Moden Spectral Graph Theory **Winter 2022**

Lecture 6: Locally Testable Codes

Lecturer: Shayan Oveis Gharan *??*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

Definition 6.1 (Error Correcting Codes)**.** *An error correcting code, C, of length n over an alphabet Σ is a subset of Σ^n. If Σ is a field and C is a subspace of Σ^n, then C is called a linear code. Unless otherwise specified, in these lectures we study linear binary error correcting cod, namely $\Sigma = \mathbb{F}_2$ and Σ subspace of $\mathbb{F}_2^n$. The elements of C are called the* codewords*.*

Definition 6.2 (Generator of a Code)**.** *Let $C \subseteq \mathbb{F}_2^n$ be a linear code of dimension k. A matrix $G \in \mathbb{F}^{n\times k}$ is said to be a* generator matrix *for C if its k columns span C. Note that the generator matrix G provides a way to encode a message $x \in \mathbb{F}_2^k$ as the code word $Gx \in C$*

Definition 6.3 (Parity Check Matrix)**.** *For any (binary) linear code $C \subseteq \mathbb{F}_2^n$ of dimension k there is a matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ (of full rank) such that*

$$C = \{c \in \mathbb{F}_2^n : Hc = 0\},$$

i.e., C is the set vectors in the null-space of H. The matrix H is called the parity check matrix.

If G is the generator of C, it can be written in the standard form $G = \begin{bmatrix} I_k \\ P \end{bmatrix}$. Then, $H = \begin{bmatrix} -P^T & | & I_{n-k} \end{bmatrix}$. Here I_k is the k *timesk* identity matrix.

Definition 6.4 (Rate of a Code)**.** *For a code $C \subseteq \Sigma^n$, the rate of C is defined as*

$$\rho_C = \frac{\log |C|}{n \log |\Sigma|}.$$

Thus if C is a binary linear code of dimension k, we have

$$\rho_C = \frac{k}{n}.$$

In words, $n(1-\rho_C)$ is the amount of redundant *information sent in a codeword of C.*

Definition 6.5 (Distance of a Code)**.** *The distance of a code $C \subseteq \Sigma^n$ is*

$$\delta_C = \min_{c_1, c_2 \in C} \frac{H(c_1, c_2)}{n}$$

where $H(c_1, c_2)$ is the Hamming distance *of c_1, c_2, the minimum number of positions one need to change to turn c_1 into c_2. The importance of this definition is that if a codeword is sent over a channel and $< \frac{1}{2} n\delta_C$ positions are transmitted with error, still it is possible to re-cover the original code.*

For a word $f \in \Sigma^n$ we write $\delta(f, C)$ to denote

$$\delta(f, C) = \min_{c \in C} \frac{H(f, c)}{n}$$

the distance of f from the closest word in C.

If C is a binary code, then $H(c_1, c_2) = \|c_1 + c_2\|_1$. So, in particular,

$$\delta_C = \min_{c \in C, c \neq 0} \frac{\|c\|_1}{n}.$$

This is because for any $c_1, c_2 \in C$, we have $c_1 + c_2 \in C$.

6.1 Expander Graphs

Recall that an unweighted graph $G = (V, E)$ is a λ-expander if $\lambda_2(P) \leq \lambda$, where P is the transition probability matrix of the simple random walk on G.

Lemma 6.6. *Suppose that $G = (V, E)$ is a λ-expander, i.e., for any function $f : V \to \mathbb{R}$ such that $\langle f, \mathbf{1} \rangle = 0$, we have*

$$\langle Pf, f \rangle_{\pi_0} \leq \lambda \langle f, f \rangle_{\pi_0},$$

where as usual the inner-products are with respect to the stationary distribution of the walk π_0. For any set $T \subseteq V$ and $f = \mathbf{1}_T$, if $\langle Pf, f \rangle \geq \epsilon \langle f, f \rangle$, then

$$\pi_0(T) = \langle f, f \rangle \geq \epsilon - \lambda.$$

Proof. Let $p = \pi_0(T) = \langle f, f \rangle = \langle f, \mathbf{1} \rangle$. Write $f = p\mathbf{1} + f^\perp$, for $\langle f^\perp, \mathbf{1} \rangle = 0$.

$$p \cdot \epsilon \leq \langle Pf, f \rangle = \langle P(p\mathbf{1} + f^\perp), p\mathbf{1} + f^\perp \rangle = p^2 + \lambda \langle f^\perp, f^\perp \rangle \underset{\langle f, f \rangle = p}{\leq} p^2 + \lambda p.$$

So, $p \geq \epsilon - \lambda$ as desired. □

Lemma 6.7. *Let $G = (V, E)$ be a d-regular λ-expander. Let $T \subseteq V$ be such that the induced graph $G[T]$, has average degree at least δd. Then $|T| \geq (\delta - \lambda) \cdot |V|$, and the number of edges in $G[T]$, $E(T)$, is at least $(\delta - \lambda)\delta \cdot |E|$*

Proof. Let $f = \mathbf{1}_T$.

$$\langle Pf, f \rangle = \mathbb{E}_{u \sim \pi_0} [f(u) Pf(u)] = \mathbb{E}_{\{u,v\} \sim \pi_1} [f(u) f(v)] = \pi_1(E(T)) \geq \frac{\delta |T| d/2}{nd/2} = \frac{\delta |T|}{n} = \delta \langle f, f \rangle.$$

Now, suppose $\mathbb{P}[T] = \langle f, f \rangle = p$. Then, by previous lemma, $\pi_0(T) = p \geq \delta - \lambda$, i.e., $|T| \geq (\delta - \lambda)n$. It thus follows that

$$|E(T)| \geq (\delta d)|T|/2 = \delta(\delta - \lambda)dn/2 = \delta(\delta - \lambda)|E|,$$

as desired □

6.2 Expander Codes

Definition 6.8 (Expander Codes)**.** *Let $G = (V, E)$ be a d-regular graph with n vertices, and let C_0 be a binary linear error correcting code of length d. For every vertex $v \in G$, let $X_v = \{e \in E : e \sim v\}$ be the set of edges that are neighbor of v. Define, a local code*

$$C_v = \{f \in \mathbb{F}_2^{X_v} : f \in C_0\}.$$

Now, define a global expander code *C as follows:*

$$C := \{f \in \mathbb{F}_2^E : f|_{X_v} \in C_v, \forall v \in V\}.$$

Theorem 6.9 ([SS96])**.** *Let G be a d-regular λ-expander with n vertices. Let C_0 be code (of length d) with rate $\rho > 1/2$ and minimum distance δ Then, the expander code C has rate at least $2\rho - 1$ and minimum distance at least $\delta(\delta - \lambda)$.*

Proof. To obtain the rate of the code C it is enough to count the number of linear restrictions imposed by the constraints on the vertices. Since C_0 has rate ρ, each vertex imposes $(1-\rho)d$ many linear constraints, the total number of linear constraints is at most $n(1-\rho)d$. Therefore, the rate of the code is at least

$$\rho_C \geq \frac{nd/2 - n(1-\rho)d}{nd/2} = 1 - 2(1-\rho) = 2\rho - 1.$$

Next, we prove the bound on minimum distance. Let $w \in C$ be a non-zero word. Recall that to bound the minimum distance of C it is enough to lower bound the weight of w. Let $F \subseteq E$ be the set of edges such that $w_e = 1$; so equivalently, it is enough to show that

$$|F| \geq \delta(\delta - \lambda)nd/2.$$

Let T be the set of vertices incident to edges in F. Since w is a valid codeword, and the base code C_0 has distance δ, every vertex in F must be adjacent to at least $\delta|C_0| = \delta d$ edges of F, i.e., the average degree of the induced subgraph $G[T]$ is at least δd. So, by Lemma 6.7, $|F| \geq \delta(\delta - \lambda)nd/2$ as dires. □

We give a few remarks:

- There are explicit construction of λ-expander graphs with $\lambda \approx 1/\sqrt{d}$. Using such a construction and a given base code (which can be chosen by a random/exhaustive search) we get explicit construction of of constant rate and distance linear binary codes of any length.
- Perhaps, the simplest way to choose the base code C_0 is to choose a random $(d-k)d$ parity check matrix. Let $h(x) := x\log_2 x + (1-x)\log_2(1-x)$ be the entropy function. For $0 < \delta < 1/2$, we let $k \approx (1 - h(\delta)$ then we get a code of distance δ and rate $\approx 1 - h(\delta)$ with high probability. In other words, fo sufficiently small chosen distance we can make the rate arbitrarily close to 1.
- Note that the above construction is very local in the following sense: If we use a separate base code, $C_{0,v}$ for every vertex $v \in V$ such that they all have rate at least ρ and distance δ, still the same statement holds.

