator G_t used at step s of a walk on the s-wide replacement product and H is a d_2-regular inner graph on vertex set $[m]$ with normalized random walk operator A_H. Then there are orderings of the rows and columns of the representations of $\mathsf{S}_{W[a,t],W[t+1,b]}$ and A_H as matrices such that*

$$\mathsf{S}_{W[a,t],W[t+1,b]} = ((I \otimes \mathsf{A}_H)G_t(I \otimes \mathsf{A}_H)) \otimes \mathsf{J}/d_2^{2(b-t-1)},$$

where $\mathsf{J} \in \mathbb{R}^{[d_2]^{2(t-a)} \times [d_2]^{2(b-t-1)}}$ is the all ones matrix.

Proof. Partition the set of walks $W[a,t]$ into the sets $W_{1,1},\ldots,W_{n',m}$, where $w \in W_{i,j}$ if the last vertex of the walk $i_t = (v_t, u_t)$ satisfies $v_t = i$ and $u_t = j$. Similarly, partition $W[t+1,b]$ into the sets $W'_{1,1},\ldots,W'_{n',m}$, where $(i_{t+1},\cdots,i_b) \in W'_{i,j}$ if the first vertex of the walk $i_{t+1} = (v_{t+1}, u_{t+1})$ satisfies $v_{t+1} = i$ and $u_{t+1} = j$. Note that $\left|W_{i,j}\right| = d_2^{2(t-a)}$ and $\left|W'_{i,j}\right| = d_2^{2(b-t-1)}$ for all $(i,j) \in [n'] \times [m]$, since there are d_2^2 choices for each step of the walk.

Now order the rows of the matrix $\mathsf{S}_{W[a,t],W[t+1,b]}$ so that all of the rows corresponding to walks in $W_{1,1}$ appear first, followed by those for walks in $W_{1,2}$, and so on in lexicographic order of the indices (i,j) of $W_{i,j}$, with an arbitrary order within each set. Do a

similar re-ordering of the columns for the sets $W'_{1,1}, \ldots, W'_{n',m}$. Observe that

$$\begin{aligned}\left(\mathsf{S}_{W[a,t],W[t+1,b]}\right)_{(i_a,\cdots,i_t),(i_{t+1},\cdots,i_b)} &= \frac{\mathbf{1}_{(i_a,\cdots,i_t,i_{t+1},\cdots,i_b)\in W[a,b]}}{d_2^{2(b-t)}} \\ &= \frac{d_2^2 \cdot (\text{weight of transition from } i_t \text{ to } i_{t+1} \text{ in } (I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H))}{d_2^{2(b-t)}},\end{aligned}$$

which only depends on the adjacency of the last vertex of $(i_a, \cdots, i_t)$ and the first vertex of $(i_{t+1}, \cdots, i_b)$. If the vertices $i_t = (v_t, u_t)$ and $i_{t+1} = (v_{t+1}, u_{t+1})$ are adjacent, then

$$\left(\mathsf{S}_{W[a,t],W[t+1,b]}\right)_{(i_a,\cdots,i_t),(i_{t+1},\cdots,i_b)} = ((I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H))_{(v_t,u_t),(v_{t+1},u_{t+1})} / d_2^{2(b-t-1)},$$

for every $(i_a, \cdots, i_t) \in W[a,t]$ and $(i_{t+1}, \cdots, i_b) \in W[t+1,b]$; and otherwise $\left(\mathsf{S}_{W[a,t],W[t+1,b]}\right)_{(i_a,\cdots,i_t),(i_{t+1},\cdots,i_b)} = 0$. Since the walks in the rows and columns are sorted according to their last and first vertices, respectively, the matrix $\mathsf{S}_{W[a,t],W[t+1,b]}$ exactly matches the tensor product $((I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H)) \otimes \mathsf{J}/d_2^{2(b-t-1)}$. ∎

Corollary D.1.11. *Let $1 \le a < t < b \le k$. Suppose G is a d_1-regular outer graph with walk operator G_t used at step t of a walk on the s-wide replacement product and H is a d_2-regular inner graph with normalized random walk operator A_H. Then*

$$\sigma_2(\mathsf{S}_{W[a,t],W[t+1,b]}) = \sigma_2((I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H)).$$

Proof. Using Lemma D.1.10 and the fact that

$$\sigma_2(((I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H)) \otimes \mathsf{J}/d_2^{2(b-t-1)}) = \sigma_2((I\otimes \mathsf{A}_H)G_t(I\otimes \mathsf{A}_H)),$$

the result follows. ∎

Remark D.1.12. *Corollary D.1.11 is what causes the splittability argument to break down for*

Ta-Shma's original construction, as $\sigma_2(\mathsf{G}_t(\mathsf{I} \otimes \mathsf{A}_H)) = 1$.

