Locally Testable Codes
with constant rate, distance, and locality

Irit Dinur1 Shai Evra2 Ron Livne2 Alexander Lubotzky1 Shahar Mozes2

1Weizmann Institute, Rehovot, Israel
2Hebrew University, Jerusalem, Israel

December 15, 2021

Abstract

A locally testable code (LTC) is an error correcting code that has a property-tester. The tester reads \( q \) bits that are randomly chosen, and rejects words with probability proportional to their distance from the code. The parameter \( q \) is called the locality of the tester.

LTCs were initially studied as important components of PCPs, and since then the topic has evolved on its own. High rate LTCs could be useful in practice: before attempting to decode a received word, one can save time by first quickly testing if it is close to the code.

An outstanding open question has been whether there exist “c\(^3\)-LTCs”, namely LTCs with constant rate, constant distance, and constant locality.

In this work we construct such codes based on a new two-dimensional complex which we call a left-right Cayley complex. This is essentially a graph which, in addition to vertices and edges, also has squares. Our codes can be viewed as a two-dimensional version of (the one-dimensional) expander codes, where the codewords are functions on the squares rather than on the edges.

1 Introduction

A locally testable code (LTC) is an error correcting code that has a property-tester. The tester reads \( q \) bits (randomly - but not necessarily uniformly chosen) from a given word, and rejects words with probability proportional to their distance from the code. The parameter \( q \) is called the locality of the tester.

A random code has, with high probability, constant rate and distance, but locality that is proportional to the length. This is true even for random LDPC codes [BHR05], and a priori the mere existence of codes with constant locality is not obvious. The first LTCs appear implicitly in works on program checking [BLR90] and on probabilistically checkable proofs (PCPs) [BFL91, LFKN92, BFLS91, AS98, ALM*98]. A formal definition of an LTC appeared simultaneously in several places [BFLS91, RS96, FS13, Aro94] (see [Gol10] for a detailed history).

Spielman, in his PhD thesis [Spi96], discusses the possibility of having an error correcting code that is locally testable (he uses the term ‘checkable code’) and explains its potential applicability: “A checker would be able to read only a constant number of bits of a received signal and then estimate the chance that a decoder will be able to correct
the errors, then the checker can instantly request a retransmission of that block, before
the decoder has wasted its time trying to decode the message. Unfortunately all known
codes with local-checkers have rate approaching zero."

Goldreich and Sudan [GS06] initiated a systematic study of LTCs as objects of in-
terest in their own right. Over the years better and better LTCs were constructed
[PS94, GS06, BSVW03, BGH+06, BS05, Din07, KMRS17, GKdO+18], but, nevertheless,
experts went back and forth on whether “c-LTCs” (namely, LTCs with constant rate,
constant distance, and constant locality) are likely to exist, compare [Gol05, Con-
jecture 3.4] with [Gol10, Section 3.3.2].

We construct the first such family of LTCs,

**Theorem 1.1.** For every $0 < r < 1$, there exist $\delta, \kappa > 0$ and $q \in \mathbb{N}$ and a polynomial-time
construction of an infinite family of error correcting codes $\{C_n\}$ with rate $r$ and distance
$\delta$, such that for all $n$, $C_n$ is $\kappa$-locally testable with $q$ queries.

Namely, every code $C_n$ comes with a randomized local tester that reads at most $q$ bits
from a given word $w$ and then accepts or rejects, such that

- For all $w \in C_n$, $\mathbb{P}[\text{accept}] = 1$.
- For all $w \notin C_n$, $\mathbb{P}[\text{reject}] \geq \kappa \cdot \text{dist}(w, C_n)$.

The parameters $\delta, \kappa$ (and $q$) depend at most polynomially (and inverse polynomially)
on $1 - r$, see Remark 5.8 for more details.

Remarkably, the theorem actually implies that for every $r, \delta$ for which error-correcting
codes are known to exist (approaching the Gilbert-Varshamov bound) there are also
LTCs. Indeed, [KMRS17, GKdO+18] have shown (see [GKdO+18, Section 1.2]) how to
take an LTC with rate arbitrarily close to 1 and with constant distance, and construct
a new LTC with rate and distance approaching the Gilbert-Varshamov bound, and only
a constant overhead in the locality $q$. So the theorem above holds for all $r, \delta > 0$ that
satisfy $r + h(\delta) < 1$ where $h(\cdot)$ is the binary entropy function.

