Electronic Colloquium on Computational Complexity, Report No. 136 (2018)

List Decoding with Double Samplers

Irit Dinur* Prahladh Harsha† Tali Kaufman‡ Inbal Livni Navon§
Amnon Ta Shma¶

July 31, 2018

Abstract

We develop the notion of *double samplers*, first introduced by Dinur and Kaufman [*Proc. 58th FOCS*, 2017], which are samplers with additional combinatorial properties, and whose existence we prove using high dimensional expanders.

We show how double samplers give a generic way of amplifying distance in a way that enables *efficient* list-decoding. There are many error correcting code constructions that achieve large distance by starting with a base code C with moderate distance, and then amplifying the distance using a sampler, e.g., the ABNNR code construction [*IEEE Trans. Inform. Theory*, 38(2):509–516, 1992.]. We show that if the sampler is part of a larger *double sampler* then the construction has an *efficient* list-decoding algorithm and the list decoding algorithm is oblivious to the base code C (i.e., it runs the unique decoder for C in a black box way).

Our list-decoding algorithm works as follows: it uses a local voting scheme from which it constructs a unique games constraint graph. The constraint graph is an expander, so we can solve unique games efficiently. These solutions are the output of the list decoder. This is a novel use of a unique games algorithm as a subroutine in a decoding procedure, as opposed to the more common situation in which unique games are used for demonstrating hardness results.

Double samplers and high dimensional expanders are akin to pseudorandom objects in their utility, but they greatly exceed random objects in their combinatorial properties. We believe that these objects hold significant potential for coding theoretic constructions and view this work as demonstrating the power of double samplers in this context.

*Weizmann Institute of Science, ISRAEL. email: irit.dinur@weizmann.ac.il. Work supported in part by ERC-CoG grant no. 772839.
†Tata Institute of Fundamental Research, INDIA. email: prahladh@tifr.res.in.
‡Bar-Ilan University, ISRAEL. email: kaufmant@mit.edu.
§Weizmann Institute of Science, ISRAEL. email: inbal.livni@weizmann.ac.il.
¶Tel-Aviv University, ISRAEL. email: amnon@tau.ac.il.

ISSN 1433-8092

1 Introduction

We develop the notion of a *double sampler*, which is an "enhanced" sampler. A sampler is a bipartite graph $G = (U, V, E)$ such that for every function $f : V \to [0,1]$ the global expectation $\mu = \mathbb{E}_{v \in V}[f(v)]$ is roughly equal the local expectation $\mu_u = \mathbb{E}_{v \sim u}[f(v)]$ for most $u \in U$. More formally a (α, δ)-sampler is a graph such that at most δ-fraction of the vertices u have $|\mu_u - \mu| \geq \alpha$, see [Zuc97] for more details. In this work we study a strong extension of samplers, called **double samplers**.

Towards defining double samplers we observe that every given a sampler $G = (U, V, E)$, every $u \in U$ can be identified with the set of its neighbors $\{v \in V \mid v \sim u\}$. In this way U is a collection of subsets of V. In the other direction, given a ground set V and a collection of subsets $\{S \subset V\}$, the graph G pops out as the inclusion graph with an edge from $v \in V$ to S iff $v \in S$.

A double sampler consists of a triple (V_2, V_1, V_0) where V_0 is the vertex set, V_1 is a collection of m_1-subsets of V_0, and V_2 is a collection of m_2-subsets of V_0, where $m_2 > m_1$. We say that (V_2, V_1, V_0) is a *double sampler* if

- The inclusion graphs on (V_2, V_1), (V_1, V_0) and on (V_2, V_0) are each samplers. (An inclusion graph is a graph where we connect two subsets by an edge if one contains the other; here a single vertex is also considered to be a singleton subset).
- For every $T \in V_2$, let $V_1(T) = \{S \in V_1 \ : \ S \subset T\}$ be the sets in V_1 that are contained in T. Let $G_{|T}$ be the bipartite inclusion graph connecting vertices in T to subsets in $V_1(T)$. We require that for every $T \in V_2$, the graph $G_{|T}$ is a sampler. We call this property the *locality property* of the double sampler. (See Figure 1 for an illustration.)

Our definition of double samplers is stronger than the initial definition in [DK17] that was missing the locality property[1]. Whereas the definition in [DK17] can be obtained e.g. by concatenating two samplers, the revised definition herein is much stronger and carries properties not known to be obtained by any random construction. It is quite remarkable that high dimensional expanders [LSV05, KO18] give rise to double samplers for which $|V_i| = O(|U|)$:

Theorem 1.1. (Informal, see formal version in Theorem 2.12) For every pair $m_2 > m_1 > 1$, there is an explicit construction of a double sampler on n vertices, such that $|V_i| = O(n)$, for infinitely many $n \in \mathbb{N}$.

On random double samplers. To appreciate the remarkableness of double samplers, think of concrete parameters such as $m_1 = 2, m_2 = 3$. A random construction amounts to placing n vertices in V_0, a linear (in n) number of edges in V_1 and a linear number of triples in V_2. In any $G(n,p)$-like model, most edges will be connected to at most one triple, or vice versa, most triples will be connected to at most one edge. In either case the inclusion graph on V_1, V_2 is highly disconnected, let alone that it be a sampler[2].

We elaborate more on the construction of double samplers towards the end of the introduction.

Samplers and distance amplification. Alon, Bruck, Naor, Naor and Roth [ABN+92] showed how to amplify the distance of any code, simply by pushing the symbols along edges of a sampler graph. Let us describe their transformation in notation consistent with the above. We think of the graph as a sampler $G = (V_1, V_0 = [n])$, where V_1 is a collection of m-sets of $[n]$. Given an n-bit string $w \in \{0,1\}^n$, we place w_i on the i-th vertex and then each subset $S \in V_1$ "collects" all of the symbols

[1] The main result in [DK17] was proven directly from high dimensional expanders, and not from double samplers, so this locality property was used implicitly. It is possible that the result of [DK17] can be proven directly from our revised definition of double samplers.

[2] Observe that for the chosen parameters of $m_1 = 2$ and $m_2 = 3$, there are obvious limits on the (α, δ) parameters of the sampler, since each triple is connected to at most 3 edges.

of its elements and gets a short string $w|_S : S \to \{0,1\}$. The resulting codeword is the sequence $Enc_G(w) := (w|_S)_{S \in V_1}$ which can be viewed as a string of length $|V_1|$ over alphabet $\Sigma = \{0,1\}^m$.

If the string w came from an initial code $C \subset \{0,1\}^n$ with minimum distance $\alpha > 0$, then $Enc_G(C) := \{Enc_G(w) \ : \ w \in C\}$ is a new code. Assuming G is a (α, δ)-sampler, the minimum distance of $Enc_G(C)$ is at least $1 - \delta$. Of course the length of words in $Enc_G(C)$ depends on the size of $|V_1|$, so the shorter the better.

This elegant transformation from C to $Enc_G(C)$ is very local and easy to compute in the forward direction (from w to $Enc_G(w)$), and indeed it has been found useful in several coding theory constructions, e.g. [GI02, KMRS17]. In this work we study the inverse question, also known as decoding: given a noisy version of $Enc_G(w)$, find w. Moreover, we wish to recover from as many errors as possible.

Decoding and list decoding A decoding algorithm for $Enc_G(C)$ gets as input a string $(f_S)_{S \in V_1}$, and needs to find a word $w \in C$ such that $w|_S = f_S$ for as many $S \in V_1$ as possible. A natural approach is the "maximum likelihood decoding" algorithm: assign each vertex $i \in [n]$ the most likely symbol, by looking at the "vote" of each of the subsets $S \ni i$,

$$w'_i := \text{majority}_{S : S \ni i}[f_S(i)]$$

and then run the unique decoding algorithm of C on w'. Assuming C is efficiently unique-decodable from ϵ errors, and assuming G is a good sampler, this gives a decoding algorithm for $Enc_G(C)$ that recovers from error rates close to $1/2$.

Going beyond the unique decoding radius, the large distance of $Enc_G(C)$ guarantees, via the Johnson bound, that it is (combinatorially) list decodable up to a radius $1 - \sqrt{\delta}$ where $1 - \delta$ is the distance (see [GRS, Chapter 7]). However, the maximum likelihood decoder stops working in this regime: one cannot rule out the situation where for each vertex i, both 0 and 1 symbols occur with equal likelihood, and it is not known, in general,[3] how to recover w.

Thus, it is natural to ask for an algorithm that list decodes up to radius close to $1 - \sqrt{\delta}$. Our main result is a list decoding algorithm that goes beyond the unique-decoding $1/2$ barrier and works for error rates approaching 1. The algorithm works whenever the underlying graph $G = (V_1, [n])$ is part of a double sampler, namely where there is a collection V_2 of sets of size $m_2 > m_1 = m$ so that the triple $(V_2, V_1, [n])$ is a double sampler.

Theorem 1.2 (Main - informal, see Theorem 3.1)**.** Let $\epsilon_0, \epsilon > 0$ be constants. Suppose $C \subset \{0,1\}^n$ is a code that is efficiently decodable from ϵ_0 errors, and suppose $X = (V_2, V_1, [n])$ is a double sampler (with parameters m_2, m_1 depending on ϵ_0, ϵ). The code $Enc_X(C)$ defined over alphabet $\Sigma = \{0,1\}^{m_1}$ by

$$Enc_X(C) = \{(g|_S)_{S \in V_1} \mid g \in C\} \subset \Sigma^{V_1}$$

has block length $|V_1| = O(n)$, and is list-decodable from $1 - \epsilon$ fraction of errors.

At this point the reader may be wondering how the double-sampler property helps facilitate list decoding. Roughly speaking, a double sampler is a collection of (small) subsets that have both large overlaps as well as strong expansion properties. The expansion properties are key for distance amplification, and the large overlaps, again with good sampling properties, are key for the list decoding algorithm.

1.1 The list decoding algorithm

Our algorithm starts out with a voting step, similar to the maximum likelihood decoder. Here we vote not on the value of each bit $i \in [n]$ but rather on the value of an entire set $T \in V_2$. Since (V_2, V_1)

[3] We remark that when the base code C has additional special properties it is possible that more can be done (see e.g [GI02]), but our focus is on a generic decoding mechanism that does not depend on the code at all. In fact, we show an approximate list decoding algorithm that works even when C does not have any distance, e.g. for $C = \{0,1\}^n$.

is also a sampler, a typical T sees a noticeable fraction of S's for which $f_S = w|_S$. Since the graph $X_{|T}$ between $V_1(T)$ and T is a sampler (this is the locality property), we can come up with a short list of popular candidates for $w|_T$. This is done by looking at f_S for all subsets $S \in V_1(T)$. We define

$$\forall T \in V_2, \quad list(T) := \{\sigma \in \{0,1\}^T \ : \ \Pr_{S \subset T, S \in V_1}[f_S = \sigma|_S] > \epsilon/2 \ \}.$$

Note that since T has constant size, we are able to search exhaustively over all $\sigma \in \{0,1\}^T$ in constant time.

Given a list for each T, we now need to stitch these lists together, and here we again use the fact that (V_2, V_1) is a good sampler. Whenever $T_1 \cap T_2$ is significantly large, we will match $\sigma_1 \in list(T_1)$ with $\sigma_2 \in list(T_2)$ iff $\sigma_1|_{T_1 \cap T_2} = \sigma_2|_{T_1 \cap T_2}$. Moreover, the double sampler property allows us to come up with an *expander* graph whose vertex set is V_2, and whose edges connect T_1 to T_2 when they have significant overlap. This guarantees that for almost all edges (T_1, T_2) there is a matching between the list of T_1 and the list of T_2.

At this point what we are looking at is a **unique games instance**, where the said expander is the constraint graph, and the said matchings are the unique constraints.[4] We now make two important observations. First, a word with noticeable correlation with the received word, corresponds to a solution for the unique games instance with very high value (i.e., satisfying a large fraction of the constraints). Second, the algorithm of [AKK+08] will find a high-value solution, because the underlying unique games constraint graph is an expander! It is important to understand that a more naive greedy belief propagation algorithm would fail miserably because it takes about $\log n$ steps to reach a typical point in an expander graph, and this accumulate an intolerable $\epsilon \cdot \log n \gg 1$ amount of error.

We are almost done, it remains to run the unique decoding algorithm of C on each unique games solution, to remove any small errors, and this completes the list decoding.

The above high level description gives the rough idea for our algorithm, but the implementation brings up some subtle difficulties, which we explain below.

Every set T induces a constant size "local view" $C|_T$ on the code C, which has no reason to be an error correcting code, and in particular has no distance. This makes the task of finding a short $list(T)$ more difficult, since there could be several valid candidates $\sigma \in \{0,1\}^T$ that are very close in Hamming distance. Suppose $\sigma, \sigma' \in list(T_1)$ differ only in a single bit, then for most $T_2 \cap T_1$, we don't know which element in $list(T_2)$ should be matched to σ and which to σ'.

In order to solve this difficulty, we prune each $list(T)$ and enforce minimal distance r between each two list items. The pruning should also promise a "covering" property - that if σ was in the initial list, then exists some $\sigma' \approx \sigma$ (where $\approx$ is with relation to r) in the final list.

For any predetermined distance parameter r, one can come up with counterexamples showing that this is impossible. Our solution is to let the pruning algorithm choose r dynamically. The algorithm starts with an initial distance r, and gradually increases it till it reaches a radius at which both the distance and covering properties holds together (this is done in Section 4).

Given T_1, T_2 with $list(T_1), list(T_2)$ and the same radius r, we match $\sigma_1 \in list(T_1)$ to $\sigma_2 \in list(T_2)$ if they are close (with respect to r) on $T_1 \cap T_2$. If however T_1, T_2 have different radii, we don't know how to match these lists correctly. Therefore, our unique games instance is created on a subgraph containing only those vertices T that share the same (most popular) radius r. We show that there exists such a subgraph which is itself an expander.

1.2 Double Samplers and High dimensional expanders

Let us briefly explain how double samplers are constructed from high dimensional expanders (proving Theorem 1.1). A high dimensional expander is a d-dimensional simplicial complex X, which is just a hypergraph with hyperedges of size $\leq d+1$ and a closure property: for every hyperedge in

[4] For definitions, please see the preliminary section.

the hypergraph, all of its subsets are also hyperedges in the hypergraph. The hyperedges with $i+1$ elements are denoted $X(i)$, and the complex is said to be an expander if certain spectral conditions are obeyed, see Section 6.

In [DK17] the authors prove that a (two-sided spectral) high dimensional expander gives rise to a multi-partite graph with interesting spectral expansion properties. The graph has vertices $X(d) \cup X(d-1) \cup \ldots \cup X(0)$, and we place edges for inclusion. Namely, $S \in X(m_1)$ is connected by an edge to $T \in X(m_2)$ if $S \subset T$. It is shown that the graph induced by focusing on layers i and j has $\lambda(G(X(i), X(j))) \leq \frac{i+1}{j+1} + o(1)$. We show, in Section 6, that by narrowing our focus to three layers in this graph (namely, $X(m_2-1), X(m_1-1), X(0)$) we get a double sampler. This is proven by observing that the spectral properties are strong enough to yield a sampler (an expander mixing lemma argument suffices since we are only seeking relatively weak sampling properties).

Better double samplers? Double samplers with super-linear (polynomial and even exponential) size have appeared implicitly (or somewhat similarly as "intersection codes") in the works of [IKW12, IJKW10]. Two concrete constructions were studied,

- The first where $V_i = \binom{V}{m_i}$, so $|V_i| \approx n^{m_i}$.
- The second where V is identified with a vector space over some finite field and then V_i consists of all d_i-dimensional subspaces of V. Here $|V_i| \approx n^{d_i}$.

The current work is the first to construct double samplers with linear size. This raises the question of finding the best possible parameters for these objects. In particular, for given sampler parameters α and δ, how small can $|V_1|/|V_0|$ be?

Our current construction is based on Ramanujan complexes of [LSV05] that are optimal with respect to the spectrum of certain Laplacian operators, and not necessarily with respect to obtaining best possible double samplers. It is an interesting challenge to meet and possibly improve upon these parameters through other constructions.

Unlike other pseudorandom objects, there is no known random construction of a double sampler. In particular, we cannot use it as a yardstick for the quality of our parameters. It remains to explore what possible parametric limitations there are for these objects.

We believe that double samplers capture a powerful feature of high dimensional expanders whose potential for TCS merits more study. Previously, in [DK17], it was shown that high dimensional expanders give rise to a very efficient de-randomization of the direct product code that is nevertheless still testable. Part of the contribution of the current work is a demonstration of the utility of these objects in a new context, namely of list decoding.

1.3 Derandomized Direct Product and Approximate List Decoding

Our list decoding algorithm can also be viewed in the context of decoding derandomized direct products. The *direct product* encoding takes $g \in \{0,1\}^N$ and encodes it into $Enc(g) = (g|_S)_{S \in \mathcal{S}}$ where $\mathcal{S} = \binom{[N]}{k}$ contains all possible k-subsets of $[N]$. An encoding with $|\mathcal{S}| \ll \binom{N}{k}$, as in this paper, is called a *derandomized* direct product encoding.