6.3 Locally Testable Codes

The main object of this part of the course is to study locally testable codes. We say a code C is κ-locally testable with q queries if there is a randomized local tester that reads at most q bits from a given word w and then accepts or rejects w, such that

- For all $w \in C$, $\mathbb{P}[\text{accept}] = 1$.
- For all $w \notin C$, with $\delta(w, C) \geq \Omega(1)$ we have $\mathbb{P}[\text{reject}] \geq \kappa$. The codes that we will construct in fact have a stronger property that for any $w \notin C$,

$$\mathbb{P}[\text{reject}] \geq \kappa \cdot \delta(w, C).$$

Are there any family of locally testable codes? Yes, perhaps the simplest such family comes from the [BLR93] test: Given a function $h : \mathbb{F}_2^n \to \mathbb{F}_2$ we say h is *linear* if $h(x+y) = h(x) + h(y)$ for any $x, y \in \mathbb{F}_2^n$. Now given a function $h : \mathbb{F}_2^n \to \mathbb{F}_2$ we wan to test whether h is linear by running only $O(1)$ many queries.

Blum, Luby, Rubinfeld [BLR93] proposed the following test:

1. Choose $x, y \in \mathbb{F}_2^n$ uniformly at random
2. Accept if $h(x) + h(y) = h(x+y)$ and reject otherwise.

They proved the following theorem:

Theorem 6.10. *If the function h agrees with a linear function at at least $(1-\epsilon)$ fraction of the inputs, then*

$$\mathbb{P}\left[\textit{BLR acceptes}\right] \geq 1 - 6\epsilon.$$

The proof is a simple Fourier Analysis argument that we skip here. But here is the consequence: Consider the following error correcting code of length 2^n: Given a (secret word) $w \in \{0,1\}^n$, we encode w by writing down the image of the function $h_w : \{0,1\}^n \to \{0,1\}$ where for any $x \in \{0,1\}^n$, $h_w(x) = wx$. Then, by the above theorem such a code is locally testable. But, unfortunately, this code has a very poor rate, only $\frac{n}{2^n}$. So the main question that we are trying to address in these lectures is how to construct locally testable codes with constant rate and distance.

Are Expander Codes Locally Testable? Can the above construction of expander codes be locally testable? It turns out that not necessarily. Let us elaborate: Say we construct an expander code C with the parity check matrix H. Notice every row of H essentially ensures a local parity construct for the base code C_0 around a vertex $v \in G$. Now, let h be the first row of H and let H' be the Parity check matrix where the first row is removed. This simply means that we use a slightly different base code for a vertex $v \in G$. It follows from the above proof that the new code C' (with parity matrix H') still has eesentially same rate and the same distance.

Now, we claim that there are codewords $c' \in C'$ that are far from all code words of C. First notice since the parity check matrix of C' has 1 less constraint, the dimension of C' is one more than then dimension of C. So, there are code words $c' \in C' \setminus C$. Fix such a codeword $c' \in C' \setminus C$. For any codeword $c \in C$ we have that $c + c' \in C'$, so $\|c + c'\|_1 \geq \delta(C') \approx \delta_C$. So, $\delta(c', C) \approx \delta_C$. Now, such a word $c' \notin C' \setminus C$, notice that c' satisfies the local constraint at every vertex of G (except the vertex v with the parity constraint on the first row of H). So, if we choose a random vertex it is very likely that c' satisfies every constraint that we see even though it is very far from all codewords in C.

Are Random Codes Locally Testable? Most likely not. The reason is that a local testable a code comes with a local tester the only looks at a few bits. In a random code, if we look at any (constant) q many bits, most likely all possible 2^q arrangements of this bits are already present in out code. So, we cannot know whether the given word is indeed part of our random code or not. In other words, we are looking for constant rate and distance code that is not too local as in the expander codes and too random as in random codes.

6.4 Tensor Codes

Definition 6.11 (Tensor Codes)**.** *Given binary linear codes C_1 and C_2 of length n_1, n_2 and dimension k_1, k_2 respectively. Let G_1, G_2 be the corresponding generator matrices of the codes. Their* tensor code

$C_1 \otimes C_2 \subseteq \mathbb{F}_2^{n_1 \times n_2}$ is a binary linear code whose code words may be viewed as $n_1 \times n_2$ matrices X explicitly given as

$$\{G_1 X G_2^T : X \in \mathbb{F}_2^{k_1 \times k_2}\}$$

Fact 6.12. *A matrix $c \in \mathbb{F}_2^{n_1 \times n_2}$ is a codeword of $C_1 \otimes C_2$ if and only if every row is a codeword of C_2 and every column is a codeword of C_1.*

Proof. To see that notice by definition we can write $c = G_1 X G_2^T$ for some $X \in \mathbb{F}_2^{k_1 \times k_2}$. Now, we can write $c = G_1 Y$, for $Y \in \mathbb{F}_2^{k_1 \times n_2}$ and $Y = X G_2^T$. So, since G_1 is the generator of C_1 every column of c is a word of C_1. The other case can be proven similarly. □

In other words, we can write

$$C_1 \otimes C_2 = \{c \in \mathbb{F}_2^{n_1 \times n_2} : c(i,.) \in C_2, c(.,j) \in C_1, \forall i \in [n_1], j \in [n_2]\}.$$

Exercise 6.13. *We leave it an exercise to prove the converse of the above fact: Namely, if for $c \in \mathbb{F}_2^{n_1 \times n_2}$ every columns is a word of C_1 and every row is a word of C_2 then $c \in C_1 \otimes C_2$.*

Lemma 6.14. *If C_1, C_2 have dimensions k_1, k_2 and minimum distance δ_1, δ_2 respectively, then the tensor code $C_1 \otimes C_2$ has dimension $k_1 \times k_2$ and minimum distance $\delta_1 \delta_2$.*

Proof. First notice that for any $X \neq 0$ and $X \in \mathbb{F}_2^{k_1 \times k_2}$, $G_1 X G_2^T \neq 0$. This simply implies that the dimension statement.

Next, we argue the minimum distance. Fix arbitrary $X \in \mathbb{F}_2^{k_1 \times k_2}$ such that $X \neq 0$ we show that $G_1 X G_2^T$ has at least $\delta_1 \delta_2 n_1 n_2$ non-zero coordinates. Since $X \neq 0$, at least one column of X, say $X_i \neq 0$. Let $Y = G_1 X$, so $Y_i = G X_i$. Since the distance of X_1 is $\delta_1 n_1$, $\|Y_i\|_1 \geq \delta_1 n_1$. Therefore, at least $\delta_1 n_1$ rows of Y and ($\delta_1 n_1$ columns of Y^T) are non-zero.