**Expander codes, one dimension up**

The celebrated expander-codes of Sipser and Spielman [SS96] are a family of error cor-
correcting codes constructed from a single base code $C_0 \subseteq \mathbb{F}_2$ and a family of $d$-regular
expander graphs $G_n = (V_n, E_n)$ such that the code corresponding to $G_n$ consists of func-
tions on $E_n$ such that for every vertex in $V_n$, the local view at the neighboring edges
(assuming some arbitrary fixed ordering) is itself in the base code $C_0$,

$$C = \{ f : E_n \to \mathbb{F}_2 \mid \forall v \in V_n, f|_{\text{edges}(v)} \in C_0 \}.$$  

Similarly, our codes will also be defined via a fixed base-code and an infinite family of
expander graphs. Our graphs will have, in addition to vertices and edges, also two-
dimensional faces, called squares, where each square touches four edges and four vertices.

Our codewords are functions on the squares such that for every edge, the bits on the
neighboring squares form a codeword in the base code. It is natural to view our code as
a Tanner code [Tan81] with bits on the squares and constraints on the edges; whereas
the expander-codes have bits on the edges and constraints on the vertices.

Inspecting our code on the set of squares neighboring a fixed vertex, we see an
intermediate code, whose constraints come from the edges neighboring that vertex.

We thus have three codes for the three dimensions of links: the base code $C_1$ at the
link of an edge, the intermediate code $C_0$ at the link of a vertex, and the global code $C$
at the link of the empty face which is the set of all squares.
Left-Right Cayley Complex. Let us describe our construction of a graph-with-squares, namely a square complex (for a more formal description see Definition 3.1). Let $G$ be a finite group with two symmetric sets of generators $A, B$. We define the left-right Cayley complex $X = \text{Cay}^2(A, G, B)$ as follows

- The vertices are $X(0) = G$.
- The edges are $X(1) = X^A(1) \sqcup X^B(1)$ where $X^A(1) = \{ \{ g, ag \} \mid g \in G, a \in A \}$, $X^B(1) = \{ \{ g, gb \} \mid g \in G, b \in B \}$.

The fact that with $A$ we multiply on the left, and with $B$ we multiply on the right, gives a local commutativity which generates many four-cycles, namely, squares. Indeed for every $a, g, b$ the graph has a cycle of length 4 with alternating $A$ and $B$ edges, given by the walk $g, gb, agb, ag, g$. We place a square for each of these four-cycles.

- The squares are a set of the following four-cycles in the graph, $X(2) = \{ (g, gb, agb, ag, g) \mid g \in G, a \in A, b \in B \}$.

We denote by $[a, g, b]$ the square containing the edges $\{ g, ag \}$ and $\{ g, gb \}$. By changing the ‘root’ of the square we get $[a, g, b] = [a^{-1}, ag, b] = [a^{-1}, agb, b^{-1}] = [a, gb, b^{-1}]$.

The Code. Fix a left-right Cayley complex $X = \text{Cay}^2(A, G, B)$, and fix a pair of base codes $C_A \subseteq \mathbb{F}_2^A$ and $C_B \subseteq \mathbb{F}_2^B$ (assuming $|A| = |B| = d$ we can take both to be isomorphic to some $C_1 \subseteq \mathbb{F}_2^d$). Our code is defined to be

$C = C[A, G, B, C_A, C_B] = \{ f : X(2) \to \mathbb{F}_2 \mid \forall a, g, b, f([\cdot, g, \cdot]) \in C_A, \text{ and } f([a, g, \cdot]) \in C_B \}.$

Observe that for a codeword $f \in C$ and a fixed vertex $g \in G$, the restriction of $f$ to the squares touching $g$ is $f([\cdot, g, \cdot])$. It is not difficult to check that this word necessarily belongs to the tensor code $C_A \otimes C_B$, see Lemma 4.1. Thus, by putting the constraints around each edge, we get an intermediate code on the squares touching a vertex, which turns out to be a tensor code! Tensor codes have non-trivial dependencies among the constraints defining them. This often implies local testability of tensor codes [BS06, DSW06, BV09], and turns out important for showing that our code $C$ can be locally tested by the following simple test:

Local test: Choose a random vertex $g$, and accept iff $f([\cdot, g, \cdot]) \in C_A \otimes C_B$.