Direct products and derandomized direct products are important in several contexts, primarily for **hardness amplification**. One begins with a string $g \in \{0,1\}^N$ that is viewed as a truth table of a function $g : \{0,1\}^n \to \{0,1\}$ (here $N = 2^n$), and analyzes the hardness of the new function defined by $Enc(g)$. A typical hardness amplification argument proceeds by showing that if no algorithm (in a certain complexity class) computes g on more than $1 - \epsilon_0$ of its inputs, then no algorithm computes $Enc(g)$ on more than ϵ of its inputs. Namely, $Enc(g)$ is much harder than g.

Such a statement is proven, as first described in [Tre05, Imp03], through a (list-) decoding argument: given a hypothetical algorithm that computes $Enc(g)$ successfully on at least ϵ fraction of inputs, the approximate list decoder computes g on $(1 - \epsilon_0)$ of its inputs. We prove,

Theorem 1.3 (Approximate list decoding - informal)**.** Let $\epsilon_0, \epsilon > 0$ be constants. Suppose $X = (V_2, V_1, [N])$ is a double sampler (with parameters depending on ϵ_0, ϵ). Let Enc_X be the encoding that takes $g \in \{0,1\}^N$ to $Enc_X(g) = (g|_S)_{S \in V_1}$. There is an algorithm (running in time $poly(N)$) that when given a word $(f_S)_{S \in V_1}$ that is ϵ-correlated with $Enc_X(g)$ for some $g \in \{0,1\}^N$, namely, $\Pr_{S \in V_1}[f_S = g|_s] > \epsilon$, finds $g' \in \{0,1\}^N$ such that $dist(g', g) \leq \epsilon_0$.

This theorem differs from Theorem 1.2 only in that g does not come from any initial error correcting code $C \subset \{0,1\}^N$. This is why we can only provide an approximate answer g' instead of g.

Our list decoding result falls short of being useful for hardness amplification, because it is not local or efficient enough. We leave it as an open question whether a more efficient and local list decoding algorithm exists for these codes.

There is a significant technical hurdle that one faces, related to the diameter of the bipartite graph corresponding to (V_1, V_0). In the local list decoding constructions analyzed in [IKW12, IJKW10] (both derandomized and non-derandomized), the diameter is $O(1)$, and this is crucially used in the list decoding algorithm.

When we move to a linear size derandomized direct product encoding, as we do in this work, we pay by enlarging the diameter to become super-constant. This is what makes the approximate list decoding algorithm performed in our work much more challenging (even in the non-local setting), and the algorithm more complicated than the analogous task performed by [IKW12, IJKW10].

1.4 Future directions

The construction in this paper starts with a binary code and constructs a code over a larger alphabet that has efficient list decoding. The larger alphabet size arises because we use a derandomized direct product, i.e., each vertex $S \in V_1$ collects the bit symbols from all indices $i \in S$. It is natural to consider the *direct sum* operation, where for each $S \in V_1$ we output a single bit that is the sum (over $\mathbf{F}_2$) of all values in the direct sum. A recent example of such a code is the recent construction in [Ta-17]. This code achieves close to optimal rate (and this is made possible because the direct product operator is replaced with a direct sum) and has explicit encoding (the sampler it uses is a variant of the sampler that is obtained through random walks on good expanders). One obvious shortcoming of the code of [Ta-17] is that while it has efficient encoding, it is not known how to efficiently decode (or list decode) it. It is still a major open problem to find a binary code with distance close to half, close to optimal rate (as in [Ta-17]) and efficient encoding *and* decoding.

It is possible that the results in this paper might help with finding such a code:

- First, while the result in this paper uses direct product, it is conceivable that with a refinement of the double sampler notion one might do with direct sum. This is true, e.g., in a situation where for each $T \in V_2$, the values f_S for $S \in V_1$,$S \subset T$ (and notice that now $f_S \in \{0,1\}$) define a code, or even, an approximately list decodable code. This property is different from the locality property we require from our double samplers, but is close in spirit to it.
- Then, having that, one needs to improve the parameters of the algorithms presented in the paper, and also find appropriate "double samplers" (with the stronger property we require). It is not known whether such objects exist, and even for double samplers it is not clear what the best possible parameters are. This question is quite intriguing as we *cannot* compare our desired explicit object to non-explicit random objects, simply because here random objects are no good.

Thus, while this approach seems right now technichally challenging, and it is not even clear parameter-wise (or non-explicitly) whether it is possible, we believe it opens up a new, and exciting, research agenda that we hope will eventually lead to near-optimal binary codes that have both explicit encoding and decoding.

2 Preliminaries and Notations

2.1 Weighted graphs and expanders

Definition 2.1. (Weighted graph) We say that (G, w) is a weighted graph if $G = (V, E)$ is an undirected graph, and $w : E \to \mathbb{R}_{\geq 0}$ is a weight function that associates with each edge e a non-negative weight w_e. We have the convention that non-edges have zero weight. Given the edge weights w_e the weight w_v of a vertex is defined as $w_v := \sum_{e:v\in e} w_e$. The edge weights induce a distribution on edges (and vertices) which we denote by μ_G.

Definition 2.2. (Edge expansion) Let $(G = (V, E), w)$ be a weighted graph. The edge expansion of G is

$$h_G := \min_{V'\subset V, \mu(V')\leq \frac{1}{2}} \frac{\mu(E(V', V \setminus V'))}{\mu(V')},$$

where $E(A, B)$ denotes the set of edges between A and B.

Definition 2.3. For every weighted graph G, let the normalized adjacency matrix A be defined as $A_{u,v} = \frac{w_{u,v}}{\sqrt{w_u w_v}}$. Let $\lambda(G)$ be the second largest eigenvalue (in absolute value) of A.

2.2 Samplers

Definition 2.4. (Weights on k-partite graphs) Let $G = (V_k, \ldots, V_1, E)$ be a k-partite graph and let W be a distribution over V_k. Define a joint distribution Π whose values are k-partite paths $(v_k, \ldots, v_1) \in V_k \times \cdots \times V_1$ chosen by sampling $v_k \in V_k$ according to W, then choosing a random neighbor $v_{k-1} \in V_{k-1}$ of v_k and so forth. We denote by Π_i the i-th coordinate of Π.

We will say that $v_1 \sim (\Pi_1 | \Pi_2 = v_2)$ to mean that v_1 is a random neighbor (in V_1) of v_2.

Definition 2.5. (Sampler) Let $G = (V_2, V_1, E)$ be a bipartite graph with a distribution W on V_2. We say (G, W) is an (α, δ) sampler, if the following holds for every $f : V_1 \to [0, 1]$,

$$\Pr_{v_2\sim\Pi_2}\left[\left|\mathop{\mathbb{E}}_{v_1\sim(\Pi_1|\Pi_2=v_2)}[f(v_1)] - \mathop{\mathbb{E}}_{v_1\sim\Pi_1}[f(v_1)]\right| \geq \alpha\right] \leq \delta.$$

Definition 2.6 (Two-step walk graph)**.** Let $(G = (V_2, V_1, E, W)$ be a bipartite graph with distribution W on V_2. The two-step walk of G is the weighted graph $(G_2 = (V_2, E), \{w_e\})$ whose edge weights are given by selecting $S \sim \Pi_1$ and then two independent copies $T_1, T_2 \sim (\Pi_2 | \Pi_1 = S)$. More explicitly

$$\forall e = (u_1, u_2), \qquad w_e = \sum_{S\in V_1} \Pi_1(S) \Pr_{T_1,T_2\sim(\Pi_2|\Pi_1=S)}[T_1 = u_1 \wedge T_2 = u_2].$$

The following simple fact is important,

Fact 2.7. If we choose a random edge in G_2, and a random vertex T in it, then T is distributed according to Π_2.

Theorem 2.8 (Every sampler contains an induced expander)**.** Let $\alpha, \beta, \delta \in (0, 1)$ be such that $\alpha, \delta < \frac{\beta^2}{100}$. Let $(G_{samp} = (V_2, V_1, E_s), W_S)$ be an (α, δ) sampler. Let $(G = (V_2, E), W)$ be the two-step walk graph of G_{samp}. Let $A \subseteq V_2$ be any set with $\mu_G(A) \geq \beta$. Then there exists a set $B \subseteq A$ such that:

- $\mu_G(B) \geq \frac{\beta}{4}$.
- Let G_B be the induced graph of G on B. $\lambda(G_B) \leq \frac{99}{100}$.

Furthermore, given A, such a set B can be found in time polynomial in $|V|$.

The theorem is proven in Appendix A.

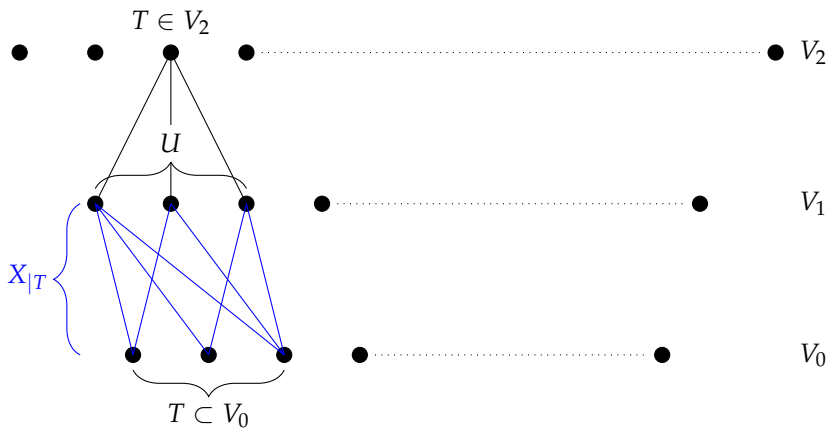

Figure 1: double sampler

2.3 Double Samplers

Definition 2.9. (Inclusion graph) An *inclusion* graph $X = (V_2, V_1, V_0)$ with cardinalities $m_2 > m_1 > 0$ is a tri-partite graph with vertices $V = V_2 \cup V_1 \cup V_0$ where $V_i \subseteq \binom{V_0}{m_i}$ for every $i > 0$ and $(a, b) \in E$ iff $a \subseteq b$.

Given an inclusion graph $X = (V_2, V_1, V_0)$ and distribution W over V_2, recall from Definition 2.4 that W induces a distribution Π over paths (v_2, v_1, v_0) whose components are $\Pi_2 = W, \Pi_1$ and Π_0.

Definition 2.10 (Double Sampler)**.** Let X be an inclusion graph with a distribution W over V_2, let $X(V_i, V_{i+1})$ be the bipartite graph between V_i, V_{i+1}.

We say that (X, W) is a $((\alpha_{2,1}, \delta_{2,1}), (\alpha_{1,0}, \delta_{1,0}), (\alpha_{local}, \delta_{local}))$ double sampler, if

1. $(X(V_2, V_1), \Pi_2)$ is a $(\alpha_{2,1}, \delta_{2,1})$ sampler.
2. $(X(V_1, V_0), \Pi_1)$ is a $(\alpha_{1,0}, \delta_{1,0})$ sampler.
3. For every $T \in V_2$, we define the weighted bipartite graph

$$(X_{|T} = (U, T, E)\ ,\ W_T)$$

where $U = \{S \in V_1 \mid S \subseteq T\}$ and $(S, i) \in E$ for $S \in U$ and $i \in T$ iff $i \in S$, and $W_T = (\Pi_1 | \Pi_2 = T)$. We require that $(X_{|T}, W_T)$ is a $(\alpha_{local}, \delta_{local})$ sampler.

Furthermore, X is called *regular* if Π_0 is uniform over V_0 and for each T, $(\Pi_0 | \Pi_2 = T)$ is uniform over T.

Note that the distribution $(\Pi_1 | \Pi_2 = T)$ on U is by definition the uniform distribution on U.

We remark that items 1 and 3 already imply item 2 in the definition with $\alpha_{1,0} = \alpha_{2,1} + \alpha_{local}, \delta_{1,0} = \delta_{2,1} + \delta_{local}$. In fact, our use of double samplers relies only on items 1 and 3.

Short of being uniform, we formulate a "flatness" property of the distributions Π_1, Π_2 involved in the double sampler,

Definition 2.11. A distribution Π over V is said to be D-flat for an integer $D \in \mathbb{N}$ if there is some R such that $\Pi(v) \in \{\frac{1}{R}, \frac{2}{R}, \ldots, \frac{D}{R}\}$ for each $v \in V$.

This property will allow us to treat the distribution as uniform by duplicating each element at most D times.

2.4 Double Samplers Exist

We prove, in Section 6, that linear size double samplers are implied by the existence of high dimensional expanders.

Theorem 2.12. For every $\alpha_{2,1}, \delta_{2,1}, \alpha_{1,0}, \delta_{1,0}, \alpha_{local}, \delta_{local} > 0$ there exist $D, m_2, m_2 \in \mathbb{N}$ and a family of explicitly constructible double samplers (X_n, W_n) for infinitely many $n \in \mathbb{N}$ such that

- X_n is an inclusion graph (V_2, V_1, V_0) where $|V_0| = n$, $V_i \subseteq \binom{V_0}{m_i}$ for $i = 1, 2$, with distribution W_n over $V_2(X_n)$.
- X_n is a regular $((\alpha_{2,1}, \delta_{2,1}), (\alpha_{1,0}, \delta_{1,0}), (\alpha_{local}, \delta_{local}))$ double sampler.
- $|V_1|, |V_2| \leq D \cdot n$. The distributions Π_1, Π_2 are D-flat.

2.5 UG Constraint Graphs

Definition 2.13. (UG constraint graph) Let $G = (V, E, w)$ be a weighted graph. We say $(G, \{\pi_e\}_{e \in E})$ is a UG constraint graph with ℓ labels if $\pi_e : [\ell] \to [\ell]$ is a permutation. We say a is an assignment for G if $a : V \to [\ell]$. We say the assignment a satisfies an edge $e = (u, v)$ if $\pi_e(a(u)) = a(v)$. The *value* of an assignment is the fraction of satisfied edges. We say $(G, \{\pi_e\})$ is p-satisfiable if there exists an assignment with value at least p.

Arora et. al. [AKK+08] showed how to solve unique games instances on expander graphs in polynomial time. This result was improved by Makarychev and Makarychev [MM10], who proved,

Theorem 2.14 ([MM10, Theorem 10])**.** Let G be a regular graph with second smallest Lapacian eigenvalue λ_G and edge expansion h_G. There exist positive absolute constants c and C and a polynomial time approximation algorithm that given a $1 - \delta$ satisfiable instance of UG on G with $\frac{\delta}{\lambda_G} \leq c$, the algorithm finds a solution of value $1 - C\frac{\delta}{h_G}$.

We need a version of this theorem with two modifications:

- The theorem, as stated, refers to unweighted, regular graphs. We need the same results for non regular weighted graphs.
- The theorem finds one assignment with high value. However, we need to get an approximation to *all* assignments with high value.

In the appendix we go over [AKK+08, MM10] and show that the same result holds for weighted, non-regular graphs (see Appendix B.2). We also show how to output a list that contains an approximation to all assignments with high value. To do that we rerun the algorithm of Theorem B.4 several times, each time peeling off the solution that is found (see Appendix B.1). We prove:

Theorem 2.15. Let $(G = (V, E, w), \{\pi_e\})$ be a weighted undirected UG constraint graph with labels $[\ell]$ such that $\lambda(G) \leq \frac{99}{100}$. There exits an absolute constant c and a polynomial time algorithm that outputs a list L of assignments such that for every assignment $a : V \to [\ell]$ with value $1 - \eta$ there exists an assignment $b \in L$ which satisfies $\mu(\{v : a(v) = b(v)\}) \geq 1 - \eta c^{\ell}$.

2.6 Miscellaneous notation

Definition 2.16. Let $\sigma, \sigma' \in \{0,1\}^n$, and let $S \subseteq n$, then

$$\mathrm{dist}_S(\sigma, \sigma') = \frac{1}{|S|} \left| \{ i \in S \mid \ \sigma_i \neq \sigma'_i \} \right|.$$

In the case of $S = [n]$, we omit the subscript S.

3 The main theorem: encoding, list decoding and correctness proof

In this section we describe how to transform any uniquely decodable code to a list-decodable code, using double samplers. We begin by describing the new list decodable code through its encoding algorithm (Section 3.1). We then describe the list decoding algorithm (Section 3.2), and finally prove the correctness in Section 3.3.

3.1 The encoding

Let C be an $[|V_0|, k]_2$ linear binary code. Let $(X = (V_2, V_1, V_0), W)$ be a regular double sampler as in Theorem 2.12 with $|V_0| = n$, and parameters as follows: $((\alpha_{2,1} = \epsilon_0^2 10^{-5}, \delta_{2,1} = \frac{\epsilon_0}{200} c^{-\frac{8}{\epsilon}}), (\alpha_{local} = \frac{\epsilon_0}{20} 10^{-\frac{8}{\epsilon}}, \delta_{local} = \frac{\epsilon_0}{200} c^{-\frac{8}{\epsilon}}))$ such that $m_2 > \frac{1}{\alpha_{local}}$, for c an absolute constant. We assume that Π_1 is D-flat for a parameter D that depends exponentially on $\mathrm{poly}(1/\alpha + 1/\delta)$ (where α is the smallest of the α's and δ is the smallest of the δ's).