D.1.5 Parameter Choices

In this section, we choose parameters to finally obtain Theorem D.1.1, for which we must argue about bias, rate and splittability.

A graph is said to be an (n, d, λ)-graph provided it has n vertices, is d-regular, and has second largest singular value of its normalized adjacency matrix at most λ.

Notation D.1.13. *We use the following notation for the graphs G and H used in the s-wide replacement product.*

- *The outer graph G will be an (n'', d_1, λ_1)-graph.*
- *The inner graph H will be a (d_1^s, d_2, λ_2)-graph.*

The parameters $n'', d_1, d_2, \lambda_1, \lambda_2$ and s are yet to be chosen.

We are given the dimension D of the desired code and its bias $\varepsilon \in (0, 1/2)$. We set a parameter $\alpha \leq 1/128$ such that (for convenience) $1/\alpha$ is a power of 2 and

$$\frac{\alpha^5}{4\log_2(1/\alpha)} \geq \frac{1}{\log_2(1/\varepsilon)}. \tag{D.2}$$

By replacing $\log_2(1/\alpha)$ with its upper bound $1/\alpha$, we observe that the value $\alpha = \Theta(1/\log_2(1/\varepsilon)^{1/6})$ satisfies this bound, and so we choose $s = \Theta(\log_2(1/\varepsilon)^{1/6})$.

The inner graph H. The choice of H is same as Ta-Shma's choice. More precisely, we set $s = 1/\alpha$ and $d_2 = s^{4s}$. We obtain a Cayley graph $H = \mathrm{Cay}(\mathbb{F}_2^{4s\log_2(d_2)}, A)$ such that H is an $(n_2 = d_2^{4s}, d_2, \lambda_2)$ graph where $\lambda_2 = b_2/\sqrt{d_2}$ and $b_2 = 4s\log_2(d_2)$. (The set of generators, A, comes from a small bias code derived from a construction of Alon et al. [AGHP92].)

The base code $\mathcal{C}_0$. This is dealt with in detail in Section 5.5. We choose $\varepsilon_0 = 1/d_2^2$ and use Corollary 5.6.2 to obtain a code $\mathcal{C}_0'$ in $\mathbb{F}_2^{n'}$ that is ε_0-biased and has a blocklength $\Omega(D/\varepsilon_0^c)$ for some constant c. Call this blocklength of $\mathcal{C}_0'$ to be n'. Next we replicate the codewords $m = d_1^s$ times to get code $\mathcal{C}_0$ in $\mathbb{F}_2^n$ with the same bias but a rate that is worse by a factor of m. In the proofs below, we only use properties of $\mathcal{C}_0$ that is of multiplicity m, has rate $\Omega(\varepsilon_0^c)/m$ and has bias ε_0, as specified in Theorem D.1.1.

The outer graph G. Set $d_1 = d_2^4$ so that $n_2 = d_1^s$ as required by the s-wide replacement product. We apply Ta-Shma's explicit Ramanujan graph lemma (Lemma 2.10 in [TS17]) with parameters n', d_1 and θ to obtain an (n'', d_1, λ_1) Ramanujan graph G with $\lambda_1 \leq 2\sqrt{2}/\sqrt{d_1}$ and $n'' \in [(1-\theta)n', n']$ or $n'' \in [(1-\theta)2n', 2n']$. Here, θ is an error parameter that we set as $\theta = \lambda_2^4/6$ (this choice of θ differs from Ta-Shma). Because we can construct words with block length $2n'$ (if needed) by duplicating each codeword, we may assume w.l.o.g. that n'' is close to n' and $(n' - n'') \leq \theta n' \leq 2\theta n''$. See [TS17] for a more formal description of this graph.

Note that $\lambda_1 \leq \lambda_2^4/6$ since $\lambda_1 \leq 3/\sqrt{d_1} = 3/d_2^2 = 3 \cdot \lambda_2^4/b_2^4 \leq \lambda_2^4/6$. Hence, $\varepsilon_0 + 2\theta + 2\lambda_1 \leq \lambda_2^4$, as needed to apply Corollary D.1.8.