The construction of locally testable codes is completed by describing, in Section 6, an explicit family of groups and expanding generating sets which give expanding left-right Cayley complexes; and, in Section 5, a suitable choice of base codes $C_A, C_B$.

Let us now describe how the expansion of the complex facilitates a propagation argument for proving local testability.

Propagation from local to global. Sipser and Spielman proved distance of their expander codes [SS96] through propagation: expansion of the underlying graph is used to “lift” the distance of the base code to the distance of the global code. In our codes distance is shown similarly.
More interestingly, a similar type of argument, but more involved, serves for proving local testability as well. The local testability of the intermediate code $C_A \otimes C_B$ is lifted, via expansion, to imply local testability of the entire code, see Section 4.2.

We remark that our code has many dependencies among the constraints defining it. This is to be expected by [BGK+10]. It is the point where it most clearly differs from expander codes: in expander codes one can have a single violated constraint, which, because it is independent of all other constraints, has no further propagating effect. This leads to a word that is far from the code but cannot be detected by any tester, as proven in [BHR05].

**Locally Testable Codes: historical background and techniques**

Let us describe in some more detail the historical background pertaining to locally testable codes, including some works that were already mentioned earlier.

The study of LTCs arose naturally in works on program checking and PCPs. The Hadamard code was the first code proven to be locally testable in the work of Blum, Luby, and Rubinfeld on linearity testing [BLR90]. The low (logarithmic) rate of this code was quickly improved to polynomial rate by moving from linear functions (codewords of the Hadamard code) to low degree polynomial functions (codewords of the Reed-Muller code). Subsequent works studied “low degree tests” which are in fact proofs that the Reed-Muller code is locally testable. These works were crucial for progress leading up to the proof of the PCP theorem. More on the relation between PCPs and LTCs, as well as the historical development, can be found in Goldreich’s survey [Gol10].

A systematic study of LTCs was initiated by Goldreich and Sudan in [GS06], and a sequence of works constructed both LTCs and PCPs with improved parameters [GS06, BSVW03, BGH+06, BS08, Din07], achieving constant locality and distance, but rate $1/\text{poly log } n$. Some experts believed that low rate is inherently needed and some attempts to prove upper bounds on the rate have been made [BGK+10, DK11, BV12, BSS05], although these lower bounds are in rather restrictive models.

This, perhaps, has triggered works from the other end of the spectrum [KMRS17, GKdO+18] which focused on constructing error correcting codes with constant rate and distance, that are locally testable with smallest possible locality. These works achieve constant rate and quasi-poly-logarithmic distance and locality.

In terms of techniques, many of the earlier constructions of LTCs have two notable features. Firstly, they are based on the properties of low degree polynomials, and secondly, they come hand in hand with PCP constructions, so that both share the same composition-recursion structure.

The gap amplification technique [Din07] of the first author is a construction of both a PCP and an LTC that relies on expander graphs and concatenation and departs from the domain of low degree polynomials. Meir [Mei08] gave a tensor-code-based construction of LTCs that is neither related to low degree functions nor to PCPs altogether. Further works [KMRS17, GKdO+18] also construct LTCs without any PCP counterpart.

A feature shared by all previous constructions of LTCs with mildly high rate is their recursive nature. One first constructs codes with weaker properties and then enhances them by concatenation, possibly with different iterations. The overall composed structure of the code is somewhat complicated and begs for a more direct “one-shot” construction.

A path leading towards a one-shot construction seemed to open up with the connection to high dimensional expanders.
High Dimensional Expansion

The current paper is mainly elementary and almost self-contained (with the exception of Section 6 which uses the existence of some Ramanujan Cayley graphs with specific properties and can be taken as a black box). But it came up as a result of a much longer and intensive journey. Some interesting open problems were left aside along the way. It is, therefore, worthwhile to give the story here.

The journey started by the first and fourth authors during a year-long program at the IIAS (Israeli Institute of Advanced Studies) on high dimensional expanders in 2017: the hope was to use the Ramanujan complexes (à la [LSV05a, LSV05b]) to construct LTCs as high-dimensional versions of expander-codes over Ramanujan graphs as explained above. Although expander codes are typically not locally testable [BHR05] the hope was that higher dimensional versions would be.