As explained in the introduction, we wish to encode $g \in \{0,1\}^{V_0}$ by $(g|_S)_{S \in V_1}$. The double sampler assigns the set V_1 a probability distribution Π_1. If Π_1 is not uniform, we view V_1 as a multiset that has multiple copies of each S to account for the different probabilities. Choosing S uniformly in the multiset V_1 is identical to choosing $S \sim \Pi_1$. The fact that Π_1 is D-flat guarantees that each S needs to be repeated at most D times. From now on we view V_1 as a multiset and let $|V_1|$ denote the number of elements in V_1 counted with multiplicity.

Denote $\Sigma = \{0,1\}^{m_1}$. We define the encoding

$$ENC : \{0,1\}^{|V_0|} \to \Sigma^{|V_1|}$$

Given $g : V_0 \to \{0,1\}$ we let $Enc(g) : V_1 \to \Sigma$ be defined by

$$\forall S \in V_1, \qquad (\mathrm{Enc}(g))(S) = g_{|S}.$$

We let $ENC \circ C$ be the code $\{Enc(g) \mid g \in C\}$.

The encoding is essentially the same as [ABN+92], with weights on the sampler graph. We remark that the construction also works with any non-linear code C, in which case the resulting code is also not (necessarily) linear.

Theorem 3.1. Let ϵ_0, ϵ be constants, and let X be a $((\alpha_{2,1} = \epsilon_0^2 10^{-5}, \delta_{2,1} = \frac{\epsilon_0}{200} c^{-\frac{8}{\epsilon}}), (\alpha_{local} = \frac{\epsilon_0}{20} 10^{-\frac{8}{\epsilon}}, \delta_{local} = \frac{\epsilon_0}{200} c^{-\frac{8}{\epsilon}}))$ double sampler, with $m_2 > \frac{1}{\alpha_{local}}$, $D \leq \exp(\exp(1/\epsilon))$, $|V_1| = |V_2| \leq D \cdot |V_0|$, and c an absolute constant.

Suppose C is a $[|V_0|, k]_2$ code with an efficient unique-decoding from $\epsilon_0 |V_0|$ errors. Then $ENC \circ C$ is an $[|V_1|, k]_\Sigma$ code with an efficient $(\epsilon, \ell = \frac{8}{\epsilon})$ list decoding algorithm, meaning there exist a $\mathrm{poly}(|V_0|)$ time algorithm, which on input w it returns all codewords in $ENC \circ C$ at distance ϵ from w.

3.2 The list decoding algorithm

The input to the list decoding algorithm is a received word $y : V_1 \to \Sigma$ which we interpret as $(f_S)_{S \in V_1}$ by setting $f_S = y(S)$[5]. We are promised that there is some $g : V_0 \to \{0,1\}$ so that $\Pr_S[f_S = g|_S] \geq \epsilon$.

1. **Approximate List Decoding of Local Views**

[5] A subtle point is that if S is repeated several times in V_1, then $y(S)$ can take several possibly different values, per each repetition of S. In this case f_S will consist not of a single "deterministic" value in $\{0,1\}^S$ but rather a distribution over the different values. For clarity of presentation we will ignore this issue and treat f_S as if it were concentrated on one value for each S. The reader can check that our voting scheme works just as well if f_S were a distribution over values.

For every $T \in V_2$ we construct a list L_T of size $O(\frac{1}{\epsilon})$ of well-separated elements such that every g that has ϵ agreement with $(f_S)_{S \in V_1}$ on $(\Pi_1 | \Pi_2 = T)$ is very close to one of the elements in the list. More precisely,

Lemma 1. Let $\rho = \frac{\epsilon_0}{2} 10^{-\frac{8}{\epsilon}} \geq 0$. There exists a decoding algorithm that given $T \in V_2$ returns a list $L_T \subset \{0,1\}^T$ of size at most $\ell = \frac{8}{\epsilon}$ and a radius $r_T \in \{\rho \cdot 10^i\}_{i \leq \frac{8}{\epsilon}}$, such that:

- If $g : T \to \{0,1\}$ is such that $\Pr_{S \sim (\Pi_1 | \Pi_2 = T)}[f_S = g_{|S}] \geq \frac{\epsilon}{2}$ then there exits $\sigma \in L_T$ such that
$$\mathrm{dist}_T(\sigma, g) \leq \frac{r_T}{9}.$$
- For every $\sigma, \sigma' \in L_T$, $\mathrm{dist}_T(\sigma, \sigma') \geq r_T$.

We prove the lemma in Section 4. We let $V_2^{(i)}$ be the set of all $T \in V_2$ with $r_T = \rho 10^i$.

2. **Creating a UG Constraint Graph**
   - We define the graph $G_C = (V_C = V_2, E_C, w)$, to be the two step walk of the weighted bipartite graph $(X(V_2, V_1), \Pi_2)$ (see Definition 2.6).
   - We label every vertex with labels from $[\ell]$.
   - For every $e = (T_1, T_2) \in E_C$ we define a constraint permutation π_e as follows. Suppose $L_{T_1} = \sigma_1, \ldots \sigma_\ell$ and $L_{T_2} = \tau_1 \ldots \tau_\ell$. We choose $S \in V_2$ according the distribution $(\mathcal{S} | \mathcal{T}_1 = T_1, \mathcal{T}_2 = T_2)$. We go over $i \in [\ell]$: if there exists an unmatched $\tau_j \in L_{T_2}$ such that
   $$\mathrm{dist}_S(\sigma_i, \tau_j) \leq \frac{r_{T_1}}{2},$$
   we set $\pi(i) = j$. At the end, for every unmatched $i \in [\ell]$ we set $\pi(i)$ to an arbitrary unmatched label.

3. **Finding a large expanding UG constraint subgraph**

   For every $0 \leq i \leq \frac{8}{\epsilon}$ such that $\mu_G(V_2^{(i)}) \geq 10 \min\{\sqrt{\alpha_{2,1}}, \sqrt{\delta_{2,1}}\}$ let $G_C^{(i)}$ be the induced subgraph of G on $V_2^{(i)}$. By Theorem 2.8 we can find $V_i \subset V_2^{(i)}$ such that:
   - $\Pr_{T \sim \Pi_2}[T \in V_i] \geq \frac{1}{4} \Pr_{T \sim \Pi_2}[T \in V_2^{(i)}]$.
   - Denote by G_i the induced subgraph of G_C on V_i. Then, $\lambda(G_i) \leq \frac{99}{100}$.

4. **Solving the Unique Constraints**

   For every i as above, we apply Theorem 2.15 on G_i and get a list L of assignments. For each assignment $a : V_2^{(i)} \to [\ell]$ in the list L we define $w \in \{0,1\}^{V_0}$ by doing: for every $j \in V_0$ we pick $T \sim (\Pi_2 | \Pi_0 = j)$ and let $w_j = a(T)|_j$. We run the unique decoding algorithm of the code C on w and output its result.

The algorithm we described is randomized, but it can easily be derandomized since the random choices are local and we can enumerate over them in parallel. For example, in the step of constructing the constraint graph, each constraint is constructed randomly, but these random choices can clearly be enumerated over in parallel for all constraints.

3.3 Proof of correctness

Proof. (of Theorem 3.1) Let y be the given input, such that the interpretation of y as $(f_S)_{S\in V_1}$ has ϵ agreement with some codeword of $ENC \circ C$, i.e., there exists a function $g : V_0 \to \{0,1\}$ such that

$$\Pr_{S\sim\Pi_1}[f_S = g_{|S}] \geq \epsilon.$$

For each T, let L_T be the list from Item 1 of the decoding algorithm, and r_T the radius. The list and radius satisfy the conclusion of Lemma 1 above.

Let $G_C = (V_C, E_C, w = \{w_e\}_{e\in E_C})$ be the constraint graph described in Item 2 of the decoding algorithm, with the edge constraints $\pi_e : [\ell] \to [\ell]$ for every $e \in E_C$. Recall that $V_2^{(1)}, V_2^{(2)}, \ldots V_2^{(\frac{8}{\epsilon})}$ is a partition of $V_2 = V_C$ according to the list radius r_T.

Definition 3.2. A constraint π_e for $e = (T_1, T_2) \in E_C$ is *correct with respect to* g if there exist $\sigma_i \in L_{T_1}$ and $\sigma_j \in L_{T_2}$ such that:

- $dist_{T_1}(\sigma_i, g) \leq \frac{r_{T_1}}{9}$.
- $\text{dist}_{T_2}(\sigma_j, g) \leq \frac{r_{T_2}}{9}$.
- $\pi_e(i) = j$.

In Section 5 we prove:

Lemma 2. With high probability ($\exp(-n)$) there exists a $0 \leq i \leq \frac{8}{\epsilon}$ such that the graph $G_C^{(i)}$ satisfies

- $\mu_{G_C}(V_2^{(i)}) \geq \frac{\epsilon}{16}$.
- $\Pr_{(T_1,T_2)\sim w}[\pi_{(T_1,T_2)} \text{ is correct with respect to } g | T_1, T_2 \in V_2^{(i)}] \geq 1 - 10(\delta_{1,2} + \ell\delta_{local})$.

By Theorem 2.8, there exists a subset $V_i \subset V_2^{(i)}$ such that $\mu_{G_C}(V_i) \geq \frac{\epsilon}{64}$, and $\lambda(G(V_i)) \geq \frac{99}{100}$ (indeed the conditions for applying the theorem hold by our choice of parameters, $\frac{\epsilon}{16} \geq 10\min\{\sqrt{\alpha_{2,1}}, \sqrt{\delta_{2,1}}\}$). Furthermore, since $\mu_{G_C}(V_i) \geq \frac{1}{4}\mu_{G_C}(V_2^{(i)})$, the probability of an incorrect edge increases at most 4-fold:

$$\begin{aligned}\Pr[\pi_{(T_1,T_2)} \text{ is not correct} | T_1, T_2 \in V_i] &= \frac{\Pr[\pi_{(T_1,T_2)} \text{ is not correct} | T_1, T_2 \in V_2^{(i)}]}{\Pr[T_1, T_2 \in V_i \mid T_1, T_2 \in V_2^{(i)}]} \\ &\leq 4 \cdot (10(\delta_{1,2} + \ell\delta_{local}))\,,\end{aligned}$$

where correct means correct with respect to g.

Applying the UG algorithm (Theorem 2.15) on the graph G_i, the graph induced by V_i, we receive a list of assignments. Theorem 2.15 guarantees that there exists an assignment a in the list such that

$$\Pr_{T\sim\Pi_2}[\text{dist}_T(a(T), g) \leq \frac{r_T}{9}] \geq 1 - c^\ell 40(\delta_{1,2} + \ell\delta_{local}).$$

Recall our definition of w in Item 4 of the list decoding algorithm. It follows that $\Pr_{j\sim\Pi_0,\, T\sim(\Pi_2|\Pi_0=j)}[w_j \neq g(j)] \leq \frac{r_T}{9} + c^\ell 40(\delta_{1,2} + \ell\delta_{local}) \leq \frac{\epsilon_0}{2}$. Hence, with high probability (at least half for sure), $\text{dist}(w, g) \leq \epsilon_0$ and we output g. If we repeat the process t times independently,the probability that we fail reduces to 2^{-t}.

□

The proof makes it clear that our algorithm in fact first finds an approximate list decoding, as claimed in Theorem 1.3, and then runs the decoder for C as a black box to yield the list-decoding as claimed in Theorem 1.2, (and more precisely in Theorem 3.1).

4 Approximate list decoding local views

We are given $(f_S)_{S\in V_1}$. For every $T \in V_2$ we return a short list L_T of size $O(\frac{1}{\epsilon})$ of well-separated elements such that any g that has ϵ agreement with $(f_S)_{S\in V_1}$ on $(\Pi_1|\Pi_2 = T)$ is very close to one of the elements in the list. More precisely,

We repeat Lemma 1 and prove it.

Lemma 1. Let $\rho \geq 0$. There exists a decoding algorithm that given $T \in V_2$ returns a list $L_T \subset \{0,1\}^T$ of size at most $\ell = \frac{8}{\epsilon}$ and a radius $r_T \in \{\rho \cdot 10^i\}_{i\leq \frac{8}{\epsilon}}$, such that:

- If $g : T \to \{0,1\}$ is such that $\Pr_{S\sim(\Pi_1|\Pi_2=T)}[f_S = g_{|S}] \geq \frac{\epsilon}{2}$ then there exits $\sigma \in L_T$ such that
$$\mathrm{dist}_T(\sigma, g) \leq \frac{r_T}{9}.$$
- For every $\sigma, \sigma' \in L_T$, $\mathrm{dist}_T(\sigma, \sigma') \geq r_T$.

4.1 The approximate list decoding algorithm

We are given as input $T \in V_2$ and ρ. We remind the reader that $V_2 \subseteq \binom{V_0}{m_2}$ and $V_1 \subseteq \binom{V_0}{m_1}$.

Set an initial set : For every $\sigma \in \{0,1\}^T$ insert σ to L_0 if $\Pr_{S\sim(\Pi_1|\Pi_2=T)}[f_S = \sigma_{|S}] \geq \frac{\epsilon}{2}$.

Pruning : Set $i = 0, r_i = \rho$. Repeat:

1. Say $v_1, v_2 \in L_i$ are independent if $\mathrm{dist}_T(v_1, v_2) \geq r_i$. Find a maximal independent set L_{i+1}.
2. If $L_i = L_{i-1}$ quit the loop and output L_i, r_i.
3. Set $r_{i+1} = 10r_i$, $i = i + 1$ and repeat the loop.

4.2 Bounding the list size

Lemma 3. $|L_1| \leq \frac{8}{\epsilon}$.

Proof. We look at the weighted bipartite graph $X_{|T}$ as defined in Definition 2.10. By our assumption on the double sampler (X, Π_2) we know that the graph $(X_{|T} = (U, T, E), W_T)$ is an $(\alpha_{local}, \delta_{local})$ sampler.

Denote $\ell_1 = |L_1|$.

Claim 4.1. For every $\sigma \neq \sigma' \in L_1$,
$$\Pr_{S\sim(\Pi_1|\Pi_2=T)}[f_S = \sigma_{|S} = \sigma'_{|S}] \leq \delta_{local}.$$

Proof.

- Let $B \subseteq T$ be $B = \left\{i \in T \,\middle|\, \sigma_i \neq \sigma'_i\right\}$. Recall that W_T induces a uniform distribution on T (this is because (X, W) is regular, see Definition 2.10). Therefore, after the first pruning step,
$$\Pr_{i\sim(\Pi_0|\Pi_2=T)}[i \in B] = \Pr_{i\in T}[i \in B] \geq \mathrm{dist}_T(\sigma, \sigma') \geq \rho.$$
- Let $A \subseteq U$ be all sets S such that $f_S = \sigma_{|S} = \sigma'_{|S}$.

By definition, for every $S \in A$, $S \cap B = \emptyset$. Since G is a $(\alpha_{local}, \delta_{local})$ sampler and $\alpha_{local} < \rho$,

$$\Pr_{S \sim (\Pi_1 | \Pi_2 = T)}[S \in A] \leq \delta_{local}.$$

□

Now, for every $\widehat{L_1} \subseteq L_1$ of size $\ell \leq \ell_1$,

$$\begin{aligned}
1 \geq & \Pr_{S \sim (\Pi_1 | \Pi_2 = T)}[\exists \sigma \in \widehat{L_1} \textbf{ s.t } f_S = \sigma_{|S}] \\
\geq & \sum_{\sigma \in \widehat{L_1}} \Pr_{S \sim (\Pi_1 | \Pi_2 = T)}[f_S = \sigma_{|S}] - \sum_{\sigma \neq \sigma' \in \widehat{L_1}} \Pr_{S \sim (\Pi_1 | \Pi_2 = T)}[f_S = \sigma_{|S} = \sigma'_{|S}] \qquad \text{(using Claim 4.1)} \\
\geq & \ell \frac{\epsilon}{2} - \ell^2 \frac{\delta_{local}}{2}.
\end{aligned}$$

Thus, $\ell^2 \frac{\delta_{local}}{2} - \ell \frac{\epsilon}{2} + 1 \geq 0$ for all $0 \leq \ell \leq \ell_1$. Looking at the parabola $\ell_1^2 \frac{\delta_{local}}{2} - \ell_1 \frac{\epsilon}{2} + 1$ has discriminant $\frac{\epsilon^4}{4} - 2\delta_{local}$ and therefore two real intersections with 0 (because $\delta_{local} < \frac{\epsilon^2}{8}$), say $z_0 < z_1$. As $\ell^2 \frac{\delta_{local}}{2} - \ell \frac{\epsilon}{2} + 1 \geq 0$ for all $0 \leq \ell \leq \ell_1$ we must have $\ell_1 \leq z_0 = \frac{1}{2\delta_{local}} \left(\frac{\epsilon}{2} - \sqrt{\frac{\epsilon^2}{4} - 2\delta_{local}} \right) \leq \frac{8}{\epsilon}$.