The walk length. Set the walk length $k-1$ to be the smallest integer such that

$$(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(k-1)} \leq \varepsilon.$$

This will imply using Ta-Shma's analysis that the bias of the final code is at most ε as shown later.

$s = 1/\alpha, \quad s = \Theta(\log(1/\varepsilon)^{1/6})$, so that $\frac{\alpha^3}{4\log_2(1/\alpha)} \geq \frac{1}{\log_2(1/\varepsilon)}$
$H : (n_2, d_2, \lambda_2), \quad n_2 = d_1^s, \quad d_2 = s^{4s}, \quad \lambda_2 = \frac{b_2}{\sqrt{d_2}}, \quad b_2 = 4s \log d_2$
$\mathcal{C}_0' :$ bias $\varepsilon_0 = 1/d_2^2$, blocklength $n' = O(D/\varepsilon_0^c)$
$\mathcal{C}_0 :$ bias $\varepsilon_0 = 1/d_2^2$, multiplicity $m = d_1^s$, blocklength $n = O(mD/\varepsilon_0^c)$
$G : (n'', d_1, \lambda_1), \quad n'' \approx n' = O(D/\varepsilon_0^c), \quad d_1 = d_2^4, \quad \lambda_1 \leq \frac{2\sqrt{2}}{d_1}$
$k :$ smallest integer such that $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(k-1)} \leq \varepsilon$

Proof of Theorem D.1.1. We will prove it in the following claims. We denote by $W(k) \subseteq [n]^k$ the collection of walks on the s-wide replacement product obtained above, and we denote by $\mathcal{C}$ the final code obtained by doing the direct sum operation on $\mathcal{C}_0$ using the collection of tuples $W(k)$. The explicitness of $W(k)$ follows from Ta-Shma's construction since all the objects used in the construction have explicit constructions.

Next, the multiplicity $m = d_1^s = d_2^{4s} = s^{16s^2} = 2^{16s^2 \log s} \leq (2^{s^6})^{o(1)} = (1/\varepsilon)^{o(1)}$.

Claim D.1.14. *We have $k - 1 \geq s/\alpha = s^2$, and that $k - 1 \leq 2s^5$, so that*

$$\Theta(\log(1/\varepsilon)^{1/3}) \leq k \leq \Theta(\log(1/\varepsilon))$$

Proof. Using $d_2 = s^{4s}$ and Eq. (D.2), we have

$$\begin{aligned}\left(\frac{1}{\lambda_2^2}\right)^{(1-5\alpha)(1-\alpha)s/\alpha} &\leq \left(\frac{1}{\lambda_2^2}\right)^{s/\alpha} = \left(\frac{d_2}{b_2^2}\right)^{s/\alpha} \leq (d_2)^{s/\alpha} = s^{4s^2/\alpha} \\ &= 2^{4s^2 \log_2(s)/\alpha} = 2^{4\log_2(1/\alpha)/\alpha^3} \leq 2^{\log_2(1/\varepsilon)} = \frac{1}{\varepsilon}.\end{aligned}$$

Hence, $\varepsilon \geq (\lambda_2^2)^{(1-5\alpha)(1-\alpha)s/\alpha}$ and thus $k - 1$ must be at least s/α.

In the other direction, we show that $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)2s^5} \leq \varepsilon$, which will imply $k \leq \Theta(s^5) \Rightarrow k \leq \Theta(s^6) = \Theta(\log(1/\varepsilon))$.

$$(\lambda_2^2)^{(1-5\alpha)(1-\alpha)2s^5} \le \left(\frac{b_2^2}{d_2}\right)^{s^5} \le \left(\frac{1}{s^{3s}}\right)^{s^5} = 2^{-\Theta(s^6 \log s)} \le 2^{-\Theta(s^6)} = 2^{-\log(1/\varepsilon)} \le \varepsilon$$

■

Remark D.1.15. *By the minimality of k, we have $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)(k-2)} \ge \varepsilon$. Since $1/(k-1) \le \alpha$, we get $(\lambda_2^2)^{(1-5\alpha)(1-\alpha)^2(k-1)} \ge \varepsilon$. This will be useful in rate computation.*