This optimistic belief was inspired by local to global behavior of certain high dimensional complexes that was uncovered already by Garland in his seminal work [Gar73].

In that paper, Garland proved a conjecture of Serre, that the cohomology of cocompact lattices in high-rank simple $p$-adic groups vanishes. Equivalently, if $X$ is a finite simplicial quotient of a Bruhat-Tits building of dimension at least two, its cohomology vanishes. The proof of Garland is “local-to-global”: he showed that if the links of relevant cells have a spectral gap, then so does the global Laplacian of $X$. Namely, if $X$ is locally an expander, then it is also globally so. (For a purely combinatorial treatment and generalizations - see [Opp18]). The global spectral gap implies the vanishing of the cohomology.

This “local to global” approach is a high-dimensional phenomenon that does not hold for graphs! In graphs, the local structure does not reveal any information about the global expansion. To illustrate this, the reader may recall the LPS-Ramanujan graphs [LPS88] which are (p+1)-regular expander graphs with large girth. One can easily get (p+1)-regular graphs with large girth (and hence locally isomorphic to the LPS ones) which are far from being expanders. In contrast, the Garland method shows that local expansion implies global expansion in the high dimensional case.

The local to global approach was also the key ingredient, in [KKL14, EK16] where Gromov’s overlapping problem was solved using the Ramanujan complexes.

At this point there was already some interest from the theoretical computer science community. The fact that high dimensional expansion is related to property testing in computer science was observed for the first time by Kaufman and the fourth author [KL14]. The first author and Kaufman proved that high dimensional expansion implies an efficient agreement-test [DK17], which is related to both PCPs and LTCs. Anari et al [ALOV19] resolved a conjecture regarding convergence of certain Markov chains by analyzing the global random walk through local analysis at the links.

Inspired by all this, the idea was to construct LTC codes by using the local-to-global behavior of the Ramanujan complexes in an analog to the way [SS96] used Ramanujan graphs for LDPC codes. For simplicity, we will describe it from now on only in dimension 2, but one can do the same in higher dimensions.

The original idea was as follows: fix a large prime $p$ and take an infinite family of Ramanujan complexes $X$, quotients of the Bruhat-Tits building of $G = SL(3, \mathbb{Q}_p)$. The complex $X$ is a 2-dimensional complex, the link of every edge of it is in one-to-one correspondence with the projective line $\mathbb{P}^1$ over $\mathbb{F}_p$, and the link of every vertex is the graph of lines versus points of the projective plane over $\mathbb{F}_p$. One can define a base code (“the small code”) $C_1$ on $\mathbb{P}^1$ to be a 'projective' variant of the Reed-Solomon code. This code induces a 'big code' $C$ as a subspace of the $\mathbb{F}_p$ functions on $X(2)$- the 2-dimensional cells of $X$- whose local views at every edge are in the base code of the edge. The goal
was then to propagate the rate, distance, and local testability of Reed-Solomon codes from the small code $C_1$ to the big code $C$.

This turned out to be easier to say than to do. At some point, we were hoping to use $p$-adic uniformization. Recall the work of Mumford [Mum79] who used the combinatorial structure of one such Ramanujan complex to prove a result on algebraic surfaces appearing as locally symmetric quotients of $SU(2,1)$. We were hoping to go in the opposite direction and to use the theory of algebraic surfaces to study our combinatorial objects. The theory of $p$-adic uniformization was developed in depth by Varshavsky in his thesis [Var98] (written under the supervision of the 3rd author of the current paper). This is an opportunity to thank Yakov Varshavsky who gave upon our request a semester-long course describing this work. While we eventually are not using this, we were fortunate to be exposed to an amazing chapter of deep mathematics.

Propagating local testability from the small code to the big code when these are defined over a high dimensional expander is possible. This was proved in [DDFH18] with the hope that it would serve our original plan. For our codes to fit, the intermediate code, $C_0$ - the one that is defined on the link of a vertex through the small Reed-Solomon codes $C_1$ on the edges - needed to be itself locally testable. Unfortunately we failed to prove that $C_0$ is locally testable. Here the problem is very concrete: Find $C_1$ inside $\mathbb{F}_p^n$ such that the induced intermediate code $C_0$ on the link of a vertex is locally testable. Here, the link of a vertex is nothing but the lines versus points graph of the projective plane.