□

This establishes that L_1 is short, but it might not yet be well-separated, which is the reason for the loop in the pruning step.

4.3 The closeness property

We next show that after all the pruning, the list still contains an element approximating any string with large initial support.

Claim 4.2. Let L be the list output by the algorithm and let r be the radius. For every $\sigma' \in \{0,1\}^T$ such that $\Pr_{S \sim (\Pi_1 | \Pi_2 \mathcal{T})}[f_S = \sigma'_{|S}] \geq \frac{\epsilon}{2}$ there exists $\sigma \in L$ such that $\text{dist}_T(\sigma, \sigma') \leq \frac{r}{9}$.

Proof. Fix any $\sigma' \in \{0,1\}^T$ such that $\Pr_{S \sim (\Pi_1 | \Pi_2 \mathcal{T})}[f_S = \sigma'_{|S}] \geq \frac{\epsilon}{2}$. Let L_0 be the initial list of the algorithm. By definition of L_0 we have that $\sigma' \in L_0$. Suppose we run the pruning algorithm K steps, then L_0 is the initial set, $L_K = L$ the final set, and $L_K \subsetneq L_{K-1} \subsetneq L_0$. In particular $K \leq |L_1| \leq \frac{8}{\epsilon}$.

Let $\sigma_0 = \sigma' \in L_0$. Let $\sigma_1 \in L_1$ be the element closest to σ_0. We know that $\text{dist}_T(\sigma_0, \sigma_1) < r_1$. Similarity, let $\sigma_i \in L_i$ be the element closest to $\sigma_{i-1} \in L_{i-1}$. We know that $\text{dist}_T(\sigma_{i-1}, \sigma_i) < r_i$. Now,

$$\text{dist}_T(\sigma', \sigma_K) = \text{dist}_T(\sigma_0, \sigma_K) \leq \sum_{i=1}^{K} \text{dist}_T(\sigma_{i-1}, \sigma_i) \leq \sum_{i=1}^{K} r_i = r_K \sum_{i=1}^{K} 10^{i-K-1} \leq \frac{r_K}{9}.$$

□

4.4 Proof of Lemma 1

It is clear from the algorithm description that when the algorithm terminates the distance between every two elements in L is at least $r_K = r$. We saw that there exists some $\sigma_K \in L_K$ such that $\text{dist}(\sigma', \sigma_K) \leq \frac{r_K}{9}$. Also, $|L_K| \leq |L_1| \leq \frac{8}{\epsilon}$, and $K \leq |L_1|$ so $r_K \leq \rho \cdot 10^{8/\epsilon}$.

5 Creating the Constraint Graph

5.1 The setting

Let $(X = (V_2, V_1, V_0), W)$ be the double sampler used for encoding. Recall that W defines a distribution Π over paths of (x, S, T) **s.t** $x \in V_0, S \in V_1, T \in V_2$ in X. For each vertex $T \in V_2$ there is a short list L_T of cardinality exactly ℓ (if the list is shorter just add dummy strings that obey the distance requirements) and a radius r_T. We now describe a randomized algorithm that outputs a constraint graph.

The graph : The graph is the two-step walk graph obtained from $X(V_2, V_1)$ as per Definition 2.6 (the vertices are V_2 and the edges correspond to choosing a random $S \sim \Pi_1$ and then T_1, T_2 independently from the distribution $(\Pi_2|\Pi_1 = S)$.) We call this graph $G_C = (V_C = V_2, E_C)$.

The labels : For every T, the label set is L_T (note that $|L_T| = \ell$).

The constraints : We let the (correlated) random variables $\mathcal{T}_1, \mathcal{S}, \mathcal{T}_2$ be defined by sampling S according to Π_1, and then sampling T_1, T_2 independently from $(\Pi_2|\Pi_1 = S)$.

Given an edge $(T_1, T_2) \in E_C$ chose a random subset $S \subset T_1 \cap T_2$ according to the distribution $(\mathcal{S} \mid \mathcal{T}_1 = T_1, \mathcal{T}_2 = T_2)$.

1. For every $\sigma \in L_{T_1}$ if there is an unmatched $\sigma' \in L_{T_2}$ such that
$$\text{dist}_S(\sigma, \sigma') \leq \frac{r_{T_1}}{2},$$
then set $\pi(\sigma) = \sigma'$.
2. For every unmatched $\sigma \in L_T$, set $\pi(\sigma)$ to an arbitrary unmatched label.

Observe that we always output unique constraints, because we only set $\pi(\sigma)$ to an unmatched label.

Fix $g : V_0 \to \{0,1\}$ such that $\Pr_{S\sim\Pi_1}[f_S = g_{|S}] \geq \epsilon$. In this section we prove Lemma 2, restated bellow. To simplify the notations, in this section "correct" always means correct with respect to the fixed function g.

Definition 5.1 (Correct list index)**.** For $T \in V_2$. Let $C(T)$ be the element in L_T closest to $g|_T$. Ties are broken arbitrarily. We say π is *correct* on $e = (T_1, T_2) \in E$ if $\text{dist}_{T_1}(C(T_1), g) \leq \frac{r_{T_1}}{9}$, $\text{dist}_{T_2}(C(T_2), g) \leq \frac{r_{T_1}}{9}$ and $\pi_e(C(T_1)) = C(T_2)$.

Lemma 4 (Lemma 2 restated)**.** Let G_C be the constraint graph as above. Then with high probability (over the randomness of the algorithm) there exists $i \in \{1, \ldots \frac{8}{\epsilon}\}$, such that

- $G_C^{(i)}$ **is large:** $\Pr_{v\sim\Pi_2}[v \in V_C^{(i)}] \geq \frac{\epsilon}{16}$.
- **Almost all edges in** $G_C^{(i)}$ **are correct:** $\Pr_{(T_1,T_2)}[\pi_{(T_1,T_2)} \text{ is correct} | T_1, T_2 \in V_C^{(i)}] \geq 1 - 10\delta_{1,2} - 10\ell\delta_{local}$, where (T_1, T_2) is a random edge selected according to the edge weights.

Recall that we partitioned the vertices according to their radius and defined $V_C^{(i)} = \{T \in V_C \mid r_T = \rho 10^i\}$, and $G_C^{(i)} = (V_C^{(i)}, E_C^{(i)})$ the induced subgraph of G_C on $V_C^{(i)}$.

Definition 5.2. A vertex $T \in V_2$ is *good* if there exists $\sigma \in L_T$ such that $\text{dist}_T(g, \sigma) \leq \frac{r_T}{9}$.

Claim 5.3. $\Pr_{T\sim\Pi_2}[T \text{ is good}] \geq 1 - \delta_{1,2}$.

Proof. We look at the bipartite graph $G = (V_2, V_1, E)$. Let

$$\begin{aligned} B &= \{S \in V_1 \mid f_S = g_S\}, \\ A &= \{T \in V_2 \mid \Pr_{S\sim(\mathcal{S}|\mathcal{T}_1=T)}[S \in B] < \frac{\epsilon}{2}\}. \end{aligned}$$

By assumption $\Pr_{S\sim\Pi_1}[S \in B] \geq \epsilon$. As $(G(V_2, V_1), (\Pi_2, \Pi_1))$ is a $(\frac{\epsilon}{2}, \delta_{1,2})$ sampler, $\Pr_{T\sim\Pi_2}[T \in A] \leq \delta_{1,2}$. For any $T \notin A$, by Lemma 1 there exists some $\sigma \in L_T$ such that $\mathrm{dist}_T(\sigma, g) \leq \frac{r_T}{9}$. □

5.2 Good edges in the bipartite graph $G(V_2, V_1)$

Definition 5.4. Let $S \subseteq T$ with $S \in V_1$, $T \in V_2$. Suppose L_T, r_T are the list and radius of T, and let $\sigma = C(T)$. We say that the edge (S, T) is *good* if

$$\begin{aligned} \mathrm{dist}_T(\sigma, g) &\leq \frac{r_T}{9}, && (1) \\ \mathrm{dist}_S(\sigma, g) &\leq \frac{r_T}{4}, && (2) \\ \forall \sigma' \neq \sigma \quad \mathrm{dist}_S(\sigma', g) &> \frac{3}{4} r_T. && (3) \end{aligned}$$

Notice that the edges (T, S) are edges in the sampler graph $G(V_2, V_1)$ and not in the constraint graph G_C.

Claim 5.5.

$$\Pr_{(T,S)\sim(\Pi_2,\Pi_1)}[(T, S) \text{ is good}] \geq 1 - \delta_{1,2} - \ell\delta_{local}\,.$$

Proof. First we show that for a good vertex $T \in V_2$, almost all edges (S, T) are good. Fix a good T, and denote $\sigma = C(T)$, since T is good requirement (1) holds, i.e. $\mathrm{dist}_T(\sigma_1, g) \leq \frac{r_T}{9}$.

Let $X_{|T} = (U, T, E)$ be the weighted bipartite graph defined in Definition 2.10. Define $B \subseteq T$ and $A \subseteq U$ by:

$$\begin{aligned} B &= \{i \in T \mid \sigma_i \neq g_i\}\,. \\ A &= \left\{S \in U \;\middle|\; \Pr_{i\in S}[i \in B] > \frac{r_T}{4}\right\}. \end{aligned}$$

Since T is good,

$$\Pr_{i\sim(\Pi_0|\Pi_2=T)}[i \in B] = \Pr_{i\in T}[i \in B] \leq \frac{r_T}{9},$$

where the equality is because X is regular. Since G is a $(\frac{\rho}{9}, \delta_{local})$ sampler and $r_T \geq \rho$, we conclude that $\Pr_{S\sim(\Pi_1|\Pi_2=T)}[S \in A] \leq \delta_{local}$. For any $S \notin A$, the edge (S, T) satisfies (2).

For (3), for every $\sigma' \in L_T, \sigma' \neq \sigma$ we define $B_{\sigma'} \subseteq T$ and $A_{\sigma'} \subseteq U$:

$$\begin{aligned} B_{\sigma'} &= \{i \in T \mid (\sigma_j)_i \neq g_i\} \\ A_{\sigma'} &= \left\{S \in U \;\middle|\; \Pr_{i\in S}[i \in B_j] \leq \frac{3r_T}{4}\right\}. \end{aligned}$$

From Lemma 1, $\mathrm{dist}_T(\sigma', \sigma) \geq r_T$, which implies $\mathrm{dist}_T(\sigma', g) \geq \mathrm{dist}_T(\sigma', \sigma) - \mathrm{dist}_T(\sigma, g) \geq \frac{8r_T}{9}$. Therefore, $\Pr_{i\in T}[i \in B_j] \geq \frac{8r}{9}$. The graph $X_{|T}$ is a $(\frac{\rho}{9}, \delta_{local})$ sampler, so $\Pr_{S\sim(\Pi_1|\Pi_2=T)}[S \in A\sigma'] \leq \delta_{local}$.

For every $S \subset T$ such that $S \notin A_{\sigma'}$ for all $\sigma' \in L_T$, the edge (S,T) is good. This happens with probability at most $\ell\delta_{local}$.

To finish the proof, we use Claim 5.3 which proves that $T \sim \Pi_2$ is good with high probability:

$$\begin{aligned}\Pr_{(S,T)\sim(\Pi_1,\Pi_2)}[(S,T) \text{ is good}] \geq &\Pr_T[T \text{ is good}] \Pr_{(S,T)}[(S,T) \text{ is good}|T \text{ is good}] \\ \geq &1 - \delta_{1,2} - \ell\delta_{local}.\end{aligned}$$

□

5.3 Good edges implies correctness

Lemma 5. Suppose that the constraint for (T_2, T_2) is constructed by choosing $S \subset T_1 \cap T_2$. If $(S,T_1), (S,T_2)$ are both good and $r_{T_1} = r_{T_2}$, then $\pi_{(T_1,T_2)}$ is correct.

Proof. Let $L_1 =, L_2$ be the lists of T_1, T_2 respectively, and denote $r = r_{T_1} = r_{T_2}$. To shorten the notations let $\sigma_1 = C(T_1)$ and $\sigma_2 = C(T_2)$. Since $(T_1,S), (T_2,S)$ are good, $\text{dist}_{T_1}(\sigma_1, g) \leq \frac{r}{9}, \text{dist}_{T_2}(\sigma_2, g) \leq \frac{r}{9}$.

We are left with proving that $\pi_{(T_1,T_2)}(\sigma_1) = \sigma_2$. The proof has two steps. First we show that the constraint algorithm on σ_1 outputs σ_2 if still unmatched. Then we show that no other $\sigma' \in L_1$ is matched to σ_2.

- For the first step, the edges $(S,T_1), (S,T_2)$ are both good, which implies for $j \in \{1,2\}$: $\text{dist}_S(\sigma_j, g) \leq \frac{r}{4}$. By the triangle inequality

  $$\text{dist}_S(\sigma_1, \sigma_2) \leq \text{dist}_S(\sigma_1, g) + \text{dist}_S(\sigma_2, g) \leq \frac{r}{4} + \frac{r}{4} \leq \frac{r}{2}.$$

  Hence the constraint algorithm will match σ_1 to σ_2 if still unmatched.

- For the second step of the proof let $\sigma' \in L_1, \sigma' \neq \sigma_1$. Since (S,T_2) is good, $\text{dist}_S(\sigma_2, g) \leq \frac{1}{4}r$. Since (S,T_1) is good, for every $\sigma' \neq \sigma_1$, $\text{dist}_S(\sigma', g) > \frac{3r}{4}$. It follows that

  $$\text{dist}_S(\sigma', \sigma_2) > \frac{r}{2}.$$

  Hence σ' cannot be matched to σ_2.

□

5.4 The large good component

Proof of Lemma 4. For every i let

$$p^{(i)} = \Pr_{(S,T)\sim(\Pi_1,\Pi_2)}[(S,T) \text{ is good}|T \in V_C^{(i)}].$$

Denote $\eta = \delta_{1,2} + \ell\delta_{local}$, by Claim 5.5,

$$1 - \eta \leq \Pr_{(S,T)\sim(\Pi_1,\Pi_2)}[(S,T) \text{ is good}] = \sum_i \Pr_{T\sim\Pi_2}[T \in V_C^{(i)}] \cdot p^{(i)}. \tag{4}$$

Claim 5.6. There exists $i \leq \frac{8}{\epsilon}$ with $\Pr_{T\sim\Pi_2}[T \in V_C^{(i)}] \geq \frac{\epsilon}{16}$ and $p^{(i)} \geq 1 - 2\eta$.

Proof. Assume towards contradiction that no such i exists. Denote $\ell = \frac{8}{\epsilon}$ and for every $i \in [\ell]$, $\mu(V_C^{(i)}) = \Pr_{T\sim\Pi_2}[T \in V_C^{(i)}]$. Let $q = \sum_{i:\mu(V^{(i)})<\frac{1}{2\ell}} \mu(V_C^{(i)})$. Since there are at most are $\ell = \frac{8}{\epsilon}$ distinct $V_C^{(i)}$'s, $q < \frac{1}{2}$. Therefore,

$$\begin{aligned}\sum_{i\in[\ell]} \mu(V_C^{(i)})p^{(i)} &= \sum_{i:\mu(V_C^{(i)})\geq\frac{1}{2\ell}} \mu(V^{(i)})p^{(i)} + \sum_{i:\mu(V^{(i)})<\frac{1}{2\ell}} \mu(V^{(i)})p^{(i)} \\ &< (1-2\eta)(1-q)+q \\ &= 1-2\eta(1-q) < 1-\eta.\end{aligned}$$

Which contradicts (4). □

Now, fix i such that $\mu(V_C^{(i)}) \geq \frac{1}{2\ell}$ and $p^{(i)} \geq 1-2\eta$ and look at the induced graph on $V_C^{(i)}$. By union bound,

$$\Pr_{(T_1,S,T_2)\sim(\mathcal{T}_1,\mathcal{S},\mathcal{T}_2)}[(T_1,S) \text{ or } (T_2,S) \text{ are not good} | T_1,T_2 \in V_C^{(i)}] \leq 4\eta. \tag{5}$$

Together with Lemma 5, this means that in expectation, $1-4\eta$ of the edge constraints are going to be correct. It remains to show that the same holds with high probability. We show it by Hoeffding inequality, because choice of S is independent for each edge and the edge distribution is rather flat.

For each edge $(T_1,T_2) \in E_C^{(i)}$ let $I_{(T_1,T_2)}(S)$ be the indicator random variable to (T_1,S) or (T_2,S) not being good. Denote by p_{T_1,T_2} the probability that $I_{T_1,T_2}(S)=1$, for $S \sim (\mathcal{S}|\mathcal{T}_1=T_1,\mathcal{T}_2=T_2)$.