Now, notice $Y G_2^T = (G_2 Y^T)^T$. Since C_2 has distance $\delta_2 n_2$, for every non-zero column j of Y^T, $\|(G_2 Y)_j\|_1 \geq \delta_2 n_2$. Finally, since at least $\delta_1 n_1$ columns of Y^T are non-zero, $G_2 Y^T$ has at least $\delta_1 \delta_2 n_1 n_1$ non-zeros. □

Testing Tensor Codes. Here is a natural test to see whether $f \in \mathbb{F}_2^{n_1 \times n_2}$ is in $C_1 \otimes C_2$. Randomly choose a row or a column of f, and check whether the restriction of f to $f(.,j) \in C_1$ or $f(i,.) \in C_2$.

For $f \in \mathbb{F}_2^{n_1 \times n_2}$, let

$$\delta^{col}(f, C) = \delta(f, C_1 \otimes \mathbb{F}_2^{n_2}, \delta^{row}(f, C) = \delta(f, \mathbb{F}_2^{n_1} \otimes C_2).$$

In words, $\delta^{col}(f, C)$ is the minimum fraction of entries of f we need to change such that every column becomes a word of C_1 and similarly $\delta^{row}(f, C)$ is the minimum fraction of entries of f to change such that every row is a word of C_2. We say code $C = C_1 \otimes C_2$ is ρ-robustly testable if

$$\beta = \min_{f \notin C_1 \otimes C_2} \frac{\frac{1}{2}(\delta^{col}(f, C) + \delta^{row}(f, C))}{\delta(f, C_1 \otimes C_2)}$$

Note that obviously $\beta \leq 1$. Roughly speaking if C is robust testable, then the fraction of bits we need to fix in f to get a codeword of C, is up to a constant factor the same as the maximum fraction of bits of f we need to change just to make sure the rows are satisfied and the fraction we need to change to make sure column constraints are satisfied.

Local Testability of Tensor Codes. Suppose that C_1, C_2 have length $n_1 = n_2 = n$ and constant rate and distance. Then, the code $C = C_1 \otimes C_2$ has a constant rate and distance and length n^2. Suppose further that it has a constant robust testability, β. Now, suppose for a word $f \in \mathbb{F}_2^{n \times n}$, $\delta(f, C_1 \otimes C_2) = \epsilon$. So, $\frac{1}{2}(\delta^{col}(f,C) + \delta^{row}(f,C)) \geq \beta\epsilon$; so say $\delta^{col}(f,C) \geq \beta\epsilon/2$. So, we get at that least $\frac{\epsilon\beta n}{2}$ many columns of f are not in C_1. So, if we run the the test in the previous paragraph, with probability at least $\frac{\epsilon\beta}{4}$ we reject f. Notice that such a test needs to query $O(\beta\sqrt{\text{len}(C)}\delta(f,C))$ many entries to reject f. So, we don't get the ideal local testability but we are doing much better than expander codes or random codes as we just need to see $\sqrt{\text{len}}$ many coordinates as opposed to linear.

6.5 Robust Testable Tensor Codes

Robust testability of tensor codes was first studied by Ben-Sasson and Sudan [BSS06] where they prove robust testability of 3 tensors. In this section we discuss a result of Dinur, Sudan, Widgerson [DSW06] which gives a construction of robust testable (two) tensor codes.

Definition 6.15 (Low density parity check matrix (LDPC) Codes)**.** *Let $c, d, n \in \mathbb{N}$. A (c,d,n)-LDPC code is given by a (c,d)-regular bipartite graph $([n],[m],E)$ (called a factor graph) with n left vertices and $m = nc/d$ right vertices, called parity checks, such that all right vertices have degree d and all left vertices have degree c. The code is defined to be*

$$C = \left\{ w \in \mathbb{F}_2^n : \forall j \in [m], \sum_{\{i,j\} \in E} w(i) = 0 \ mod\ 2 \right\}.$$

There are many ways to construct LDPC codes: A simplest way is to choose a random (c,d) regular bipartite graph. We also remark that the expander codes we defined in section 6.2 also give LDPC codes. In particular if the parity check matrix of the base code itself is an LDPC code then the blown-up expander code will also be an LDPC code.

Definition 6.16 (Smooth code)**.** *Let $c, d, n \in \mathbb{N}$ and $\alpha, \beta, \delta > 0$. A (c,d,n)-LDPC code $C \subseteq \mathbb{F}_2^n$ is (α,β,δ)-smooth if for every $Y \subseteq [m]$ with $|Y| \leq \alpha \cdot m$ there is some $X \subseteq [n]$ with $|X| \leq \beta \cdot n$ such that the code $C(Y^-)|_{X^-}$ has distance at least δ, where $Y^- = [m] \setminus Y$ and $X^- = [n] \setminus X$. Here the code $C(Y^-)|_{X^-}$ is the code obtained by removing the constraints in Y and then removing the coordinates of the code in X.*

It is not hard to see that expander codes are smooth. This is because given a (regular) expander graph, say we (adversarially) delete $|Y| = 0.01V$ vertices. Then, there is a set $\tilde{Y} \supset Y$ such that $|\tilde{Y}| \leq 2|Y|$ and the induced graph on $G[V \setminus \tilde{Y}]$ is also an expander. So, we can simply define $X = E \setminus E(V \setminus \tilde{Y})$ (in the smoothness definition).

Lemma 6.17 ([DSW06])**.** *For any c,d,m-LDPC code C_1 that is $(\alpha, \frac{\delta_{C_1}}{2}, \frac{\delta_{C_1}}{2})$-smooth and (a binary linear code) $C_2 \subseteq \mathbb{F}_2^n$ let $C = C_1 \otimes C_2$. For any $F \notin C_1 \otimes C_2$, if $\frac{1}{2}(\delta^{col}(F,C) + \delta^{row}(F,C)) \leq \min\left\{\alpha\frac{\delta_{C_2}}{2d^2}, \frac{\delta_{C_1}\delta_{C_2}}{8}\right\}$ then, $\beta(f) \geq 1/8$. Therefore,*

$$\beta(C) \geq \min\left\{\alpha\frac{\delta_{C_2}}{2d^2}, \frac{\delta_{C_1}\delta_{C_2}}{8}\right\}.$$

Proof Sketch. Fix such a word F and let $F_1 \in C_1 \otimes \mathbb{F}_2^{n_2}$ be the closest word to F and $F_2 \in \mathbb{F}_2^{n_1} \otimes C_2$ be the closest word to F. Define (the error matrix) $E = F_1 - F_2$. Observe that

$$\frac{\|E\|_1}{n_1 n_2} = \delta(F_1, F_2) \leq \delta^{row}(F) + \delta^{col}(F).$$

So, the assumption of the lemma implies $\frac{\|E\|_1}{n_1 n_2} \leq 2\min\{\alpha\frac{\delta_{C_2}}{2d^2}, \frac{\delta_{C_1}\delta_{C_2}}{8}\}$.