One can generalize this challenge to get such a code also on higher dimensional spherical buildings. Are such spherical codes locally testable?

We, therefore, changed direction and replaced $G = SL(3,\mathbb{Q}_p)$ by a product $G = SL(2,\mathbb{Q}_p) \times SL(2,\mathbb{Q}_q)$. This time the quotients obtained from congruence lattices in $G$ give rise to square complexes. These complexes were shown long ago to be Ramanujan cubical complexes [JL99] and the dynamic of walks along them was studied in [Moz91]. This time the local intermediate code is a tensor code (since the link of every vertex is the complete bipartite graph) and there are plenty of tensor codes that are locally testable as mentioned above. A subtle obstacle arose at this point which does not exist in the graph codes of [SS96]: one needs to name the squares in such a way that the function defined on the link of an edge $\{u,v\}$ will be in or out the code independently if we look at it from the vertex $u$ or the vertex $v$. It might be that this challenge can be overcome, but at that point, we realized that by changing from these square complexes to the left-right Cayley complexes as defined above, this problem is easily fixed. Moreover, it became also easier to argue about the rate- making the whole paper much simpler than we expected!

As explained, our long journey left a number of unsettled issues. We believe they are interesting in their own right (and in all dimensions) even if not needed anymore for the concrete goal of locally testable codes.

The left-right Cayley complexes seem to be objects that are worth studying for their own sake. It is actually somewhat surprising that in spite of over 100 years of studying Cayley graphs, these objects, as far as we know, have never been studied before (but see [BE21], about which we learned only after writing this paper). An immediate curiosity is whether there are higher-dimensional analogs or whether a group “has only two sides” and hence these exist only in dimension 2. Anyway, it seems that this paper solves one problem but opens many others.

After this work was completed and announced, we have learned about related developments which have happened independently in the last few months.
As part of an ongoing effort to build a good LDPC quantum error correcting code \(qECC\) by homological methods (see \[BE21, PK21a\]), Breuckmann and Eberhardt \[BE21\], defined a “balanced product of \(G\)-graphs”. When specialized to Cayley graphs, this gives the Left/Right Cayley complexes defined in Definition 3.1 below. Using these, some quantum error correcting codes are defined as chain complexes of length 3. Cutting them to length 2 gives classical codes. In retrospect, one can see that our codes were hidden there (but without the LTC property).

Even more recently, Pantaleev and Kalachev \[PK21b\] announced solutions for both problems: good quantum LDPC codes as well as good classical locally testable codes (with rate up to 1/2).

## 2 Preliminaries

### 2.1 Expander Graphs

A \(d\)-regular graph \(G\) is said to be a \(\lambda\)-one-sided expander if it has eigenvalues \(d = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq -d\) which satisfy \(\lambda_i \leq \lambda \cdot d\) for all \(i > 1\).

The following is a standard lemma by Alon and Chung,

**Lemma 2.1** \([AC88]\). Let \(G = (V, E)\) be a \(d\)-regular \(\lambda\)-one-sided expander. Let \(T \subseteq V\) be such that the graph induced on \(T\), denoted \(G(T)\), has average degree at least \(\delta d\). Then \(|T| \geq (\delta - \lambda) \cdot |V|\), and the number of edges in \(G(T)\) is at least \((\delta - \lambda)\delta \cdot |E|\).

This lemma holds in more general situations where instead of a \(d\)-regular graph we have a weighted Markov operator as long as it has an orthonormal basis of eigenvectors. Let \(\mathcal{D}\) be any probability distribution over a finite set \(V\), and define an inner product by

\[
\langle \cdot, \cdot \rangle_{\mathcal{D}} : \mathbb{R}^V \times \mathbb{R}^V \rightarrow \mathbb{R}, \quad \langle f, f' \rangle_{\mathcal{D}} = \mathbb{E}_{x \sim \mathcal{D}} [f(x)f'(x)].
\]

Let \(1_T \in \mathbb{R}^V\) be the indicator function of a set \(T \subseteq V\). We have \(\langle 1_T, 1_T \rangle_{\mathcal{D}} = \mathbb{P}_\mathcal{D}[T]\), and moreover the probability, with respect to \(\mathcal{D}\), that a random walk described by \(M\) starts at \(T\) and after one step still stays in \(T\) is \(\langle 1_T, M1_T \rangle_{\mathcal{D}}\).