Claim D.1.16. *The code $\mathcal{C}$ is ε-balanced.*

Proof. Using Corollary D.1.8, we have that the final bias

$$b := \left(\sigma_2(H^2)^{s-1} + (s-1)\cdot\sigma_2(H^2)^{s-2} + (s-1)^2\cdot\sigma_2(H^2)^{s-4}\right)^{\lfloor (k-1)/s \rfloor}$$

is bounded by

$$\begin{aligned}
b &\le (3(s-1)^2\sigma_2(H^2)^{s-4})^{((k-1)/s)-1} && \text{(Using } \sigma_2(H^2) \le 1/3s^2\text{)} \\
&\le ((\sigma_2(H^2)^{s-5})^{(k-1-s)/s} \\
&= \sigma_2(H^2)^{(1-5/s)(1-s/(k-1))(k-1)} \\
&\le \sigma_2(H^2)^{(1-5\alpha)(1-\alpha)(k-1)} \\
&= \left(\lambda_2^2\right)^{(1-5\alpha)(1-\alpha)(k-1)} \le \varepsilon,
\end{aligned}$$

where the last inequality follows from $s = 1/\alpha$ and $k-1 \ge s/\alpha$, the latter from Claim D.1.14.

■

Claim D.1.17. *$\mathcal{C}$ has rate $\Omega(\varepsilon^{2+28\cdot\alpha})$.*

Proof. The support size is the number of walks of length k on the s-wide replacement

product of G and H (each step of the walk has d_2^2 options), which is

$$\begin{aligned}|V(G)||V(H)|d_2^{2(k-1)} = n'' \cdot d_1^s \cdot d_2^{2(k-1)} &= n'' \cdot d_2^{2(k-1)+4s} \leq n' \cdot d_2^{2(k-1)+4s} \\ &= \Theta\left(\frac{D}{\varepsilon_0^c} \cdot d_2^{2(k-1)+4s}\right) \\ &= \Theta\left(D \cdot (d_2^2)^{k-1+2s+c}\right) \\ &= O\left(D \cdot (d_2^2)^{(1+3\alpha)(k-1)}\right),\end{aligned}$$

where the penultimate equality follows from the assumption that ε_0 is a constant.

Note that $d_2^\alpha = d_2^{1/s} = s^4 \geq b_2$ since $b_2 = 4s\log_2(d_2) = 16s^2\log_2(s) \leq s^4$. Thus,

$$d_2^{1-2\alpha} = \frac{d_2}{d_2^{2\alpha}} \leq \frac{d_2}{b_2^2} = \frac{1}{\sigma_2(H^2)}.$$

We obtain

$$\begin{aligned}(d_2^2)^{(k-1)} &\leq \left(\frac{1}{\sigma_2(H^2)}\right)^{\frac{2(k-1)}{1-2\alpha}} \\ &\leq \left(\frac{1}{\varepsilon}\right)^{\frac{2}{(1-2\alpha)(1-5\alpha)(1-\alpha)^2}} && \text{(Using Remark D.1.15)} \\ &\leq \left(\frac{1}{\varepsilon}\right)^{2(1+10\alpha)},\end{aligned}$$

which implies a block length of

$$O\left(D \cdot (d_2^2)^{(1+3\alpha)(k-1)}\right) = O\left(D\left(\frac{1}{\varepsilon}\right)^{2(1+10\alpha)(1+3\alpha)}\right) = O\left(D\left(\frac{1}{\varepsilon}\right)^{2(1+14\alpha)}\right).$$

■

Claim D.1.18. *$W(k)$ is τ-splittable for $\tau \leq 2^{-\Theta(\log(1/\varepsilon)^{1/6})}$.*

Proof. As we saw in Corollary Corollary D.1.11, the splittability τ can be upper bounded by $\sigma_2((I \otimes \mathsf{A}_H)G_t(I \otimes \mathsf{A}_H))$, which is at most $\sigma_2(G) + 2 \cdot \sigma_2(H) + \sigma_2(H)^2$ by Fact D.1.5. So, the collection $W(k)$ is τ-splittable for

$$\begin{aligned}\tau \leq \sigma_2(G) + 2 \cdot \sigma_2(H) + \sigma_2(H)^2 \leq 4\lambda_2 &= 4b_2/d_2^{1/2} \\ &= 64s^2 \log s / s^{2s} \\ &= 2^{-\Theta(s \log s)} \\ &\leq 2^{-\Theta(s)} \\ &= 2^{-\Theta(\log(1/\varepsilon)^{1/6})}\end{aligned}$$

■

■