Lemma 6.18. *Let $\{i_1, \ldots, i_d\}$ be a parity constraint of (the LDPC code) C_1 (i.e., every codeword of $c \in C_1$ satisfies $c_{i_1} + \cdots + c_{i_d} = 0$). Let E^i denote the i-th row of E. Suppose $\frac{\|E^{i_j}\|_1}{n_2} < \delta_{C_2}/d$ for* every $j \in [d]$. *Then $E^{i_1} + \cdots + E^{i_d} = 0$.*

To put it differently, if the rows corresponding to this particular constraint of C_1 in E are "sparse", then every column of E (and every column of F) satisfies this constraint

Proof. Recall F_1^i is the i-th row of F_1. Note that these rows are not necessarily codewords of any nice code - it is only the columns of F_1 that are codewords of C_1). Thus, observe that $F_1^{i_1} + \cdots + F_1^{i_d} = 0$. On the other hand, since each row of F_2 is a word of C_2, we have $F_2^{i_1} + \cdots + F_2^{i_d}$ is a codeword of C_2. But this implies

$$E^{i_1} + \cdots + E^{i_d} = (F_1^{i_1} - F_2{}^{i_1}) + \cdots + (F_1^{i_d} - F_2^{i_d}) = (F_1^{i_1} + \cdots + F_1^{i_d}) - (F_2^{i_1} + \cdots + F_2^{i_d}) = (F_2^{i_1} + \cdots + F_2^{i_d}) \in C_2$$

Now we use the fact that the E^i's have small weight. This implies that

$$\frac{\|E^{i_1} + \cdots + E^{i_d}\|_1}{n_2} = \sum_{j=1}^{d} \frac{\|E^{i_j}\|_1}{n_2} < \delta_{C_2}.$$

But this implies that indeed we must have $E^{i_1} + \cdots + E^{i_d} = 0$. □

Having the above fact, we can divide the rows of E into two groups: Heavy group which has a fraction at least δ_{C_2}/d many ones, and the light rows. Assuming that $\frac{1}{2}(\delta^{col}(F, C) + \delta^{row}(F, C)$ is small, we can see that most rows of E are sparse. We delete every constraint in the LDPC code C_1 which has a heavy row. By smoothness of C_1 we can drop a small fraction of the coordinates of C_1 (and equivalently, rows of E) such that the resulting code still has distance $\delta_{C_1}/2$. But then all column constraints code E' and similarly F_2' are satisfied. So, the number of changes we need to do to correct F is at most n_2 times the number of deleted rows, plus the $\delta^{row}(F)n_1n_2$. □

It turns out that for the analysis of locally testable code, we need a slightly different property of tensor codes called agreement testability.

Definition 6.19 (Agreement Testability)**.** *Let $\kappa > 0$. Let $C_1 \subseteq \mathbb{F}_2^{n_1}, C_2 \subseteq \mathbb{F}_2^{n_2}$. We say that $C_1 \otimes C_2$ is κ-agreement testable if for every $w_1 \in C_1 \otimes \mathbb{F}_2^{n_2}$ and $w_2 \in \mathbb{F}_2^{n_1} \otimes C_2$, there exists $w \in C_1 \otimes C_2$ such that*

$$\kappa \cdot \left(\mathbb{P}_i\left[w_1(i,.) \neq w(i,.)\right] + \mathbb{P}_j\left[w_2(.,j) \neq w(.,j)\right]\right) \leq \mathbb{P}_{i\sim[n_1], j\sim[n_2]}\left[w_1(i,j) \neq w_2(i,j)\right]$$

It is not hard to see that robust testability implies agreement testability. In particular if $C_1 \otimes C_2$ is β-robustly testable then $C_1 \otimes C_2$ is κ-agreement testable, for

$$\kappa = \frac{2\beta\delta_1\delta_2}{\delta_2 + \delta_1(1 + 2\beta)}.$$

6.6 Intro to Group Theory

A group is a set G together with a binary "product" operations such that for any two elements $a, b \in G$, $ab \in G$ and satisfies the following properties:

Associativity For all $a, b, c \in G$, $(ab)c = a(bc)$.

Identity Element There exists an element $e \in G$ such that $ea = a$ for all $a \in G$.

Inverse Element For any element $a \in G$ there exists an element a^{-1} such that $aa^{-1} = a^{-1}a = e$.

Note that the product operations is not necessarily commutative, i.e., $ab \neq ba$.

Definition 6.20 (Generator of a Group)**.** *We say a set $A \subseteq G$ is a generator of G if every element $a \in G$ can be generated by taking a product of a (finitely) many elements of A. For example G is a generator of itself. We say A is a* symmetric *generator if for any $a \in A$ we also have $a^{-1} \in A$.*

Definition 6.21 (Cayley Graphs)**.** *Given a (finite) group G with a symmetric generator A, we can definite a (Cayley) graph,* $\mathrm{Cay}(G, A)$ *with vertex set identified with elements of G, and for any vertex $g \in G$ and element $a \in A$, we have an edge $\{g, ag\}$. Note that since A is a symmetric generator, the edge $\{g, ag\}$ coincides with the edge $\{ag, (a^{-1})ag\}$, i.e., the graph is undirected.*

In [**Mor94**], Morgenstern presented for every prime power q, infinitely many groups $G_i = PSL_2(q^i)$ each with a symmetric generator A_i of size $q + 1$ generators such that $\mathrm{Cay}(G_i, A_i)$ is Ramanujan, i.e., λ_2 of the simple random walk on G is at most $\frac{2\sqrt{q}}{q+1}$. Here, we do not provide more details; we just point out that $PSL_2(q^i)$ is the group of all 2×2 matrices with entries from $\mathbb{F}_{q^i}$ of determinant 1 where we have quotient out by the set of multiplies of the identity matrix in the group.

6.7 Left-Right Cayley Complex

Definition 6.22 (Left-Right Cayley Complex)**.** *Let G be a finite group with two symmetric sets of generators A, B. We assume that the identity element of G is neither in A nor in B.*

Define the Left-Right Cayley Complex $X = \mathrm{Cay}^2(A, G, B)$ as follows

- *The vertices are the elements of the group G, $X(0) = G$.*
- *The edges, $X(1) = X^A(1) \cup X^B(1)$ where*

  $$X^A(1) = \{\{g, ag\} : g \in G, a \in A\}, X^B(1) = \{\{g, gb\} : g \in G, b \in B\}.$$

  Note that we always multiply a on the left and b on the right. The non-commutativity of the group plays an important role here.
- *The faces of "dimension" 2 are squares*

  $$[a, g, b] := \{g, ag, agb, gb\}.$$

  Note that since A, B are symmetric, this square is equivalent to the squares

  $$[a^{-1}, ag, b], [a^{-1}, agb, b^{-1}], [a, gb, b^{-1}].$$

Note $\mathrm{Cay}^2(G, A, B)$ is a 2-dimensional *square* complex (as we will see later in a normal 2-dimenional simplicial complex every face of dimension 2 is a triangle).

Note that $(G, X^A(1))$ and $(G, X^B(1))$ are exactly the Cayley graphs we defined in the previous section. The important point here is that a always multiplies on left and b multiples on right. This gives a local commutativity which leads to many squares/4cycles. Namely, we can start from any g and construct a square as explained above.

We assume that the generators A, B satisfy the total no-conjugacy condition (TNC), namely

$$g^{-1}ag \neq b, \forall g \in G, a \in A, b \in B. \tag{TNC}$$

No Parallel edges. The above condition implies that the $(G, X^A(1) \cup X^B(1))$ has no parallel edges, i.e., $\{g, ag\} \neq \{g', g'b\}$ for all $g, g' \in G, a \in A, b \in B$. If not, either $g = g'$, but then (TNC) implies $ag \neq gb$ so we get two different edges. Or, $ag = g'$ but then again (TNC) implies $a^{-1}ag \neq g'b$. This implies that

$$|X(1)| = \frac{|A| + |B|}{2}|G|. \tag{6.1}$$

In addition, we can show all squares are simple 4-cycles. Pick a square $\{g, ag, agb, gb\}$. We have $g \neq ag, g \neq agb\}$ because the identity elements are not in A, B. (TNC) implies that $g \neq agb$. So, we have

$$|X(2)| = \frac{|A| \cdot |B|}{4}|G|. \tag{6.2}$$

Definition 6.23 (Links)**.** *For each $g \in G$, the link of g is $X_g \subseteq X(2)$ is defined as $\{[a, g, b]|a \in A, b \in B\}$ the set of all squares that have g. Recall that $|X(g)| = |A| \cdot |B|$ by the (TNC) property.*

For every edge $e = \{g, ag\}$, the link of e is denoted $X_e \subseteq X(2)$ is defined as $\{[a, g, b]|b \in B\}$ the set of all squares that have the edge e. So, by (TNC), $|X_e| = |B|$. Similarly if $e = \{g, gb\}$ we let $X_e = \{[a, g, b]|a \in A\}$.