Denote by \(1 \in \mathbb{R}^V\) the constant 1 function.

**Lemma 2.2.** Let \(M : \mathbb{R}^V \rightarrow \mathbb{R}^V\) be a symmetric Markov operator such that \(M1 = 1\), and such that for all \(h\) with \(\langle h, 1 \rangle_{\mathcal{D}} = 0\), \(\langle h, Mh \rangle_{\mathcal{D}} \leq \lambda \langle h, h \rangle_{\mathcal{D}}\).

Let \(f = 1_T\) be the indicator of a set \(T \subseteq V\). If \(\langle f, Mf \rangle_{\mathcal{D}} \geq \delta \cdot \langle f, f \rangle_{\mathcal{D}}\) then \(\mathbb{P}_\mathcal{D}[T] \geq \delta - \lambda\), and \(\langle f, Mf \rangle_{\mathcal{D}} \geq \delta (\delta - \lambda)\).

**Proof.** Denote \(p = \mathbb{P}_\mathcal{D}[T]\). We can write \(f = p1 + h\) with \(\langle h, 1 \rangle_{\mathcal{D}} = 0\). We get

\[
\delta \cdot p \leq \langle f, Mf \rangle_{\mathcal{D}} = \langle p1 + h, M(p1 + h) \rangle_{\mathcal{D}} \leq p^2 + \lambda \langle h, h \rangle_{\mathcal{D}} \leq p^2 + \lambda p.
\]

where the last inequality is because \(\langle h, h \rangle_{\mathcal{D}} \leq \langle f, f \rangle_{\mathcal{D}} = p\). When rearranging, this gives the lemma. \(\square\)

### 2.2 Error Correcting Codes

A linear code \(C \subseteq \mathbb{F}_2^n\) is an \(\mathbb{F}_2\)-linear subspace of \(\mathbb{F}_2^n\). The block-length of the code is \(n\). The rate and distance of the code are the relative dimension of the code and relative Hamming weight of the smallest weight non-zero codeword, respectively, namely,

\[
\text{Rate}(C) = \frac{1}{n} \dim(C) \quad \text{and} \quad \text{dist}(C) = \frac{1}{n} \min_{w \in C \setminus \{0\}} |\{i \in [n] | w_i \neq 0\}|.
\]
We recall the definition of locally testable codes from [GS06]. The definition given here is that of a “strong” LTC, and implies all other definitions of locally testable codes. See [Gol17, Chapter 13].

**Definition 2.3** (Locally Testable Code (LTC)). For \(\kappa > 0\) and \(q \in \mathbb{N}\) we say that an error correcting code \(C \subseteq \mathbb{F}_2^n\) is \(\kappa\)-locally testable with \(q\) queries if there is a distribution over a collection of \(q\)-element subsets \(S \subseteq [n]\) such that each subset \(S\) is associated with a set \(V_S \subseteq \mathbb{F}_2^n\) of allowed local views, and such that, denoting by \(f|_S\) the restriction of \(f\) to the set \(S\), the following hold:

- If \(f \in C\) then for every \(S, f|_S \in V_S\).
- For every \(f \in \mathbb{F}_2^n\), the rejection probability

\[
P_S[f|_S \notin V_S] \geq \kappa \cdot \text{dist}(f, C).
\]

The parameter \(\kappa\) is called the detection probability.

**Definition 2.4** (Tensor Code). Let \(n_1, n_2 \in \mathbb{N}\) and let \(C_i \subseteq \{ f : [n_i] \to \mathbb{F}_2 \}\) for \(i = 1, 2\) be two linear codes. Define the tensor code \(C = C_1 \otimes C_2\) by

\[C = \{ M : [n_1] \times [n_2] \to \mathbb{F}_2 \mid \forall i \in [n_1], j \in [n_2], M(i, \cdot) \subseteq C_2, M(\cdot, j) \subseteq C_1 \}.\]

It is easy to check that \(\dim(C_1 \otimes C_2) = \dim(C_1) \cdot \dim(C_2)\), and that \(\text{dist}(C_1 \otimes C_2) = \text{dist}(C_1) \cdot \text{dist}(C_2)\). We view the elements of \(C\) as \(n_1\)-by-\(n_2\) matrices \(w\) and write \(w(i, \cdot) \in \mathbb{F}_2^{n_2}\) for the \(i\)-th row of \(w\), and similarly \(w(\cdot, j) \in \mathbb{F}_2^{n_1}\) is the \(j\)-th column of \(w\).