By (5), $\mathbb{E}_{(T_1,T_2)}[I_{T_1,T_2}] \leq 4\eta$, where the expectation is on choosing a random edge in $E_C^{(i)}$ according to the edge weights.

The random variables $\{I_{T_1,T_2}\}$ are independent (because the algorithm chooses a set $S \sim (\mathcal{S}|\mathcal{T}_1 = T_1,\mathcal{T}_2=T_2)$ independently for each edge (T_1,T_2)).

Let $w = \{w_{T_1,T_2}\}$ be the distribution over picking an edge $(T_1,T_2) \in E_C^{(i)}$. This distribution is D^2-flat, because the distribution Π_2 over V_C is D-flat. Using the Hoeffding inequality,

$$\Pr\left[\sum_{(T_1,T_2)} w_{T_1,T_2}I_{T_1,T_2} \geq 8\eta\right] \leq e^{-\frac{\eta^2}{\sum_{T_1,T_2} w_{T_1,T_2}^2}} \leq e^{-\frac{\eta^2|V_C^{(i)}|}{D^4}} \leq e^{-c'n},$$

for a constant $c' \leq \frac{\epsilon}{16D^4}$.

So with high probability, $\Pr_{(T_1,S,T_2)\sim(\mathcal{T}_1,\mathcal{T}_2)}[(T_1,T_2) \text{ is correct} | T_1,T_2 \in V_C^{(i)}] \geq 1-8\eta$. □

6 High Dimensional Expanders yield Double Samplers

In this section we prove Theorem 2.12, relying on high dimensional expanders.

6.1 Preliminaries: Second eigenvalue for a weighted bipartite graph

Let $G=(U,V,E)$ be a weighted bipartite graph with edge weights $\{w_e\}_{e\in E}$. We define a Markov operator associated with a random walk in the bipartite graph (see [DK17]).

The edge weights $\{w_e\}_{e\in E}$ imply a marginal distribution over the vertices U and V given by $w_u = \sum_{v\in V} w_{u,v}$ and $w_v = \sum_{u\in U} w_{u,v}$. All expectations over the vertex sets U,V are done with respect to these distributions defined by w_u and w_v. We define an inner product space on $L^2(U)$, for every two functions $f,f' : U \to \mathbb{R}$,

$$\langle f,f'\rangle = \mathop{\mathbb{E}}_u[f(u)f'(u)] := \sum_{u\in U} w_u f(u)f'(u).$$

We define similar expectation and inner product space for V.

The natural linear operator corresponding to the graph G is $A : L^2(U) \to L^2(V)$ defined as follows. For every $f : U \to \mathbb{R}$,

$$(Af)(v) = \mathop{\mathbb{E}}_{u|v}[f(u)] = \sum_{u \in U} w_{u,v} f(u).$$

In matrix form, $(A)_{u,v} = \frac{w_{u,v}}{w_u}$. It can be verified that for every $f \in L^2(U), g \in L^2(v)$:

$$\langle Af, g \rangle = \sum_{u \in U, v \in V} w_{u,v} f(u) g(v).$$

It holds that $A\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the vector in which all its entries equals 1. We define

$$\lambda(G) = \sup_{f \in \mathbb{R}^V \setminus \{0\}, f \perp \mathbf{1}} \frac{\|Af\|}{\|f\|}.$$

In the case of a non bipartite weighted graph, it is possible to define the same operators by identifying U and V, and this coincides with the (normalized) second largest eigenvalue in absolute value.

The following is quite standard, and very similar to the proof of the expander mixing lemma, see e.g. [DK17, Proposition 2.8].

Claim 6.1. Let $f : V \to [0,1]$ and let $g : U \to [0,1]$. Then

$$\left| \mathop{\mathbb{E}}_{uv \in E}[f(v)g(u)] - \mathbb{E}[f]\,\mathbb{E}[g] \right| \leq \lambda \sqrt{\mathbb{E}[f]\,\mathbb{E}[g]}.$$

6.2 High Dimensional Expanders

A d-dimensional complex X is a non-empty collection of sets of a ground set $[n]$. The sets have size at most $d+1$. We call a set of size $i+1$ an *i-dimensional face* (or i-face for short), and denote the collection of all i-faces by $X(i)$. A d-dimensional complex X is *pure* if every i-face is a subset of some d-face. We will only be interested in pure complexes.

Let X be a pure d-dimensional complex. Given a probability distribution $\mathcal{D}_d$ on its top-dimensional faces $X(d)$, for each $i < d$ we define a distribution $\mathcal{D}_i$ on the i-faces using the following experiment: choose a top-dimensional face according to $\mathcal{D}_d$, and remove $d-i$ points at random. We can couple all of these distributions to a random vector $\mathcal{D} = (\mathcal{D}_d, \ldots, \mathcal{D}_{-1})$ of which the individual distributions are marginals.

For every i-dimensional face $s \in X(i)$ for $i < d-1$, consider the following weighted graph X_s defined as follows:

- The vertices are points $x \notin s$ such that $s \cup \{x\} \in X(i+1)$ is a face.
- The edges are pairs of points $\{x, y\}$ such that $s \cup \{x, y\} \in X(i+1)$ is a face. (Since the complex is pure, x and y are vertices.)
- The weight of the edge $\{x, y\}$ is

$$w_s(x,y) := \Pr_{t \sim \mathcal{D}_{i+1}}[t = (s \cup \{x,y\}) \mid t \supset s].$$

Note that the weights define a probability distribution w_s on the edges. We denote the marginal of w_s on its first coordinate by

$$w_s(x) := \sum_{y \neq x} w_s(x,y) = \Pr_{t \sim \mathcal{D}_{i+1}}[t \supset s \cup \{x\} \mid t \supset s] = \Pr_{(u,v) \sim (\mathcal{D}_{i+1}, \mathcal{D}_i)}[u = s \cup \{x\} \mid v = s].$$

This is also the marginal of w_s on its second coordinate.

There are several different definitions of high dimensional expansion. For our purposes, the most relevant is the two-sided spectral expansion as defined in [DK17], where it is called "γ-HD":

Definition 6.2 (two-sided high dimensional spectral expander)**.** A d-dimensional complex is said to be a γ-*two-sided spectral expander* if for every $i < d-1$, and every face $s \in X(i)$ the graph X_s is an expander with $|\lambda(X_s)| \leq \gamma$.

Lubotzky, Samuels and Vishne [LSV05] constructed an explicit family of Ramanujan complexes. The LSV complexes themselves are only *one*-sided spectral expanders in that they give $\lambda(X_S) \leq \gamma$ but not $-\gamma \leq \lambda(X_S) \leq \gamma$ as we require. In the proof of [DK17, Lemma 1.5] it is shown that if we truncate a k-dimensional LSV complex and leave only $X(0), \ldots, X(d)$ for $d/k < \gamma/2$ then we get a γ-two-sided spectral high-dimensional expander.

Lemma 6 (Lemma 1.5 in [DK17])**.** For every $\gamma > 0$ and every $d \in \mathbb{N}$ there exists an explicit infinite family of bounded degree d-dimensional complexes which are γ-two-sided spectral expanders.

Recall that a distribution is said to be D-flat for an integer $D \in \mathbb{N}$ if there is some R such that each of its atomic probabilities belong to the set $\{\frac{1}{R}, \ldots, \frac{D}{R}\}$ (see Definition 2.11).

Claim 6.3. The complexes from Lemma 6 have top distribution $\mathcal{D}_d$ that is D-flat, for $D \leq (1/\gamma)^{O(d^2/\gamma^2)}$. Furthermore, the ratio $|X(d)|/|X(0)|$ is also bounded by D.

Proof. (sketch) These complexes come from truncation of a $k = 2d/\gamma$ - dimensional LSV complex, whose top distribution is uniform. The top distribution of the truncation, $\mathcal{D}_d$, comes from the *uniform* distribution on $X(k)$ so the probability of each d-face can be written as the number of k-faces containing it divided by $|X(k)| = R$. To bound the maximal number of k-faces containing a fixed face we observe that it is at most the size of a "link" which is isomorphic to a spherical building, therefore, it is bounded by number of linear subspaces in a k-dimensional vector space over a field whose size is roughly $1/\gamma^2$. This number is at most $(1/\gamma)^{O(k^2)}$ giving the parameters as claimed. □

6.3 Construction of double samplers

Let X be a d-dimensional simplicial complex, with distribution $\mathcal{D}_d$ over $X(d)$. Fix $d \geq m_2 - 1 > m_1 - 1 \geq 0$ and define an inclusion graph as follows. Let $V_2 = X(m_2 - 1)$, $V_1 = X(m_1 - 1)$, and $V_0 = X(0)$ and look at the inclusion graph (V_2, V_1, V_0) together with the distribution Π_2 on V_2 obtained by selecting a top face according to $\mathcal{D}_d$ and then removing $d - m_2 + 1$ random elements from it.

Lemma 7 (Spectral version of double sampler)**.** Let $G_{2,1} = (V_2, V_1)$ and $G_{1,0} = (V_1, V_0)$ where $m_2 > 2$. The following spectral bounds hold,

- $\lambda^2(G_{1,0}) \leq 1/m_1 + O(m_1\gamma)$ and
- $\lambda^2(G_{2,1}) \leq m_1/m_2 + O(m_1 m_2 \gamma)$
- For every $T \in V_2$, $(\Pi_0 | \Pi_2 = T)$ is uniform on T. Furthermore, $\lambda^2(X_{|T}) \leq \frac{1}{m_1} \cdot \frac{m_2 - m_1}{m_2 - 1}$ where $X_{|T} = (U, T, E)$ is defined by $U = \{S \in V_1 \mid S \subseteq T\}$ and $(S, i) \in E$ for $S \in U$ and $i \in T$ iff $i \in S$.
- $\mathcal{D}_0$ in uniform on $X(0)$.

Moreover, the distribution Π_1 induced from Π_2 is D-flat, and $\frac{|X(d)|}{|X(0)|} \leq D$ for $D \leq (1/\gamma)^{O(d^2/\gamma^2)}$.

Proof. The first two items are proven in [DK17, Theorem 1.8] where the corresponding graphs are A_{0,m_1-1} and A_{m_1-1,m_2-1}. The regularity follows from the fact that LSV complexes induce a unform distrubution over the vertices. For the locality property we observe that for each $T \in V_2 = X(m_2 - 1)$ the graph $X_{|T}$ is a bipartite graph. On one side it has all $S \subset T$, $|S| = m_1$ and on the other side all elements in T.

The claim on the eigenvalue follows from considering the two step walk and noticing it is a convex combination of the identity matrix with probability $1/m_1 \cdot (m_2 - m_1)/(m_2 - 1)$ and the all-ones matrix (normalized) with remaining probability. The claim on the flatness and on the sizes follows from Claim 6.3. □

6.4 From spectral gaps to samplers

We use a variant of the expander mixing lemma, Claim 6.1, to deduce the sampler property from the spectral gaps.

Claim 6.4. A bipartite weighted graph (U, V, E) with second eigenvalue λ is an $(\alpha, \frac{\lambda^2}{\alpha^2})$ sampler. In other words, to get an (α, δ) sampler, it suffices to take a graph with $\lambda < \alpha\sqrt{\delta}$.

Proof. Let $f : V \to [0,1]$ have $\mathbb{E}[f] = \beta$. Let A be the set of vertices that see too little of f

$$A = \left\{ u \in U \;\middle|\; \mathop{\mathbb{E}}_{v|u}[f(v)] < \beta - \alpha \right\}.$$

Let B be the set of vertices that see too much of f,

$$B = \left\{ u \in U \;\middle|\; \mathop{\mathbb{E}}_{v|u}[f(v)] > \beta + \alpha \right\}.$$

We will show $\Pr[A], \Pr[B] \leq \lambda^2\beta/\alpha^2 \leq \lambda^2/\alpha^2$. Write

$$(\beta + \alpha)\Pr[B] \leq \mathop{\mathbb{E}}_{uv}[f(v)1_B(u)] \leq \mathbb{E}[f]\Pr[B] + \lambda\sqrt{\mathbb{E}[f]\Pr[B]}$$

where the first inequality is by definition of B and the second inequality is relying on Claim 6.1. Dividing both sides by $\sqrt{\mathbb{E}[f]\Pr[B]}$ and rearranging, we get $\frac{\Pr[B]}{\mathbb{E}[f]} \leq \lambda^2/\alpha^2$ so $\Pr[B] \leq \lambda^2\beta/\alpha^2$.

Similarly for A, by Claim 6.1

$$|\mathbb{E}[f]\Pr[A] - \mathop{\mathbb{E}}_{uv}[f(v)1_A(u)]| \leq \lambda\sqrt{\mathbb{E}[f]\Pr[A]}$$

so

$$\mathbb{E}[f]\Pr[A] - \lambda\sqrt{\mathbb{E}[f]\Pr[A]} \leq \mathop{\mathbb{E}}_{uv}[f(v)1_A(u)] \leq (\beta - \alpha)\Pr[A]$$

and again we get $\Pr[A] \leq \lambda^2\beta/\alpha^2$.

So, if we want an (α, δ) sampler, we choose $\lambda < \alpha\sqrt{\delta}$. □

6.5 Proof of Theorem 2.12

We restate the theorem for convenience.

Theorem 2.12. (restated) For every $\alpha_{2,1}, \delta_{2,1}, \alpha_{1,0}, \delta_{1,0}, \alpha_{local}, \delta_{local} > 0$ there exist $D, m_2, m_2 \in \mathbb{N}$ and a family of explicitly constructible double samplers (X_n, W_n) for infinitely many $n \in \mathbb{N}$ such that

- X_n is an inclusion graph (V_2, V_1, V_0) where $|V_0| = n$, $V_i \subseteq \binom{V_0}{m_i}$ for $i = 1, 2$, with distribution W_n over $V_2(X_n)$.
- X_n is a regular $((\alpha_{2,1}, \delta_{2,1}), (\alpha_{1,0}, \delta_{1,0}), (\alpha_{local}, \delta_{local}))$ double sampler.
- $|V_1|, |V_2| \leq D \cdot n$. The distributions Π_1, Π_2 are D-flat.

Proof. We construct a spectral double sampler as in Lemma 7, but first, let us choose the parameters. We choose $\gamma < 1/(m_1m_2)^2$ small enough so that the term $O(m_1m_2\gamma)$ is negligible wrt m_1/m_2 and wrt $1/m_1$. We choose m_1, m_2 so that $\lambda(G_{1,0})^2 < 2/m_1 < \min(\alpha_{1,0}^2\delta_{1,0}, \alpha_{local}^2\delta_{local},)$, and similarly $\lambda(G_{2,1})^2 < 2m_1/m_2 < \alpha_{2,1}^2\delta_{2,1}$. Summarizing:

- $m_1 = \min(1/2\alpha_{1,0}^2\delta_{1,0}, 1/2\alpha_{local}^2\delta_{local})$

- $m_2 = m_1/\alpha_{2,1}^2\delta_{2,1}$
- $D \leq \exp(\text{poly}(m_2)) \leq \exp(\text{poly}(1/\alpha_{1,0}, 1/\alpha_{2,1}, 1/\delta_{1,0}, 1/\delta_{2,1}, 1/\alpha_{local}, 1/\delta_{local}))$.

The size of V_2, V_1 is at most $D \cdot n$, and moreover, the distribution Π_1 is D-flat, as claimed in Lemma 7. According to Claim 6.4, the constructed graph is a $\alpha_{2,1}, \delta_{2,1}, \alpha_{1,0}, \delta_{1,0}, \alpha_{local}, \delta_{local}$ double sampler.