Later on we will talk about simplicial complexes the above definition will be extended.

6.8 Left-Right Cayley Complex Error Correcting Code

For a group G and symmetric generators A, B with (TNC) as defined above, let $C_A \subseteq \mathbb{F}_2^A, C_B \subseteq \mathbb{F}_2^B$ be binary linear error correcting codes with rate ρ_A, ρ_B and minimum distance δ_A, δ_B respectively.

We define the code $C = C[G, A, B, C_A, C_B]$ as follows: For an edge $e = \{g, ag\}$ define a local code

$$C_e = \{f \in \mathbb{F}_2^{X_e} : f([a, g, .]) \in C_B\}$$

and similarly for $e = \{g, gb\}$,

$$C_e = \{f \in \mathbb{F}_2^{X_e} : f([., g, b]) \in C_A\}.$$

Consequently, for any vertex $g \in G = X(0)$ define the local tensor code,

$$C_g = \{f \in \mathbb{F}_2^{X_g} : f([., g, .]) \in C_g\}.$$

We define the global code C,

$$C = \{f \in \mathbb{F}_2^{X(2)} : f|_{X_g} \in C_g, \forall g \in X(0)\}$$

or equivalently, by Exercise 6.13,

$$C = \{f \in \mathbb{F}_2^{X(2)} : f|_{X_e} \in C_e, \forall e \in X(1)\}.$$

Compare this definition with Definition 6.8. Here, we still have a local code for every vertex, but instead of the local code being an arbitrary code, independent of all other vertices, we ask it to be a tensor code supported on all squares that contain that vertex. So, in some sense this code is one dimension larger than the expander codes and that extra dimension enforces a lot more consistency.

Observe that here we have a set of highly dependent linear constraints on the vertices $g \in X(0)$, such that the constraints associated with adjacent vertices have significant pairwise intersections. Specifically, for every two neighboring vertices, g, ag the inspected $|A| \times |B|$ tensor code share $|B|$-entries that correspond to the edge $\{g, ag\}$. Hence, if we have a word in which a constraint of the parity check matrix of a vertex g is violated, it leads to violating many other (different) constraints of neighbors of g. In particular, if we define a

"new" code by dropping few constraints from the low-density parity-check matrix of C_g, the code C remains invariant.

It is fundamental to the proof of local testability that all A-edges have exactly the same code in their local view. Otherwise, we would not have a tensor code on vertices.

Next, we briefly discuss rate and distance of left-right Cayley codes and in the next section we discuss their local testability.

Rate: There are basically two ways to measure the rate of code C: To count the number of constraints imposed by local edge codes C_e or to count the number of constraints imposed by local vertex codes C_g. We will count the former: For every edge $e = \{g, ag\}$, there are $|B|(1-\rho_B)$ constraints and for every $f = \{g, gb\}$ there are $|A|(1-\rho_A)$ constraint. Also, recall the length of C is $|X(2)| = |G||A||B|/4$. Putting these together,

$$\rho_C \geq \frac{|G|A||B|/4 - |X^A(1)||B|(1-\rho_B) - |X^B(1)||A|(1-\rho_A)}{|G||A||B|/4} = 2(\rho_A + \rho_B) - 3$$

where we used $|X^A(1)| = |G||A|/2$ and $|X^B(1)| = |G||B|/2$.

Distance Suppose that both Cayley graphs $\mathrm{Cay}(G, A), \mathrm{Cay}(G, B)$ are λ-expanders.

Let $f \in C$ be a codeword and $f \neq 0$. So, there must exist a vertex $g_0 \in G$ such that $w_{g_0} = f|_{X_{g_0}} \neq 0$. Since $w_{g_0} = G_A Y G_B^T$ for some matrix $Y \in \mathbb{F}_2^{\rho_A|A| \times \rho_B|B|}$, there are $\delta_B|B|$ nonzero columns in w_g and $\delta_A|A|$ nonzero rows. Let A_1 be the set of non-zero rows. Fix a nonzero row $a \in A_1$ and let $f_a(\{g, gb\}) = f_a([a, g, b])$ for any $g \in G, b \in B$, so $f_a \neq 0$.

Now, consider the $\mathrm{Cay}(G, B)$ with the base code C_B around every vertex. It follows by Theorem 6.9 that the weight of f is at least $\delta_B(\delta_B - \lambda)|B|$, i.e., we can write

$$\mathbb{P}_{g,b}\left[f_a(\{g, gb\} \neq 0\right] = \mathbb{P}_{g,b}\left[f([a, g, b]) \neq 0\right] \geq \delta_B(\delta_B - \lambda).$$

To bound the weight of f write

$$\|f\|_1 = \mathbb{P}_{a,g,b}\left[f([a, g, b]) \neq 0\right] = \mathbb{P}_a\left[a \in A_1\right] \mathbb{P}_{b,g}\left[f_a(\{g, gb\}) \neq 0 | a \in A_1\right] \geq \delta_A \cdot \delta_B(\delta_B - \lambda)$$

Roughly speaking, to choose a random square, we first choose a random a and then we choose a random square that contains a. Following the same argument for B_1 we can say

$$\delta_C \geq \delta_A \delta_B(\max\{\delta_A, \delta_B\} - \lambda)$$

6.9 Local Testability of Left-Right Complex Error Correcting Codes

The following theorem is the main technical result of Dinur, Evra, Livne, Lubotzky, Mozes [DELLM21].

Theorem 6.24 (Main Theorem)**.** *Suppose* $X = \mathrm{Cay}^2(A, G, B)$ *such that* $\mathrm{Cay}(G, A), \mathrm{Cay}(G, B)$ *are* λ*-expanders, and* (TNC) *holds. Further assume* $C_A \otimes C_B$ *is* κ_0*-agreement testable. If* $c_0 := \frac{\kappa_0}{8+\kappa_0} \cdot \min(\delta_A, \delta_B) > \lambda$ *then* $C = C[G, A, B, C_A, C_B]$ *is*

$$\min\left\{\frac{1}{4(1+|A|+|B|)}, \frac{c_0 - \lambda}{2(|A|+|B|)}\right\} =: \kappa$$

locally testable with $|A| \cdot |B|$ *queries. For any* $f \in \mathbb{F}_2^{X(2)}$,

$$\mathbb{P}_{g \sim X(0)}\left[f|_{X_g} \notin C_g\right] \geq \kappa \cdot \delta(f, C).$$

In other words, here there is a simple local tester algorithm: Given a word $f \in \mathbb{F}_2^{X(2)}$, choose $g \sim X(0)$ uniformly at random and test if $f|_{X_g} \in C_g$.

The above theorem should come with a decoding algorithm. Given a word $f \in \mathbb{F}_2^{X(w)}$, a natural idea is to iteratively modify f such that in each iteration we select an arbitrary 4-cycle $[a, g, b]$ and reset $f([a, g, b])$ such that it satisfies a majority of the checks that look at it (i.e., we set $f([a, g, b]) = \sigma$ if $[a, g, b]$ is assigned σ in a majority of the $C_A \otimes C_B$ tensor codes that contain this square). The decoding process terminates when no additional modification is possible (i.e., where for each $[a, g, b] \in X(2)$ the value of $f([a, g, b])$ equals the majority value assigned to this square). Although it seems that this candidate decoder works well, i.e., correctly decodes f, when f is *close* to C, here, we need to show such a decoder works on *any* f and that is unclear.