A natural test for whether a given matrix \(f \in \mathbb{F}_2^{n_1 \times n_2}\) is in \(C_1 \otimes C_2\) is as follows:

Randomly choose a row or a column, and check whether the restriction of \(f\) to that column (or row) is in \(C_1\) (or \(C_2\)).

The quality of the test is measured by the relation between the rejection probability and the distance of \(f\) from the tensor code. Formally, this is captured by the notion of robust testability.

**Definition 2.5** (Robust testability of tensor codes). Fix \(C_i \subseteq \mathbb{F}_2^{n_i}\) linear error correcting codes, for \(i = 1, 2\). For \(f : [n_1] \times [n_2] \to \mathbb{F}_2\), let

\[\delta_{\text{col}}(f) = \text{dist}(f, C_1 \otimes \mathbb{F}_2^{n_2}), \quad \delta_{\text{row}}(f) = \text{dist}(f, \mathbb{F}_2^{n_1} \otimes C_2).\]

and

\[\delta(f) = (\delta_{\text{col}}(f) + \delta_{\text{row}}(f))/2.\]

The robust testability of \(C_1 \otimes C_2\) is defined to be

\[\tau = \min_{f \neq C_1 \otimes C_2} \frac{\delta(f)}{\text{dist}(f, C_1 \otimes C_2)},\]

and we say that \(C_1 \otimes C_2\) is \(\tau\)-robustly testable.

The robust testability of tensor codes was first studied in [BS06], where it was shown that for any code \(C\) with sufficiently high distance, the \(d\)-dimensional tensor code \(C^{\otimes d}\) is robustly testable for all \(d \geq 3\). The requirement \(d \geq 3\) was puzzling because the tensor of Reed-Solomon codes is known [PS94] to be robustly testable even for \(d = 2\) and this was considered the prototype for locally testable codes. Surprisingly, Paul Valiant discovered [Val05] that there are codes \(C\) for which \(C \otimes C\) is not robustly testable, see also [GM12]. Quickly after that [DSW06] formulated a notion of smooth codes, broadened later to ‘weakly smooth’ in [BV09], and showed that the tensor product of a smooth code and any other code is in fact robustly testable. To define smooth codes recall the definition of LDPC codes,
Definition 2.6 (LDPC code). 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

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

Definition 2.7 (Smooth code). Let \(c, d, n \in \mathbb{N}, \alpha, \beta, \delta > 0\). A \((c, d, n)\)-LDPC code \(C \subset \mathbb{F}_2^n\) is \((\alpha, \beta, \delta)\)-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 \(\delta\), 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 in \(X\).

Random low density parity check codes (LDPC) are smooth, see Section 5.

Agreement Testability. A related testing notion focuses on the agreement between pairs of overlapping local views. We think of the following situation,

- For each column we are given a codeword of \(C_1\), and these are aggregated into \(w_1 \in C_1 \otimes \mathbb{F}_2^n\).

- For each row we are given a codeword of \(C_2\), and these are aggregated into \(w_2 \in \mathbb{F}_2^n \otimes C_2\).

- We check “agreement”, namely, pick a random pair of row \(i\) and column \(j\), and check whether they agree on their intersection, i.e. whether

\[
w_1(i, j) \equiv w_2(i, j).
\]

Agreement testability is defined to be the ratio between the amount of pairwise disagreement to the distance from the code \(C_1 \otimes C_2\). Formally,

Definition 2.8 (agreement testability). Let \(\kappa > 0\). Let \(C_i \subseteq \{ f : [n_i] \rightarrow \mathbb{F}_2 \}\) for \(i = 1, 2\). We say that \(C_1 \otimes C_2\) is \(\kappa\)-agreement testable if for every \(w_1 \in C_1 \otimes \mathbb{F}_2^n\), \(w_2 \in \mathbb{F}_2^n \otimes C_2\), there exists \(w \in C_1 \otimes C_2\) such that

\[
\kappa \cdot \left( \sum_{i \in [n_1], j \in [n_2]} \mathbb{P}[w_1(i, \cdot) \neq w_2(i, \cdot)] \right) \leq \mathbb{P}[w_1 \neq w_2].
\]

In words, given a word \(w_1\) whose rows are in \(C_1\), and given a word \(w_2\) whose columns are in \(C_2\), we say that \(C_1 \otimes C_2\) is \(\kappa\)-agreement testable if the amount of disagreement between \(w_1\) and \(w_2\) is an upper-bound for the fraction of rows or columns one needs to change in order to get to the closest word \(w \in C_1 \otimes C_2\), times \(\kappa\).