□

References

[ABN+92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. *Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs*. IEEE Trans. Inform. Theory, 38(2):509–516, 1992. 2, 10

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. *Unique games on expanding constraint graphs are easy: extended abstract*. In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 21–28. 2008. 4, 9, 29

[Chu05] Fan Chung. *Laplacians and the Cheeger inequality for directed graphs*. Ann. Comb., 9(1):1–19, Apr 2005. 23

[DG18] Irit Dinur and Yoav Gelberg. *A robustness result for expanders*, 2018. (unpublished). 23

[DK17] Irit Dinur and Tali Kaufman. *High dimensional expanders imply agreement expanders*. In *Proc. 58th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 974–985. 2017. 2, 5, 18, 19, 20

[GI02] Venkatesan Guruswami and Piotr Indyk. *Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets*. In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 812–821. 2002. 3

[GRS] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential coding theory*. (draft of book). 3

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. *Uniform direct product theorems: Simplified, optimized, and derandomized*. SIAM J. Comput., 39(4):1637–1665, 2010. (Preliminary version in *40th STOC*, 2008). 5, 6

[IKW12] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. *New direct-product testers and 2-query PCPs*. SIAM J. Comput., 41(6):1722–1768, 2012. (Preliminary version in *41st STOC*, 2009). 5, 6

[Imp03] Russell Impagliazzo. *Hardness as randomness: A survey of universal derandomization*. In *Proc. ICM 2002, Beijing*, volume 3, pages 659–672. 2003. 5

[KMRS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. *High-rate locally correctable and locally testable codes with sub-polynomial query complexity*. J. ACM, 64(2):11:1–11:42, 2017. (Preliminary version in *48th STOC*, 2016). 3

[KO18] Tali Kaufman and Izhar Oppenheim. *Construction of new local spectral high dimensional expanders*. In *Proc. 50th ACM Symp. on Theory of Computing (STOC)*, pages 773–786. 2018. 2

[LSV05] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. *Explicit constructions of Ramanujan complexes of type $\tilde{A}_d$*. European J. Combin., 26(6):965–993, 2005. 2, 5, 20

[MM10] Konstantin Makarychev and Yury Makarychev. *How to play unique games on expanders*. In *Proc. 8th International Workshop Approx. & Online Algorithms (WAOA)*, pages 190–200. 2010. 9, 26, 28, 29, 30, 31

[Ta-17] Amnon Ta-Shma. *Explicit, almost optimal, epsilon-balanced codes*. In *Proc. 49th ACM Symp. on Theory of Computing (STOC)*, pages 238–251. 2017. 6

[Tre05] Luca Trevisan. *On uniform amplification of hardness in NP*. In *Proc. 37th ACM Symp. on Theory of Computing (STOC)*, pages 31–38. 2005. 5

[Zuc97] David Zuckerman. *Randomness-optimal oblivious sampling*. Random Structures Algorithms, 11(4):345–367, 1997. (Preliminary version in *27th STOC*, 1996). 2

Appendices

A Large induced subgraphs of two-step walk on a sampler graph has an induced expander

Recall the two-step walk defined on Definition 2.6. We might expect the graph G created by a two-step walk on a sampler to be an expander, but this is not exactly true because small sets mights not expand. Nevertheless, we can prove a version of the expander mixing lemma for sampler, see Claim A.1. This claim allows us to prove that even though G is not an expander, every large induced subgraph of G contains an expanding subgraph.

Theorem 2.8. (restated) Let $\alpha, \beta, \delta \in (0,1)$ be such that $\alpha, \delta < \frac{\beta^2}{100}$. Let $(G_{samp} = (V_2, V_1, E_s), W_S)$ be an (α, δ) sampler. Let $(G = (V_2, E), W)$ be the two-step walk graph of G_{samp}. Let $A \subseteq V_2$ be any set with $\mu_G(A) \geq \beta$. Then there exists a set $B \subseteq A$ such that:

- $\mu_G(B) \geq \frac{\beta}{4}$.
- Let G_B be the induced graph of G on B. $\lambda(G_B) \leq \frac{99}{100}$.

Furthermore, given A, such a set B can be found in time polynomial in $|V|$.

The idea of finding the subgraph is based on [DG18], we gradually remove sparse cuts form G_B until reaching an expanding graph. We find sparse cuts using the proof of Cheeger inequality, which is constructive.

We present the following algorithm for finding the expanding subgraph. The algorithm receives a weighted graph $G = (V, E), W$ and a subset $A_0 \subset V$ such that $\mu_G(A_0) \geq \beta$ and outputs a set $B \subset V$ satisfying the conditions of Theorem 2.8.

Initialization : Set $i = 0$, let $G_0 = (A_0, V_0), W_0$ be the weighted graph induced by A_0.

Graph Improvement : While $\lambda(G_i) \geq \frac{99}{100}$:

1. Find a cut $(U_i, A_i \setminus U_i)$ in G_i such that $\mu_{G_i}(E(U_i, A_i \setminus U_i)) \leq \sqrt{2(1-\lambda(G_i))}\mu_{G_i}(U_i)$, let U_i be the smaller part of the cut, i.e. $\mu_{G_i}(U_i) \leq \frac{1}{2}$. See [Chu05] for the algorithm.
2. Set $A_{i+1} = A_i \setminus U_i$ and set $G_{i+1} = (A_{i+1}, E_{i+1}), W_{i+1}$ to be the graph induced by A_{i+1}, with the same edge weights W.
3. Increase i to $i+1$.

output Output $B = A_l$.

It is clear from the algorithm description that output graph G_B satisfies $\lambda(G_B) \leq \frac{99}{100}$, so we only need to show that $\mu_G(B)$ is large enough. We assume that a single vertex has $\lambda = 0$, so the algorithm always stops.

Before proving the theorem, let us state and prove prove an expander mixing lemma argument for our graph G. The graph G is a two step random walk on a sampler graph G_{samp}. Because G_{samp} is an (α, δ) sampler and not an expander, we can only prove that sets larger than α, δ are expanding.

Claim A.1. Let $(G_{samp} = (V_2, V_1, E_s), W_S)$ be an (α, δ) sampler, and let $(G = (V_2, E), W)$ be the two-step walk graph of G_{samp}.

Then for every $A, B \subset V_2$ which satisfy $\mu_G(A) > \delta, \mu_G(B) > \alpha$,

$$\Pr_{(T_1,T_2)\sim W}[T_1 \in A, T_2 \in B] \geq (\mu_G(A) - \delta)(\mu_G(B) - \alpha)$$
$$\Pr_{(T_1,T_2)\sim W}[T_1 \in A, T_2 \in B] \leq \mu_G(A)(\mu_G(B) + \alpha) + \delta.$$

Proof. Let $\Pi = (\Pi_2, \Pi_1)$ be the correlated distribution on V_2, V_1 in G_{samp}, as in Definition 2.4. We define the function $f : V_1 \to [0,1]$,

$$\forall v \in V_1, \quad f(v) = \Pr_{T \sim (\Pi_2 | \Pi_1 = v)}[T \in B].$$

In words, $f(v)$ is the probability of a random neighbor of v in G_{samp} to be in B. This implies that $\mathbb{E}_{v \sim \Pi_1}[f(v)] = \Pr_{T \sim \Pi_2}[T \in B] = \mu_G(B)$.

For every $T \in V_2$, let p_T be the probability of a random neighbour of T in G to be in B. Using f, $p_T = \Pr_{(T_1,T_2) \sim W}[T_2 \in B | T_1 = T] = \mathbb{E}_{v \sim (\Pi_1 | \Pi_2 = T)}[f(v)]$. The last equality is because G is a two step random walk on G_{samp}.

From the sampling properties of G_{samp},

$$\Pr_{T \sim \Pi_2}\left[\left|\mathop{\mathbb{E}}_{v \sim (\Pi_1 | \Pi_2 = T)}[f(v)] - \mathop{\mathbb{E}}_{v \sim \Pi_1}[f(v)]\right| > \alpha\right] \leq \delta,$$

Substituting the expectation over $f(v)$ by p_T, we get that $\Pr_{T \sim \Pi_2}[|p_T - \mu_G(B)| > \alpha] \leq \delta$.

Let $R \subset V_2$ be the set

$$R = \{T \in V_2 \mid \ |p_T - \mu_G(B)| > \alpha\}.$$

From above, $\mu_G(R) \leq \delta$. For every $T_1 \notin R$, $p_T \in [\mu_G(B) - \alpha, \mu_G(B) + \alpha]$.

Therefore,

$$\begin{aligned}\Pr_{(T_1,T_2) \sim W}[T_1 \in A, T_2 \in B] &\geq \Pr_{T_1 \in \Pi_2}[T_1 \in A \setminus R] \Pr_{(T_1,T_2) \sim W}[T_2 \in B | T_1 \in A \setminus R] \\ &\geq (\mu_G(A) - \delta)(\mu_G(B) - \alpha).\end{aligned}$$

$$\begin{aligned}\Pr_{(T_1,T_2) \sim W}[T_1 \in A, T_2 \in B] &\leq \Pr_{T_1 \sim \Pi_2}[T_1 \in R] + \Pr_{T_1 \sim \Pi_2}[T_1 \in A \setminus R] \Pr_{(T_1,T_2) \sim W}[T_2 \in B | T_1 \in A \setminus R] \\ &\leq \delta + \mu_G(A)(\mu_G(B) + \alpha).\end{aligned}$$

□

Proof of Theorem 2.8. Let B be the output of the algorithm, denote by l the number of steps the loop ran, so $B = A_l$. By the algorithm definition,the graphG_l satisfies $\lambda(G_l) \leq \frac{99}{100}$, so it remains to show that $\mu_G(G_l) \geq \frac{\beta}{4}$.

The main idea in the proof is that if each cut (A_i, U_i) is sparse in G_i, then for every $t \leq l$, there are few edges between A_t and $U_1 \cup U_2, \cdots \cup U_{t-1}$ in G. Using the expander mixing lemma, it means that for every t, either A_t or $U_1 \cup U_2, \cdots \cup U_{t-1}$ is very small. Since each U_i is chosen as the smallest part of the cut, it is not possible that $U_1 \cup U_2, \cdots \cup U_{i-1}$ is very small but $U_1 \cup U_2, \cdots \cup U_{i-1} \cup U_i$ is very large. This means that $U_1, \cdots \cup U_l$ is small, and finishes the proof.

We now write formally the idea described above. Assume towards contradiction that $\mu_G(A_l) \leq \frac{\beta}{4}$, this implies that $\mu_G(U_1 \cup \cdots \cup U_{l-1}) \geq \frac{3}{4}\beta$. Let j be the first index such that $\mu_G(U_1 \cup \cdots \cup U_j) \geq \frac{3}{4}\beta$, and denote by $U = U_1 \cup \cdots \cup U_{j-1}$ which means that $\mu_G(U) < \frac{3}{4}\beta$. We show that in fact U is very small $\mu_G(U) < \frac{1}{10}\beta$, our assumption is that $\mu_G(U \cup U_j) \geq \frac{3}{4}\beta$ and it contradicts U_j being the small part in the partition.

Let $h = \sqrt{2(1 - \frac{99}{100})}$, notice that $h \leq \frac{1}{20}$. For each i, the cut $(U_i, A_i \setminus U_i)$ is sparse in G_i, $\mu_{G_i}(E(U_i, A_i \setminus U_i)) \leq h\mu_{G_i}(U_i)$. Since G_i is an induced graph in G with the same edge weights, we can translate this inequality using Claim A.2 to

$$\mu_G(E(U_i, A_i \setminus U_i)) \leq h\mu_G(E(U_i, A_i)). \tag{6}$$

To show that U is small, we bound the measure of the cut $(A_0, A_0 \setminus U)$ and use Claim A.1. Notice that $A_{i+1} = A_i \setminus U_i$, and that $U_1, \ldots U_l$ are disjoint.

$$\begin{aligned}
\mu_G(E(U, A_0 \setminus U)) &= \sum_{i<j} \mu_G(E(U_i, A_0 \setminus U)) && (\forall i < j, A_0 \setminus U \subseteq A_i \setminus U_i) \\
&\leq \sum_{i<j} \mu_G(E(U_i, A_i \setminus U_i)) && \text{(by (6))} \\
&\leq \sum_{i<j} h\mu_G(E(U_i, A_i)) && (A_i \subset A_0) \\
&\leq \sum_{i<j} h\mu_G(E(U_i, A_0)) \\
&= h \cdot \mu_G(E(U, A_0)). && (7)
\end{aligned}$$

At this point we have

$$\mu_G(E(U, A_0 \setminus U)) \leq h \cdot \mu_G(E(U, A_0)) \tag{8}$$

The LHS is approximately $\mu_G(U) \cdot \mu_G(A_0 \setminus U) \geq \mu_G(U) \cdot \mu_G(A_0)/4$, and the RHS is approximately $h \cdot \mu_G(U) \cdot \mu_G(A_0)$, giving $1/4 < h < 1/20$, a clear contradiction. To make this approximate explanation precise, let us denote $x = \mu_G(U)$. If $x > \alpha, \delta$ then by the expander mixing lemma (Claim A.1) we have

$$\mu_G(E(U, A_0)) \leq \beta(x + \alpha) + \delta.$$

similarly,

$$\mu_G(E(U, A_0 \setminus U)) \geq (\beta - x - \delta)(x - \alpha).$$

Combining these two inequalities with (8),

$$x(\beta - x) \leq h\beta x + \alpha + \delta \tag{9}$$

Given that $x < \frac{3}{4}\beta$, $h < \frac{1}{20}$ and $\alpha, \delta < \frac{1}{100}\beta^2$, we get that $x = \mu_G(U) < \frac{\beta}{10}$.

According to our assumption $\mu_G(U \cup U_j) \geq \frac{3}{4}\beta$, so the set U_j should satisfy that $\mu_G(U_j) \geq \left(\frac{3}{4} - \frac{1}{10}\right)\beta$. This is a contradiction with the fact that $\mu_{G_j}(U_j) \leq \frac{1}{2}$, as described in the following paragraph.

Translating $\mu_{G_j}(U_j) \leq \frac{1}{2}$ into G results in $\mu_G(E(U_j, A_j)) \leq \frac{1}{2}\mu_G(E(A_j, A_j))$, by Claim A.2. Using the mixing lemma on G_{samp},Claim A.1,

$$(\mu_G(U_j) - \delta)(\mu_G(A_j) - \alpha) \leq \delta + \frac{1}{2}\mu_G(A_j)(\mu_G(A_j) + \alpha).$$

Given that $\frac{\beta}{4} \leq \mu_G(A_j) \leq \beta$ and $\alpha, \delta \leq \frac{\beta^2}{100}$ we get that $\mu_G(U_j) \leq \frac{3}{5}\beta$ which is a contradiction. □

Claim A.2. Let $G = (V, E), W$ be a weighted graph, and let $G' = (V', E'), W'$ be an induced subgraph inheriting the weights of G. Then for every $V'' \subset V', E'' \subset E'$:

$$\mu_{G'}(E'') = \frac{\mu_G(E'')}{\mu_G(E')}, \quad \mu_{G'}(V'') = \frac{\mu_G(E(V'', V'))}{\mu_G(E')}.$$

Proof. By definition (recall Definition 2.1) $\mu_{G'}(E'')$ is the probability to pick a random edge from E'' when picking a random edge in G'. The weights in G' are the same as in G.

$$\mu_{G'}(E'') = \Pr_{e \sim W'}[e \in E''] = \Pr_{e \sim W}[e \in E'' | e \in E'] = \frac{\Pr_{e \sim W}[e \in E'']}{\Pr_{e \sim W}[e \in E']} = \frac{\mu_G(E'')}{\mu_G(E')}.$$

For vertex weights, the weight of every vertex is the sum of adjacent edges. Therefore the weight of a vertex $v \in V''$ in G' is composed only by edges inside G'.

$$\begin{aligned}\mu_{G'}(V'') &= \Pr_{(u,v)\sim W'}[u \in V''] \\ &= \Pr_{(u,v)\sim W}[u \in V''|(u,v) \in E'] \\ &= \Pr_{(u,v)\sim W}[(u,v) \in E(V'',V')|(u,v) \in E'] \\ &= \frac{\mu_G(E(V'',V'))}{\mu_G(E')}.\end{aligned}$$

□

B Unique games over expanders

B.1 A list-decoding variant of the unique game algorithm over exapnder graphs

The unique games algorithm takes a solvable unique games instance and outputs a single solution. In our proof we list decode, so we need a list of all possible solutions.

Lemma 8. Let $G = (V, E), W$ be a weighted undirected graph with weights $\{w_e\}_{e\in E}$ and suppose $\lambda(G) \leq \frac{99}{100}$. Further let $\{\pi_e\}_{e\in E}$ be unique constraints over ℓ labels.

Then, there exits an absolute constant c and a polynomial time algorithm that outputs a list $L = \{a^{(1)}, \ldots, a^{(t)}\}$ with $a^{(i)} : V \to [\ell]$ such that for every assignment $a : V \to [\ell]$ that satisfies $1 - \eta$ of the constraints, there exists $a^{(i)}$ which satisfies $\Pr_{v\sim W}[a(v) = a^{(i)}(v)] \geq 1 - \eta c^{\ell}$.

The algorithm receives a weighted constraint graph $G = (V, E), W = \{w_e\}_{e\in E}, \{\pi_e\}_{e\in E}$ and returns a list of satisfying assignments $L = \{a^{(1)}, \ldots a^{(t)}\}, a^{(i)} : V \to [\ell]$. In the algorithm we repeatedly run the unique games algorithm from [MM10], remove each solution and rerun the algorithm again.

Initialization : Set $i = 1$, and set $\pi_e^{(1)} = \pi_e$ for every $e \in E$.

Solving Unique Constraints : Repeat

1. Use the unique games algorithm from [MM10], B.2, on the graph G with constraints $\{\pi^{(i)}\}_{e\in E}$.
2. If the algorithm didn't return a solution, quit the loop.
3. Else, let $a^{(i)} : V \to [\ell - i + 1]$ be the algorithm output.
4. Let $\{\pi^{(i+1)}\}_{e\in E}$ be the constraints after removing $a^{(i)}$ (see details after the algorithm).
5. Set $i = i + 1$ and repeat.

Output : Output $a^{(1)}, \ldots a^{(t)}$.

Removing the assignment a from $\pi : [j] \to [j]$, getting $\pi' : [j-1] \to [j-1]$ is done as follows.