Local Self-Correcting Algorithm Define $\zeta(f) := \mathbb{P}_g \left[f|_{X_g} \notin C_g \right]$. Given $f \in \mathbb{F}_2^{X(2)}$, the self-correcting algorithm is supposed to find a codeword $c \in C$ such that $\delta(f, c) \leq O(\zeta(f))$.

The first step in the self-correcting algorithm of [DELLM21] is to substitute $f|_{X_g}$ with the closest codeword $w_g^0 \in C_g$, for all $g \in X(0)$. Then, comes the decoding algorithm which in steps tries to decrease the following potential function:

$$\Delta(W) := \mathbb{P}_{e=\{g,g'\} \sim X(1)} \left[w_g|_{X_e} \neq w_{g'}|_{X_e} \right]$$

In other words, the local views of g, g' have a common row or column in their tensor code. In this potential function we count how many fraction of these local views disagree. Now, the algorithm at each steps replaces w_g with a new code word $w \in C_g$ if that decreases the potential function.

In the following we always assume $W^0 = \{w_g^0\}_{g \in G}$ is the initial substitution, while $W = \{w_g\}_{g \in G}$ is the final output of the local self-correcting algorithm. First notice $\Delta(f) = 0$; but

$$\Delta(W^0) \leq 2\zeta(f) \tag{6.3}$$

This is because f is an actual code so any two neighboring g, g' agree on their common row/column. On the other hand, for every dispute edge $e = \{g, g'\}$ in W^0, we have either $f_{X_g} \notin C_g$ or $f_{X_{g'}} \notin C_{g'}$. Therefore, the process of choosing an edge $e \sim X(1)$ and then an endpoint u.a.r. will lead to a modified vertex with probability at least $\Delta(W)/2$ proving the above inequality.

Fact 6.25. *Suppose the algorithm succeeds, $\Delta(W) = 0$. Then,*

$$\delta(f, C) \leq \delta(f, W) \leq 4(1 + |A| + |B|)\zeta(f)$$

Proof. By triangle inequality,

$$\delta(f, W) \leq \delta(f, W_0) + \delta(W_0, W).$$

To bound the latter notice that $\Delta(W^0) \leq 2\zeta(f)$ and the number of disputed edges decreases by at least one in every iteration of the algorithm. So, the algorithm runs for at most $2\zeta(f)|X(1)|$ many iterations and in this process we change at most $2\zeta(f)|X(1)|$ many w_g's. So the total number of $f|_{X_g}$'s that we change is at most

$$\zeta(f)(|X(0)| + 2|X(1)|) \underset{(6.1)}{=} \zeta(f)(1 + |A| + |B|)|X(0)|$$

Since changing a local view at g correspondings to changing $|A||B|$ many squares,

$$\delta(f, C) \leq \frac{\zeta(f)(1 + |A| + |B|)|X(0)||A||B|}{|X(2)|} \underset{(6.2)}{=} 4\zeta(f)(1 + |A| + |B|)$$

as desired. □

The main technical statement of the proof is the following proposition:

Proposition 6.26. *If $\Delta(W) > 0$ then, $\Delta(W) \geq \epsilon_0 = \frac{c_0 - \lambda}{|A|+|B|}$ (as defined in Theorem 6.24, where $c = \kappa_0|A| + |B| \cdot \min\{\delta_A, \delta_B\}$)*

Having that, let us finish the proof of Theorem 6.24. If the self-correcting algorithm succeeds and finds $\Delta(W) = 0$, then by Fact 6.25, $\delta(f, C) \leq 4(1 + |A| + |B|)\zeta(f)$ and we are done. Otherwise, we have $\Delta(W) \geq \epsilon_0$. And in this case

$$\delta(f, C) \leq 1 = \frac{2}{\epsilon_0} \cdot \frac{\epsilon_0}{2} \leq \frac{2}{\epsilon_0} \cdot \frac{\Delta(W)}{2} \leq \frac{2}{\epsilon_0} \cdot \frac{\Delta(W^0)}{2} \underset{(6.3)}{\leq} \frac{2(|A| + |B|)}{c_0 - \lambda}\zeta(f)$$

as desired.

6.10 Proof of Proposition 6.26

This is the most interesting part of the proof. So, basically, the proposition implies that there are local optima to the $\Delta(.)$ potential function that are very far from the code C. Let D be the set of *dispute edges*

$$D = \{e = \{g, g'\} \in X(1) : w_g|_{X_e} \neq w_{g'}|_{X_e}\}.$$

So, by definition, $\Delta(W) = |D|/|X(1)$, and to prove the claim we need to show (assuming D is non-empty),

$$|D| \geq \frac{c_0 - \lambda}{|A| + |B|}|X(1)| = \frac{c_0 - \lambda}{2} \cdot |X(0)|.$$

The proof can be seen as a higher dimensional analogue of Theorem 6.9.

For an edge $\{g, ag\} \in X^A(1)$ let

$$E^{\parallel}(\{g, ag\}) = \{\{gb, agb\} \in X^A(1) : b \in B\}$$

and similarly for an edge $\{g, gb\} \in X^B(1)$,

$$E^{\parallel}(\{g, gb\}) = \{\{ag, agb\} \in X^B(1) : a \in A\}.$$

For a vertex $g \in .G$, let

$$E^A(g) = \{\{g, ag\} : a \in A\}, E^B(g) = \{\{g, gb\} : b \in B\}.$$

Fact 6.27. *Suppose $\{g, ag\} \in D$, then*

$$|D \cap E^B(g)| + |D \cap E^B(ag)| + |D \cap E^{\parallel}(\{g, ag\})| \geq \delta_B|B|.$$

A similar statement holds for any edge $\{g, gb\} \in D$.

The above can be seen analogously to proof of Theorem 6.9; recall there we showed that if a f is a word, then every vertex adjacent to an edge (with value 1 in f) is in fact adjacent to $\delta_{C_0}d$ many such edges. Here, we say if g is adjacent to a dispute edge it must be adjacent (or parallel) to $\delta_B|B|$ many such edges.

Proof. Fix an arbitrary dispute edge, say $e = \{g, ag\} \in D$; so $w_g|_{X_e} \neq w_{ag}|_{X_e}$. Observe that both $w_g|_{X_e}, w_{ag}|_{X_e}$ are valid codewords of C_e. So, $\delta(w_g|_{X_e}, w_{ag}|_{X_e}) \geq \delta_B$, i.e., these two codewords differ in $|B|\delta_B$ many squares. The observation is that for any such square say $\{g, ag, agb, gb\}$, there must be another edge of the square along which there is also a disagreement on this square, i.e., at least two edge of such square belong to D. The fact follows. □

For the sake of these lecture notes we prove a simpler statement.

Case 1: Assume for any $e \in D$, $|D \cap E^{\|}(e)| \geq \delta_B |B|$.

For every edge $\{g, ag\}$ (or $\{g, a^{-1}g\}$) we say it has label $[a] = \{a, a^{-1}\}$. Naturally, we define $X^\sigma(1)$ to denote all edges labeled with σ. (note that assuming $a \neq a^{-1}$, $|X^\sigma(1)| = |G|$.

Definition 6.28 (Parallel Random Walk)**.** *We define a random walk on the set of edges $X(1)$ as follows. Starting from an edge e, choose uniformly a square containing e and then move to the unique edge $e' \neq e$ on that square with the same label as e.*

We use $P^{\|}$ to denote the corresponding random walk operator. This operator decomposes into corresponding random walk operators on each possible label.

$$P^{\|} f = \sum_{\sigma} P^{\|}_{\sigma} f|_{X^\sigma(1)}$$

where the sum is over all labels σ.

For $f, h : X(1) \to \mathbb{R}$ define the inner product

$$\langle f, h \rangle_{\pi_1} = \frac{1}{2}\mathbb{E}_{e \sim X^A(1)} f(e)g(e) + \frac{1}{2}\mathbb{E}_{e \sim X^B(1)} f(e)h(e).$$

As usual, this inner product defines another inner product π_0 on the group G, $\langle f, h \rangle_{\pi_0} = \mathbb{E}_{g \sim \pi_0} f(g)h(g)$.

It follows from the assumption of this case that

$$\langle P^{\|} \mathbf{1}_D, \mathbf{1}_D \rangle \geq \Omega(1) \min\{\delta_A, \delta_B\} \langle \mathbf{1}_D, \mathbf{1}_D \rangle$$

Lemma 6.29. *Assume both $\mathrm{Cay}(G, A)$ and $\mathrm{Cay}(G, B)$ are λ-expanders. If $\langle P^{\|} \mathbf{1}_D, \mathbf{1}_D \rangle \geq c \langle \mathbf{1}_D, \mathbf{1}_D \rangle$, then there exists a label σ, such that $|D \cap X^\sigma(1)| \geq (c - \lambda)|G|$.*

Note that in such a case we are basically done with the proof of Proposition 6.26.

Proof. We write $f = \mathbf{1}_D$. First, by assumption,

$$c \leq \frac{\langle P^{\|} f, f \rangle}{\langle f, f \rangle} = \frac{\mathbb{E}_\sigma \mathbb{E}_{e \sim X^\sigma(1)} f(e) P^{\|}_\sigma f(\sigma)}{\mathbb{E}_\sigma \mathbb{E}_{e \sim X^\sigma(1)} f(e)^2} \leq \max_\sigma \frac{\mathbb{E}_{e \sim X^\sigma(1)} f(e) P^{\|}_\sigma f(e)}{\mathbb{E}_{e \sim X^\sigma(1)} f(e)^2}$$

Where in $\mathbb{E}_\sigma$ the expectation is over a random label; namely with probability one have we choose a uniformly random label from A and otherwise from B.

Wlog suppose $\sigma = [a] = \{a, a^{-1}\}$ is the label maximizing the ratio in the RHS. Define $h_\sigma(g) := f(\{g, ag\})$ for all $g \in G$. Let P_B be the simple random walk operator on $\mathrm{Cay}(G, B)$. We can write

$$\frac{\langle P_B h_\sigma, h_\sigma \rangle}{\langle h_\sigma, h_\sigma \rangle} = \frac{\mathbb{E}_{g \sim \pi_0} h_\sigma(g) \mathbb{E}_{\{g,gb\}|g} h_\sigma(gb)}{\mathbb{E}_{g \sim \pi_0} h_\sigma(g)^2} = \frac{\mathbb{E}_{g \sim \pi_0} f(\{g, ag\}) \mathbb{E}_{\{g,gb\}|g} f(\{gb, agb\})}{\mathbb{E}_{g \sim \pi_0} f(\{g, ag\})^2} = \frac{\mathbb{E}_{e \sim X^\sigma(1)} f(e) P^{\|}_\sigma f(e)}{\mathbb{E}_{e \sim X^\sigma(1)} f(e)^2}$$

In the last identity we used that choosing a random edge $e \sim X^\sigma(1)$ can be done by choosing a random $g \sim \pi_0$ and then choosing edge $\{g, ag\}$. But the RHS is at least c by the choice of σ. So, $\frac{\langle P_B h_\sigma, h_\sigma \rangle}{\langle h_\sigma, h_\sigma \rangle} \geq c$. Therefore, the lemma follows by Lemma 6.6, $\langle h_\sigma, h_\sigma \rangle \geq c - \lambda$. It follows that the number of nonzero entries in h_σ is at least $|G|(c - \lambda)$. So, the number of non-zero entries of f is also at least $(c - \lambda)|G|$. □

Case 2: Assume for any $e = \{g, ag\} \in D$, $|D \cap E^B(g)| + |D \cap E^B(ag)| \geq \delta_B |B|$. Now, we get to the more interesting part. Note that so far we haven't used robust testability property of the code $C_A \otimes C_B$. This will crucially show up in this case.

Lemma 6.30. *Suppose $C_A \otimes C_B$ is κ_0 agreement testable. Then, for any $g \in G$,*

$$\frac{|D \cap E^A(g)|}{|A|} + \frac{|D \cap E^B(g)|}{|B|} = \mathbb{P}_a\left[\{g, ag\} \in D\right] + \mathbb{P}_b\left[\{g, gb\} \in D\right] \leq \frac{1}{\kappa_0}\mathbb{P}_{a \in A, b \in B}\left[\{ag, agb\} \in D \text{ or } \{gb, agb\} \in D\right].$$

Roughly speaking, this lemma shows that disagreements on edges that are incident at a vertex g translate to a proportional number of disagreements on the edges that are in 4-cycles/squares that contain g but are *not incident* to it.

Proof. Define $w_1, w_2 \in \mathbb{F}_2^{A \times B}$. For any $a \in A, b \in B$ define

$$w_1(a, b) = w_{ag}([a^{-1}, ag, b]) \qquad \text{and} \qquad w_2(a, b) = w_{gb}([a, gb, b^{-1}]).$$

Observe that $w_1 \in \mathbb{F}_2^A \otimes C_B$ and $w_2 \in C_A \otimes \mathbb{F}_2^B$. This is because the a^{-1}-th row of w_1 is the same as the a^{-1}-th row of w_{ag}, thus a word of C_B. Similarly, b^{-1}-th column of w_2 comes from b^{-1}-th column of w_{gb}. The observation is that for any a, $w_1(a, .) \neq w_g(a, .)$ iff $\{g, ag\} \in E^A(g)$ is in D and similarly, $w_2(., b) \neq w_g(., b)$ iff $\{g, gb\} \in E^B(g)$ is in D.

It follows by κ_0-agreement testability of $C_A \otimes C_B$ (see Definition 6.19) that, there exists a codeword $w^* \in C_A \otimes C_B$ such that

$$\mathbb{P}_a\left[w^*(a, .) \neq w_1(a, .)\right] + \mathbb{P}_b\left[w^*(., b) \neq w_2(., b)\right] \leq \frac{1}{\kappa_0}\mathbb{P}_{a,b}\left[w_1(a, b) \neq w_2(a, b)\right].$$

Since w is a local optima of the self-correcting algorithm, we must have that the number of disputed edges do not decrease if we replace w_g with w^*. That means that

$$\begin{aligned}\mathbb{P}_a\left[w^*(a, .) \neq w_1(a, .)\right] + \mathbb{P}_b\left[w^*(., b) \neq w_2(., b)\right] &\geq \mathbb{P}_a\left[w_g(a, .) \neq w_1(a, .)\right] + \mathbb{P}_b\left[w_g(., b) \neq w_2(., b)\right] \\ &= \mathbb{P}_a\left[\{g, ag\} \in D\right] + \mathbb{P}_b\left[\{g, gb\} \in D\right].\end{aligned}$$

Finally, notice that if $w_1(a, b) \neq w_2(a, b)$ equivalently we have $w_{ag}(a^{-1}, ag, b]) \neq w_{gb}([a, gb, b^{-1}])$. But this means that (exactly) one of the two edges $\{ag, agb\}, \{gb, agb\}$ are in D as desired. □

Next, we construct a λ-expander random walk, that naturally jumps from a vertex g to edges of 4-cycles that are not incident to g. This together with the previous lemma implies that the number of dispute edges should be a constant fraction of $|G|$.