- For every vertex v, reorder the elements such that $a(v) = j$.
- If a satisfies π, i.e. $\pi(j) = j$, then π' is equal to π restricted to $[j-1]$.
- Else, exists $i, l \neq j$ such that $\pi(l) = j, \pi(j) = i$, then set $\pi'(l) = i$, and the rest is identical to π.

In an expander graph, if a, a' are two assignments satisfying almost all of the constraints, then they must be either almost identical or completely different.

Claim B.1. Let $a, a' : V \to [\ell]$ be two assignments satisfying $1-\eta$ and $1-\eta'$ of the constraints in G, respectively. Then, either $\Pr_{v\in V}[a(v) = a'(v)] \geq 1 - 50(\eta' + \eta)$ or $\Pr_{v\in V}[a(v) = a'(v)] \leq 50(\eta' + \eta)$.

Proof. Let $D \subset E$ the set of disagreeing vertices,

$$D = \left\{ v \in V \;\middle|\; a(v) \neq a'(v) \right\}.$$

The constraints in G are unique, for every $(v_1, v_2) \in E$, if both a, a' satisfies the edge, and $a(v_1) = a'(v_1)$, then it must be that $a(v_2) = a'(v_2) = \pi_{(1,2)}(a(v_1))$.

Therefore, if an edge (a_1, a_2) has $a_1 \notin D, a_2 \in D$, it is not possible that both a, a' satisfy it, therefore

$$\mu(E(D, V \setminus D)) \leq \eta + \eta'.$$

The second largest eigenvalue of G is at most $\frac{99}{100}$, so its edge expansion is at least $\frac{1}{50}$, by Cheeger inequality.

$$\mu(E(D, V \setminus D)) \geq \frac{1}{50} \min\{\mu(D), \mu(V \setminus D)\}.$$

This implies that $\min\{\mu(D), \mu(V \setminus D)\} \leq 50(\eta + \eta')$, which finishes the proof. □

Proof of Lemma 8. Fix $a : V \to [\ell]$ to be an assignment that satisfies $1-\eta$ of the constraints of G. If $\eta \geq c^\ell$ for c the global constant, there is nothing to prove, therefore we assume that $\eta < c^\ell$ and show that there must be $\pi^{(i)} \in L$ such that $a, a^{(i)}$ are close.

For every $i \in [t]$, recall $\{\pi_e^{(i)}\}_{e\in E}$ are the constraints used in the ith step of the algorithm runtime. Let η_i be the constraints in the ith round satisfied by a:

$$\eta_i = \Pr_{(u,v)\sim W}[a(u) \neq \pi_{u,v}^{(i)}(a(v))].$$

In the following claim we show that if $a^{(i)}$ is very different than a, then after removing $a^{(i)}$ the assignment a still satisfies a large fraction of the constraints.

Claim B.2. There exist an absolute constant $b > 1$ such that if $\eta < b^{-\ell}$ and $\Pr_u[a(u) = a^{(i)}(u)] \leq \frac{1}{2}$ for all $i \leq j$, then $\eta_{j+1} \leq \eta \cdot b^{j+1}$.

Proof. We prove the claim by induction on i.

According to our assumption, a satisfies $1-\eta$ of the constraints $\{\pi_e^{(1)}\}_{e\in E^*}$.

Assume that the claim holds for $j-1$, i.e. $\eta_j < \eta b^j$, so the unique games instant $\pi^{(j)}$ has a solution satisfying $1-\eta_j$ of the constraints. By Theorem B.4, the unique games algorithm outputs $a^{(j)} : V \to [\ell - j + 1]$ which satisfies $1 - 50C\eta_j$ of the constraints.

If $\Pr_u[a(u) = a^{(j)}(u)] \leq \frac{1}{2}$ then according to Claim B.1.

$$\Pr_u[a(u) = a^{(j)}(u)] \leq 50(\eta_j + 50C\eta_j)$$

Removing the label $a^{(j)}$ from all vertices can increase the edges unsatisfied by a by at most $50(\eta_j + 50C\eta_j)$. This is because removing $a^{(j)}$ "ruins" a constraint $\pi_{u,v}$ for a only if $a, a^{(j)}$ are equal on either u or v.

$$\begin{aligned}
\eta_{j+1} &= \Pr_{(u,v)\in E}[a(u) \neq \pi_{u,v}^{(j+1)}(a(v))] \\
&\leq \Pr_{(u,v)\in E}[a(u) \neq \pi_{u,v}^{(j)}(a(v))] + \Pr_{(u,v)\in E}[a(u) = a^{(j)}(u) \vee a(v) = a^{(i)}(v)] \\
&\leq \eta_j + 50(\eta_j + 50C\eta_j).
\end{aligned}$$

choosing the appropriate constant $b = \max\{3000C, \frac{1}{100c'}\}$ we finishes the proof. □

Fix the constant c to be $c = 2b$, and assume towards contradiction that a if far from every $a^{(i)}$ in the output list, i.e. $\Pr_u[a(u) = a^{(i)}(u)] \leq \frac{1}{2}$ for every $i \in [t]$. There are two cases:

- $t < \ell$: the unique games algorithm failed to output a solution for $\{\pi_e^{(t+1)}\}_{e \in E}$. From the above claim $\eta_{t+1} \leq \eta b^t$, so a satisfies $1 - \eta b^t \geq 1 - \frac{1}{b} \geq 1 - 100c'$ of the constraints $\{\pi_e^{(t+1)}\}_{e \in E}$ and the unique games algorithm should have output a solution.
- $t = \ell$: in this case the assignments $a^{(1)}, \ldots a^{(\ell)}$ "cover" all of the possible labels, i.e. for every $u \in V$, there must be some $i \in [\ell]$ such that $a^{(i)}(u) = a(u)$.

  According to the claim above, for every $i \in [\ell]$, $\eta_i \leq \eta b^i$. By *Claim B.1* and our assumption, for every $i \in [\ell]$, $\Pr_u[a(u) = a^{(i)}(u)] \leq 50(\eta_i + 50C\eta_i) \leq b\eta_i$.

  $$\Pr_u[\exists i \textbf{ s.t } a^{(i)}(u) = a(u)] \leq \sum_{i \in [\ell]} b\eta_i \leq \eta \sum_{i \in [\ell]} b^{i+1} \leq 2\eta b^l < 1,$$

  which contradicting $a^{(i)}$ covers all solutions.

Therefor, there must be an assignment that is close to a, let j be the first assignment to satisfy $\Pr_u[a(u) = a^{(j)}(u)] \leq \frac{1}{2}$. Using Theorem B.4 and Claim B.2,

$$\Pr_u[a(u) \neq a^{(j)}(u)] \leq 50(\eta_j + 50C\eta_j)$$

which finishes the proof. □

B.2 A unique game algorithm over weighted graphs

Our starting point is the following theorem.

Theorem B.3 (Theorem 10, [MM10])**.** There exists a polynomial time approximation algorithm that given a $1 - \delta$ satisfiable instance of Unique Games on a d-regular expander graph G with $\frac{\delta}{\lambda_G} \leq c$, the algorithm finds a solution of value

$$1 - C\frac{\delta}{h_G},$$

where c and C are some positive absolute constants, λ_G is the laplacian second smallest eigenvalue and h_G is the edge expansion.

In fact, the same theorem with slightly modified proof works also for weighted non-regular graphs.

Theorem B.4 (Weighted unique games)**.** There exists a polynomial time approximation algorithm that given a $1 - \delta$ satisfiable instance of Unique Games on a weighted expander graph G such that $\frac{\delta}{\lambda_G} \leq c$, the algorithm finds a solution of value

$$1 - C\frac{\delta}{h_G},$$

where c and C are some positive absolute constants, λ_G is the laplacian second smallest eigenvalue and h_G is the edge expansion.

We reprove the theorem in the case of weighted graphs, skipping the parts of the proof which are unchanged.

We start from the SDP relaxation of the unique games. The only difference from the regular case is that the target function is weighted. For each vertex $u \in V$ and label $i \in [\ell]$ we define a vector u_i of length t.

Definition B.5 (SDP relaxation)**.** Minimize:

$$\frac{1}{\omega}\sum_{(u,v)\in E} w_{u,v}\sum_{i\in[\ell]}\|u_i - v_{\pi_{u,v}(i)}\|^2$$

Subject to

$$\forall u\in V, i\neq j\in[\ell], \quad \langle u_i, u_j\rangle = 0 \tag{10}$$

$$\forall u\in V, \quad \sum_{i\in[\ell]}\|u_i\|^2 = 1 \tag{11}$$

$$\forall u,v,x\in V, i,j,l\in[\ell] \quad \|u_i - x_l\|^2 \leq \|u_i - v_j\|^2 + \|v_j - x_l\|^2 \tag{12}$$

$$\forall u,v\in V, i,j\in[\ell] \quad \|u_i - v_j\|^2 \leq \|u_i\|^2 + \|v_j\|^2 \tag{13}$$

$$\forall u,v\in V, i,j\in[\ell] \quad \|u_i\|^2 \leq \|u_i - v_j\|^2 + \|v_j\|^2 \tag{14}$$

Where $\omega = \sum_u w_u$.

An integral solution sets for each $u\in V$ a label $i\in[\ell]$. It translates into vectors by $u_i = \mathbf{1}\frac{1}{\sqrt{t}}$, and for each $j\neq i, u_j = \mathbf{0}$, where $\mathbf{1},\mathbf{0}$ are the all 1 and all 0 vectors, respectively. Each integral solution satisfies all the constraints, so if the UG instance is $1-\delta$ satisfiable then the SDP value is at least $1-\delta$.

Similarly to [AKK+08, MM10], we define the earthmover distance.

Definition B.6. For every two sets of orthogonal vectors $\{u_i\}_{i\in[\ell]}, \{v_i\}_{i\in[\ell]}$ let

$$\Delta(\{u_i\}_{i\in[\ell]}, \{v_i\}_{i\in[\ell]}) = \min_{\tau\in\mathcal{S}_\ell}\left\{\sum_{i\in[\ell]}\|u_i - v_{\tau(i)}\|^2\right\},$$

where $\mathcal{S}_\ell$ are all permutation over ℓ elements.

For an SDP solution $\{u_i\}_{u\in V, i\in[\ell]}$, we denote by $\Delta(u,v)$ the earthmover distance between the vectors of u and the vectors of v.

Arora at el. [AKK+08] showed that the SDP solution has small average earthmover distance, i.e. that for the SDP solution $\{u_i\}_{u\in V, i\in[\ell]}$, the expression $\mathbb{E}_{u,v\in V}[\Delta(\{u_i\}_{i\in[\ell]}, \{v_i\}_{i\in[\ell]})]$ is small. Their proof has a lemma and a corollary, the lemma is for any sets of vectors and is not related to the graph. The corollary uses the graph regularity but can be easily modified for weighted graphs as well.

Lemma 9 (Lemma 2.2 in [AKK+08])**.** For every positive even integer q and every SDP solution $\{u_i\}_{u\in V, i\in[\ell]}$, there exists a set of vectors $\{\boldsymbol{V}_u\}_{u\in V}$ that for every pair $u,v\in V$,

$$\frac{1}{q}\|\boldsymbol{V}_u - \boldsymbol{V}_v\|^2 \leq \frac{1}{\ell}\Delta(u,v) \leq 2\|\boldsymbol{V}_u - \boldsymbol{V}_v\|^2 + O\left(2^{-\frac{q}{2}}\right).$$

Corollary B.7. For every constant $R\in(0,1)$, there exists a positive $c>0$ such that for any $1-\delta$ satisfiable instance of unique games on G, if $\frac{\delta}{\lambda_G} < c$, then

$$\mathop{\mathbb{E}}_{u,v\in V}[\Delta(u,v)] \leq R,$$

For u,v distributed according to their weight, and $\lambda_G = 1-\lambda(G)$ the second smallest eigenvalue of the noamalized laplacian.

We prove the corollary also for the case of weighted graphs. The proof is almost identical to the proof in [AKK+08], with slight modifications for the weights.

Proof. By Claim B.11, the second smallest eigenvalue of the laplacian of G can also be represented by

$$\lambda_G = \min_{\{z_u\}_{u\in V}} \frac{\mathbb{E}_{(u,v)\in E}[\|z_u - z_v\|^2]}{\mathbb{E}_{u,v\in V}[\|z_u - z_v\|^2]}, \tag{15}$$

where $\{z_u\}_{u\in V}$ is a set of vectors, one for every vertex, and the expectation is done according to the edge and vertex weights in G.

$$\begin{aligned}
\mathop{\mathbb{E}}_{u,v\in V}[\Delta(u,v)] \leq & 2\ell \mathop{\mathbb{E}}_{u,v\in V}\left[\|\boldsymbol{V}_u - \boldsymbol{V}_v\|^2\right] + \ell O\left(2^{-\frac{q}{2}}\right) && \text{(by Lemma 9)}\\
\leq & \frac{2\ell}{\lambda_G} \mathop{\mathbb{E}}_{(u,v)\in E}\left[\|\boldsymbol{V}_u - \boldsymbol{V}_v\|^2\right] + \ell O\left(2^{-\frac{q}{2}}\right) && \text{(by (15))}\\
\leq & \frac{2q\ell}{\lambda_G} \mathop{\mathbb{E}}_{(u,v)\in E}[\Delta(u,v)] + \ell O\left(2^{-\frac{q}{2}}\right) && \text{(by Lemma 9)}\\
\leq & \frac{2q\ell}{\lambda_G}\epsilon + \ell O\left(2^{-\frac{q}{2}}\right). && \text{(by the SDP solution)}
\end{aligned}$$

Taking large enough q such that $\ell O\left(2^{-\frac{q}{2}}\right) < \frac{R}{2}$ and $c < \frac{R}{4q\ell}$ (R, ℓ, q are all constants), we finish the proof. □

We use the rounding algorithm of [MM10], with the only difference that the initial vertex u is picked according to its weight and not uniformly.

The proof of the rounding algorithm uses the following claim from [MM10]. This claim is general about vector normlization, and is independent of any graph.

Lemma 10 (Lemma 1 from [MM10], actually proven in previous paper.)**.** For every SDP solution $\{u_i\}_{u\in V, i\in[\ell]}$, there exists a set of vectors $\{\tilde{u}_i\}_{u\in V, i\in[\ell]}$ satisfying the following properties:

1. Triangle inequalities: for every $u, v, w \in V$ and labels $i, j, l \in [\ell]$:
$$\|\tilde{u}_i - \tilde{v}_j\| + \|\tilde{v}_j - \tilde{w}_l\| \leq \|\tilde{u}_i - \tilde{w}_l\|.$$
2. For every $u, v \in V, i.j \in [\ell]$,
$$\langle \tilde{u}_i, \tilde{v}_j \rangle = \frac{\langle u_i, v_j \rangle}{\max\{\|u_i\|^2, \|v_j\|^2\}}.$$
3. For all non zero vectors u_i, $\|\tilde{u}_i\| = 1$.
4. For every $u \in V, i \neq j \in [\ell]$, $\langle \tilde{u}_i, \tilde{u}_j \rangle = 0$.
5. For every $u, v \in V, i, j \in [\ell]$,
$$\|\tilde{v}_j - \tilde{u}_i\| \leq \frac{2\|v_j - u_i\|}{\max\{\|u_i\|^2, \|v_j\|^2\}}.$$

The set of vectors $\{\tilde{u}_i\}_{u\in V, i\in[\ell]}$ can be obtained in polynomial time.

The rounding algorithm: Approximation algorithm from [MM10], with slight modification. The input is an SDP solution $\{u_i\}_{u\in V, i\in[\ell]}$ of cost ϵ.

Initialization :

1. Pick a random vertex $u \in V$ according to the vertex weights w_u.

2. Pick a random label $i \in [\ell]$, each with probability $\|u_i\|^2$.
3. Pick a random number $t \in [0, \|u_i\|^2]$.
4. Pick a random $r \in [R, 2R]$.
5. Obtain vectors $\{\tilde{u}_i\}_{u \in V, i \in [\ell]}$.

Labels Assignment : For every $v \in V$:

1. Let $S_v = \{p \in [\ell] \mid \ \|v_p\|^2 \geq t, \|\tilde{u}_i - \tilde{v}_p\|^2 \leq r\}$.
2. If $S_v = \{p\}$, assign the label p to v. Else, assign arbitrary one.

The algorithm analysis uses the following definition, it is well defined because the triangle inequality and orthonormality of the none 0 vectors.

Definition B.8. Let $\tau_{x,v}$ the partial mapping from $[\ell]$ to $[\ell]$ which maps p into q if $\|\tilde{v}_p - \tilde{x}_q\| \leq 4R$.

Definition B.9. Let $X = \{x \in V \mid \ |S_x| = 1\}$.

The following lemmas have proofs which do not assume regularity, so they can be used as is.

1. If $p \in S_v$ and $q \in S_x$ with non zero probability, for the same initial vertex and label, then $\tau_{v,x}(p) = q$.
2. $|S_v| \leq 1$.
3. If $S_v = \{p\}$, then $S_x = \{\tau_{x,v}(p)\}$ or $S_w = \emptyset$.
4. For every choice of initial vertex u, every $v \in V$, $p \in [\ell]$, $\Pr_{t,r}[S_v = \{p\}] \leq \|v_p\|^2$.