Let $P_0 = \frac{1}{2}P_A + \frac{1}{2}P_B$, be the simple random walk operator in $\mathrm{Cay}(G, A \cup B)$ where as usual P_A, P_B are the random walk operators of $\mathrm{Cay}(G, A), \mathrm{Cay}(G, B)$ respectively. Since P_A, P_B have the same uniform stationary distribution, it follows that $\lambda_2(P_0) \leq \lambda$. Consider the down uperator $P^{\downarrow} : \mathbb{R}^{X(1)} \to \mathbb{R}^{X(0)}$ and (its adjoint) the up operator $P^{\uparrow} : \mathbb{R}^{X(0)} \to \mathbb{R}^{X(1)}$. In particular, for a function $f_1 \in \mathbb{R}^{X(1)}$,

$$P^{\downarrow} f_1(g) = \mathbb{E}_{e \sim \pi_1|_g} f_1(e) = \frac{1}{2}\mathbb{E}_a f_1(\{g, ag\}) + \frac{1}{2}\mathbb{E}_b f_1(\{g, gb\}).$$

Similarly, for $f_0 : \mathbb{R}^{X(0)} \to \mathbb{R}^{X(1)}$, and any edge $e = \{g_1, g_2\}$ we have

$$P^{\uparrow} f_0(\{g_1, g_2\}) = \frac{1}{2}f_0(g_1) + \frac{1}{2}f_0(g_2).$$

It turns out that these operators are adjoint of each other:

Exercise 6.31. *Show that for any function $f_1 \in \mathbb{R}^{X(1)} and g_0 \in \mathbb{R}^{X_0}$*

$$\langle P^{\downarrow} f_1, g_0 \rangle_{\pi_0} = \langle f_1, P^{\uparrow} g_0 \rangle_{\pi_1}.$$

The following fact is an immediate consequence of this exercise.

Fact 6.32. *Let $P = P^{\uparrow} P_0 P^{\downarrow}$. Then, $\lambda_2(P) \leq \lambda$.*

Proof. Let $f_1 \in \mathbb{R}^{X(1)}$ such that $\langle f_1, \mathbf{1} \rangle_{\pi_1} = 0$. Let $f_0 = P^{\downarrow} f_1$. First notice

$$\langle f_0, \mathbf{1} \rangle_{\pi_0} = \langle P^{\downarrow} f_1, \mathbf{1} \rangle_{\pi_0} = \langle f_1, P^{\uparrow} \mathbf{1} \rangle_{\pi_1} = \langle f_1, \mathbf{1} \rangle_{\pi_1} = 0.$$

Therefore,

$$\langle P f_1, f_1 \rangle_{\pi_1} = \langle P^{\uparrow} P_0 P^{\downarrow} f_1, f_1 \rangle_{\pi_1} = \langle P_0 P^{\downarrow} f_1, P^{\downarrow} f_1 \rangle_{\pi_0} = \langle P_0 f_0, f_0 \rangle_{\pi_0} \underset{\langle f_0, \mathbf{1} \rangle = 0}{\leq} \lambda_2(P_0) \langle f_0, f_0 \rangle_{\pi_0} = \lambda \| P^{\downarrow} f_1 \|^2$$

The RHS is smaller than $\|f_1\|^2$ simply because $P^{\downarrow}$ is a stochastic operator. □

Exercise 6.33. *Show that for any stochastic operator $P \in \mathbb{R}^{m \times n}$ and any function $f \in \mathbb{R}^n$, $\|Pf\| \leq \|f\|$.*

Lemma 6.34. *Let $f = \mathbf{1}_D \in \mathbb{R}^{X(1)}$. Then,*

$$\langle Pf, f \rangle_{\pi_1} \geq \frac{\kappa_0}{8} \min\{\delta_A, \delta_B\} \langle f, f \rangle_{\pi_1}.$$

Proof. So, let us understand the LHS: Given an edge $e \in D$ the LHS is the probability that one step of the walk $P = P^{\uparrow} P_0 P^{\downarrow}$ jumps to another dispute edge in D. So, let us first understand this walk:

Step 1) Choose at random one of the endpoints of the edge, $g_1 \in e$.

Step 2) With probability $\frac{1}{2}$ let $g_2 = a_1 g_1$ for a random $a_1 \in A$, and with probability $\frac{1}{2}$ let $g_2 = g_1 b_1$ for a random $b_1 \in B$.

Step 3) With probability $\frac{1}{2}$ let $e' = \{g_2, a_2 g_2\}$ for a random $a_2 \in A$, and with probability $\frac{1}{2}$ let $e' = \{g_2, g_2 b_2\}$ for a random $b_2 \in B$. Output e'.

Now, fix an edge $e = \{g, ag\} \in D$. By Lemma 6.30,,

$$\mathbb{P}_{a,b} \left[\{a g_1, a g_1 b\} \in D \text{ or } \{g_1 b, a g_1 b\} \in D \right] \geq \kappa_0 \frac{d_{g_1}}{|B|},$$

where $d_{g_1} = |D \cap E^B(g_1)|$. Now, the question is what is the probability that e' is one of the edges $\{a g_1, a g_1 b\}$ or $\{g_1 b, a g_1 b\}$? This only happens, if in steps 2,3 we walk in alternating color, i.e., in step 2 we choose an edge of color A adjacent to g_1 then in step 3 we choose an edge of color B adjacent to g_2 or vice versa. Therefore,

$$\begin{aligned}
\mathbb{P}\left[e' \in D | g_1\right] &\geq \mathbb{P}\left[\text{A-B step}, e' \in D | g_1\right] \mathbb{P}\left[\text{B-A step}, e' \in D | g_1\right] \\
&\geq \frac{1}{4} \left(\mathbb{P}_{a,b} \left[\{a g_1, a g_1 b\} \in D \right] + \mathbb{P}_{a,b} \left[\{g_1 b, a g_1 b\} \in D \right] \right) \\
&\geq \frac{1}{4} \kappa_0 \frac{d_{g_1}}{|B|}
\end{aligned}$$

Averaging over the possibilities of g_1, we get

$$\mathbb{P}\left[e' \in D\right] \geq \frac{\kappa_0}{4|B|} \cdot \frac{|D \cap E^B(g)| + |D \cap E^B(ag)}{2}$$

But by the assumption of Case 2, the RHS is at least $\frac{\kappa_0}{4|B|} \cdot \delta_B / 2$ as desired. □

Finally, using Lemma 6.6, since $\lambda_2(P) \leq \lambda$ we get

$$\frac{|T|}{|X(1)|} = \mathbb{P}[T] = \langle \mathbf{1}_T, \mathbf{1}_T \rangle \geq \frac{\kappa_0}{8} \min\{\delta_A, \delta_B\} - \lambda.$$

This finishes the proof of Proposition 6.26.

References

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. *Self-testing/correcting with applications to numerical problems*. Dec. 1993 (cit. on p. 6-4).

[BSS06] E. Ben-Sasson and M. Sudan. "Robust locally testable codes and products of codes". In: *Random Structures & Algorithms* 28.4 (2006), pp. 387–402 (cit. on p. 6-6).

[DELLM21] I. Dinur, S. Evra, R. Livne, A. Lubotzky, and S. Mozes. "Locally Testable Codes with constant rate, distance, and locality". abs/2111.04808. 2021. URL: https://arxiv.org/abs/2111.04808 (cit. on pp. 6-10, 6-11).

[DSW06] I. Dinur, M. Sudan, and A. Wigderson. "Robust Local Testability of Tensor Products of LDPC Codes". In: *APPROX*. Ed. by J. Diaz, K. Jansen, J. D. P. Rolim, and U. Zwick. Vol. 4110. Springer, 2006, pp. 304–315 (cit. on p. 6-6).

[SS96] M. Spiser and D. Spielman. *Expander Codes*. Nov. 1996 (cit. on p. 6-3).