The following lemmas has to be slightly modified, mainly inserting $\mu(X)$ instead of $|X|$.

Lemma 11 (Lemma 5 in [MM10]). If $\frac{\epsilon}{\lambda_G} < c$, then $\mathbb{E}[\mu(X)] \geq \frac{1}{4}$.

The proof is identical except the difference in the distribution over the vertices, which is weighted instead of uniform.

Proof. Suppose u is the initial vertex, then for every $v \in V$ we express the probability of $v \in X$ using $\Delta(u, v)$.

For every label $p \in [\ell]$, if $\exists q \in [\ell]$ such that $\|u_q - v_p\|^2 \leq \frac{R}{2}\|v_p\|^2$, then

$$\|\tilde{v}_p - \tilde{u}_q\| \leq \frac{2\|u_q - v_p\|}{\max\{\|u_q\|^2, \|v_p\|^2\}} \leq R \leq r,$$

which implies $\tau_{u,v}(p) = q$. In this case, if q is the initial label and $t \leq \|v_p\|^2$, then $S_v = \{p\}$ which implies $v \in X$. Therefore,

$$\Pr_{i,t}[S_v = \{p\}] \geq \Pr_{i,t}[i = q \wedge t \leq \|v_p\|^2] = \|u_q\|^2 \min\{1, \frac{\|v_p\|^2}{\|u_q\|^2}\} = \min\{\|u_q\|^2, \|v_p\|^2\} \geq \frac{1}{2}\|v_p\|^2.$$

Where the last inequality is by the triangle inequality, using the fact that $\|u_q - v_p\|^2 \leq \frac{R}{2}\|v_p\|^2$.

Going over all possible labels p for v:

$$\begin{aligned}
\Pr[v \in X] &= \Pr_{i,t,r}[\exists p \textbf{ s.t } S_v = \{p\}] \\
&= \sum_p \Pr_{i,t,r}[S_v = \{p\}] \\
&\geq \sum_{p \textbf{ s.t } \exists q, \|u_q - v_p\|^2 \leq \frac{R}{2}\|v_p\|^2} \frac{1}{2}\|v_p\|^2 \\
&\geq \sum_p \frac{1}{2}\|v_p\|^2 - \sum_{p \textbf{ s.t } \forall q, \|u_q - v_p\|^2 > \frac{R}{2}\|v_p\|^2} \frac{1}{2}\|v_p\|^2 \\
&\geq \frac{1}{2} - \frac{1}{2}\sum_p \frac{2}{R}\min_q\{\|v_p - u_q\|^2\} \qquad \text{(since for all } q, \|v_p\|^2 < \tfrac{2}{R}\|v_p - u_q\|^2) \\
&= \frac{1}{2} - \frac{1}{R}\Delta(u,v).
\end{aligned}$$

The part so far is identical to the original proof.

By the earthmover distance lemma, $\mathbb{E}_{u,v\in V}[\Delta(u,v)] < R$, so

$$\begin{aligned}
\mathbb{E}[\mu(X)] &= \sum_{u,v\in V} \mu(u)\mu(v)\Pr[v \in X | u \text{ initial vertex}] \\
&\geq \sum_{u,v\in V} \mu(u)\mu(v)\left(\frac{1}{2} - \frac{1}{R}\Delta(u,v)\right) \\
&\geq \frac{1}{2} - \frac{1}{R}\mathop{\mathbb{E}}_{u,v\in V}[\Delta(u,v)] \\
&\geq \frac{1}{2} - \frac{1}{4}.
\end{aligned}$$

Where the distribution over picking a vertex is according to its weight. □

Corollary B.10.

$$\Pr[\mu(X) \geq \frac{1}{8}] \geq \frac{1}{8}.$$

The following lemma appears in the original proof with uniform distribution over the edges, we reprove it for edges picked according to their weight. The proof is practically the same, only choosing edges and vertices according to their weights.

Lemma 12.

$$\mathbb{E}[\mu(X, V \setminus X)] \leq \frac{6\epsilon}{R}.$$

Proof. The first part of the proof is identical to the original proof. Fix $u \in V$ the initial vertex, we bound the probability of $v \in X, x \notin X$ by $\frac{6}{R}\sum_p \|v_p - x_{\pi_{x,v}(p)}\|^2$.

If $v \in X, x \notin X$, then $S_v = \{p\}, S_x = \emptyset$. Let $q = \pi_{x,v}(p)$. Since $S_v = \{p\}$, then $\|v_p\|^2 \geq t, \|\tilde{u}_i - \tilde{v}_p\|^2 \leq r, i = \tau_{u,v}(p)$. One of the two cases must happen

1. $\|x_q\|^2 < t$.
2. $\|x_q\|^2 \geq t, \|\tilde{x}_q - \tilde{u}_i\|^2 > r$.

We sum over all p the probability that these events occur (each p has a $q = \pi_{x,v}(p)$).

$$\begin{aligned}\Pr_{i,t,r}[1] \leq& \sum_p \Pr[i = \sigma_{v,u}(p)] \Pr[\|x_q\|^2 < t \leq \|v_p\|^2 | i = \sigma_{v,u}(p)] \\ \leq& \sum_p \|u_{\sigma_{v,u}(p)}\|^2 \frac{\|v_p\|^2 - \|x_q\|^2}{\|u_{\sigma_{v,u}(p)}\|^2} \\ \leq& \sum_p \left(\|v_p\|^2 - \|x_q\|^2\right).\end{aligned}$$

$$\begin{aligned}\Pr_{i,t,r}[2] =& \sum_p \Pr[i = \sigma_{v,u}(p)] \Pr[t \leq \|v_p\|^2] \Pr[\|\tilde{u}_i - \tilde{v}_p\|^2 \leq r < \|\tilde{x}_q - \tilde{u}_i\|^2 | i = \sigma_{v,u}(p)] \\ \leq& \sum_p \|u_{\sigma_{v,u}(p)}\|^2 \frac{\|v_p\|^2}{\|u_{\sigma_{v,u}(p)}\|^2} \frac{\|\tilde{x}_q - \tilde{u}_i\|^2 - \|\tilde{u}_i - \tilde{v}_p\|^2}{R} && \text{(triangle inequality)} \\ \leq& \sum_p \|v_p\|^2 \frac{1}{R} \|\tilde{v}_p - \tilde{x}_q\|^2 \\ \leq& \sum_p \|v_p\|^2 \frac{1}{R} \frac{2\|v_p - x_q\|^2}{\max\{\|v_p\|^2, \|x_q\|^2\}} \\ \leq& \sum_p \frac{2}{R} \|v_p - x_q\|^2.\end{aligned}$$

Therefore, for every edge (v, x) $\Pr[(v, x) \in E(X, V \setminus X)] \leq \sum_p (1 + \frac{2}{R}) \|v_p - x_q\|^2$.

The expected value of the cut:

$$\begin{aligned}\mathbb{E}[\mu(E(X, V \setminus X))] =& \frac{2}{\omega} \sum_{(v,x) \in E} w_{v,w} \Pr[(v, x) \in E(X, V \setminus X)] \\ \leq& \frac{2}{\omega} \sum_{(v,x) \in E} w_{v,w} \frac{3}{R} \|v_p - x_q\|^2 && \text{(SDP value} \leq \delta) \\ \leq& \frac{6\delta}{R}.\end{aligned}$$

□

Lemma 13 (lemma 8)**.** If $\delta < \min\{c_R \lambda_G, \frac{h_G R}{1000}\}$ then with probability at least $\frac{1}{16}$, $\mu(X) \geq 1 - \frac{100\delta}{h_G R}$.

Proof. By the definition of h_G, $\mu(E(X, V \setminus X)) \geq h_G \min\{\mu(X), \mu(V \setminus X)\}$, which implies

$$\begin{aligned}\frac{6\delta}{R} \geq& \mathbb{E}[\mu(E(X, V \setminus X))] \\ \geq& h_G \mathbb{E}[\min\{\mu(X), \mu(V \setminus X)\}].\end{aligned}$$

We get that $\mathbb{E}[\min\{\mu(X), \mu(V \setminus X)\}] \leq \frac{6\delta}{h_G R}$, by Markov $\Pr[\min\{\mu(X), \mu(V \setminus X)\} \leq \frac{100\delta}{h_G R}] \geq 1 - \frac{1}{16}$
We also know that $\Pr[\mu(X) \geq \frac{1}{8}] \geq \frac{1}{16}$, so with probability at most $1/16$,$\mu(V \setminus X) \leq \frac{100\delta}{h_G R}$. □

The following lemma is independent of the graph, so its proof is unmodified.

Lemma 14. For every edge $(v, x) \in E$,

$$\Pr[v, x \in X, (v, x) \text{isn't satisfied}] \leq 4\delta_{v,x},$$

for $\delta_{v,x} = \frac{1}{2} \sum_{i \in [\ell]} \|v_i - x_{\pi_{x,v}(i)}\|^2$.

And we are ready to prove the theorem, again the proof is almost identical.

proof of Theorem B.4. We show that the randomized algorithm described above solves the UG instance with constant probability. It can then easily be derandomized.

The algorithm solves the SDP, then runs the rounding algorithm. If $\mu(X) \geq 1 - \frac{100\delta}{h_G R}$, it outputs the labeling, else it fails.

Suppose the algorithm doesn't outputs fail, then by the definition of weights,

$$\mu(E(X,X)) \geq 1 - \frac{100\delta}{h_G R},$$

as $\mu(V \setminus X) \leq \frac{100\delta}{h_G R}$).

The expected violated constraints inside X is at most,

$$\frac{2}{\omega} \sum_{(v,x)\in E} w_{v,x} 4\delta_{v,x} \leq \frac{2}{\omega} \sum_{(v,x)\in E} w_{v,x} 4 \sum_{p\in[\ell]} \|v_p - x_{\pi_{x,v}(p)}\|^2 \leq 64\delta.$$

Therefore with constant probability we get a $1 - 64\delta - \frac{100\delta}{h_G R}$ solution. □

B.2.1 Eigenvalue Proof

Claim B.11. Let $G = (V, E)$ be a weighted graph with weights $\{w_{u,v}\}_{(u,v)\in E}$, and let $\mathcal{L}$ be the normalized laplacian matrix of G,

$$\mathcal{L}_{v,u} = \begin{cases} 1 & \text{if } u = v \\ -\frac{w_{u,v}}{\sqrt{w_u w_v}} & \text{if } (u,e) \in E \\ 0 & \text{else} \end{cases},$$

where $w_u = \sum_{v \text{ s.t } (u,v)\in E} w_{u,v}$. The second smallest eigenvalue of the laplacian corresponds to

$$\lambda_2 = \min_{\{z_u\}_{u\in V}} \frac{\mathbb{E}_{(u,v)\sim w}[\|z_u - z_v\|^2]}{\mathbb{E}_{u,v\sim V}[\|z_u - z_v\|^2]}.$$

Where $\{z_u\}_{u\in V}$ is a set of vectors, $\forall u, z_u \in \mathbb{R}^t$.

Proof. We define a new matrix $\mathcal{L}' \in \mathbb{R}^{|V|t\times|V|t}$, which is composed of $t \times t$ scalar matrix blocks, i.e. for every $u, v \in V$, the matrix $\mathcal{L}'_{u,v}$ is a $t \times t$ scalar matrix, $\mathcal{L}'_{u,v} = I^{t\times t}\mathcal{L}_{u,v}$. Formally, we denote each row and column by two indices $u \in V, i \in [t]$ and

$$\mathcal{L}'_{(u,i),(v,j)} = \begin{cases} \mathcal{L}_{u,v} & \text{if i=j} \\ 0 & \text{else} \end{cases}.$$

$\mathcal{L}$ has a single eigen value 0, the new matrix $\mathcal{L}'$ has t eigenvalues 0. One eigenvectors basis for the nullspace is $y^1, \ldots y^t \in \mathbb{R}^{|V|t}$, $y^l_{u,j} = \begin{cases} y_u & l = j \\ 0 & \text{else} \end{cases}$, for y the eigenvector of $\mathcal{L}$.

The spectrum of $\mathcal{L}'$ is identical to the spectrum of $\mathcal{L}$, only each eigenvalue repeats t times. Therefore the second largest eigenvalue of $\mathcal{L}$ is equal to the $t+1$ eigenvalue of $\mathcal{L}'$, and is equal

$$\lambda_2 = \min_{x\in\mathbb{R}^{t|V|}} \left\{ \frac{\langle x, \mathcal{L}'x\rangle}{\langle x, x - y^1\langle x, y^1\rangle - \cdots - y^t\langle x, y^t\rangle\rangle} \right\}. \tag{16}$$

The numerator explicitly:

$$\begin{aligned}\langle x, \mathcal{L}'x\rangle &= \sum_{u,v\in V, i,j\in[t]} x_{u,i}\mathcal{L}'_{(u,i),(v,j)}x_{v,j}\\ &= \sum_{u,v\in V, i\in[t]} x_{u,i}\mathcal{L}_{u,v}x_{v,i}\\ &= \sum_{u\in V, i\in[t]} x_{u,i}^2 - 2\sum_{(u,v)\in E, i\in[t]} \frac{w_{u,v}}{\sqrt{w_u w_v}}x_{u,i}x_{v,i}\\ &= \sum_{u\in V}\|x_u\|^2 - 2\sum_{(u,v)\in E}\frac{w_{u,v}}{\sqrt{w_u w_v}}\langle x_u, x_v\rangle.\end{aligned}$$

Where x_u is the length t vector containing $x_{u,i}$ for $i \in [t]$.

The denominator:

$$\begin{aligned}\langle x, x - y^1\langle x, y^1\rangle - \cdots - y^t\langle x, y^t\rangle\rangle =& \langle x, x\rangle - \left(\langle x, y^1\rangle\right)^2 - \cdots \left(\langle x, y^t\rangle\right)^2\\ =& \sum_{u\in V, i\in[t]} x_{u,i}^2 - \sum_{l\in[t]}\left(\langle x, y^l\rangle\right)^2\\ =& \sum_{u\in V, i\in[t]} x_{u,i}^2 - \sum_{l\in[t]}\sum_{u,v\in V, i,j\in t} x_{u,i}y^l_{u,i}x_{v,j}y^l_{v,j}\\ =& \sum_{u\in V, i\in[t]} x_{u,i}^2 - \sum_{l\in[t]}\sum_{u,v\in V} x_{u,l}y_u x_{v,l}y_v\\ =& \sum_{u\in V}\|x_u\|^2 - \sum_{u,v\in V} y_u y_v\langle x_u, x_v\rangle\\ =& \sum_{u\in V}\|x_u\|^2 - \sum_{u,v\in V}\frac{\sqrt{w_u w_v}}{\omega}\langle x_u, x_v\rangle.\end{aligned}$$

We write the expectations explicitly:

$$\begin{aligned}\mathop{\mathbb{E}}_{(u,v)\sim w}[\|z_u - z_v\|^2] =& \frac{2}{\omega}\sum_{(u,v)\in E} w_{u,v}\langle z_u - z_v, z_u - z_v\rangle\\ =& \frac{2}{\omega}\sum_{(u,v)\in E} w_{u,v}(\|z_u\|^2 + \|z_v\|^2 - 2\langle z_u, z_v\rangle)\\ =& \frac{2}{\omega}\sum_{u\in V} w_u\|z_u\|^2 - \frac{4}{\omega}\sum_{(u,v)\in E} w_{u,v}\langle z_u, z_v\rangle.\end{aligned}$$

$$\begin{aligned}\mathop{\mathbb{E}}_{u,v\sim V}[[\|z_u - z_v\|^2] =& \frac{1}{\omega^2}\sum_{u,v\in V} w_u w_v\langle z_u - z_v, z_u - z_v\rangle\\ =& \frac{1}{\omega^2}\sum_{u,v\in V} w_u w_v(\|z_u\|^2 + \|z_v\|^2 - 2\langle z_u, z_v\rangle)\\ =& \frac{1}{\omega^2}\sum_{u\in V} 2\omega w_u\|z_u\|^2 - \frac{2}{\omega^2}\sum_{u,v\in V} w_u w_v\langle z_u, z_v\rangle\end{aligned}$$

For every $u \in V, i \in [t]$ let $x_{u,i} = \sqrt{w_u}z_{u,i}$,

$$\langle x, \mathcal{L}'x\rangle = \frac{\omega}{2}\mathop{\mathbb{E}}_{(u,v)\sim w}[\|z_u - z_v\|^2],$$

$$\langle x, x - y^1\langle x, y^1\rangle - \cdots - y^t\langle x, y^t\rangle\rangle = \frac{\omega}{2} \mathop{\mathbb{E}}_{u,v\sim V}[[\|z_u - z_v\|^2].$$

The factor of $\frac{\omega}{2}$ cancels out, and the minimum value is not affected by the multiplication in $\sqrt{w_u}$, as it is taken over all vectors in $\mathbb{R}$. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il