It is well known (see for example [DH09]) that agreement testability is equivalent to robust testability:

Lemma 2.9. Let \(C_i \subseteq \mathbb{F}_2^n\), and assume \(\delta_i = \text{dist}(C_i)\) for \(i = 1, 2\).

- If \(C_1 \otimes C_2\) is \(\tau\)-robustly testable then it is \(\kappa\)-agreement testable, for \(\kappa^{-1} = \frac{1}{2\delta_1} + \frac{1 + 1/(2\tau)}{\delta_2}\).

- If \(C_1 \otimes C_2\) is \(\kappa\)-agreement testable, then it is \(\tau\)-robustly testable for \(\tau = \frac{\kappa}{2(\kappa + 1)}\).

We prove this lemma in Appendix A.
3 The Left-Right Cayley Complex

We describe a new construction of a Cayley graph that in addition to vertices and edges also has two-dimensional faces, called squares. Each square contains four edges that constitute a four-cycle.

**Definition 3.1** (Left-Right Cayley Complex). Let $G$ be a group with two symmetric sets of generators $A, B$, namely, each is closed under taking inverses. We assume that the identity element of $G$ is neither in $A$ nor in $B$. Define the **Left-Right Cayley Complex** $X = \text{Cay}^2(A, G, B)$ as follows

- The vertices are $X(0) = G$.
- The edges are $X(1) = X^A(1) \cup X^B(1)$ where
  \[
  X^A(1) = \{(g, ag) \mid g \in G, a \in A\}, \quad X^B(1) = \{(g, gb) \mid g \in G, b \in B\}.
  \]
- The squares are $X(2) = A \times G \times B/\sim$ where for every $a \in A, b \in B, g \in G$,
  \[
  (a, g, b) \sim (a^{-1}, ag, b) \sim (a^{-1}, agb, b^{-1}) \sim (a, gb, b^{-1}),
  \]
  and denote the equivalence class of $(a, g, b)$ by $[a, g, b]$, so
  \[
  [a, g, b] = \{(a, g, b), (a^{-1}, ag, b), (a^{-1}, agb, b^{-1}), (a, gb, b^{-1})\}.
  \]

The graph $(X(0), X^A(1))$ is none other than the Cayley graph $\text{Cay}(G, A)$. Similarly $(X(0), X^B(1))$ is the Cayley graph $\text{Cay}(G, B)$. The fact that with $A$ we multiply on the left, and with $B$ we multiply on the right, gives a local commutativity which generates many four-cycles, namely, squares.

**Remark 3.2.** Given a group $G$ and a set of generators $A$, the Cayley graph $\text{Cay}^{left}(G, A)$ with left-multiplication edges is isomorphic to the Cayley graph $\text{Cay}^{right}(G, A)$ with right multiplication edges via the map $g \mapsto g^{-1}$. The left-multiplication edge $\{g, ag\}$ maps to the right multiplication edge $\{g^{-1}, g^{-1}a^{-1}\}$. This justifies talking about a Cayley graph without specifying left or right multiplication.

**Remark 3.3.** The product of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a square complex $X = G_1 \times G_2$ defined as follows.

- The vertices are $X(0) = V_1 \times V_2$.
- The edges are $X(1) = E_1 \times V_2 \sqcup V_1 \times E_2$, where an edge $\{(u, u'), v\} \in E_1 \times V_2$ connects $(u, v)$ with $(u', v)$, and similarly an edge $\{(v, v'), u\} \in V_1 \times E_2$ connects $(u, v)$ with $(u', v')$.
- The squares $X(2)$ are identified with $E_1 \times E_2$, so that the square corresponding to the pair of edges $e_1 = \{u, u'\} \in E_1$ and $e_2 = \{v, v'\} \in E_2$ is the four-cycle $(u, v) \rightarrow (u, v') \rightarrow (u', v') \rightarrow (u', v) \rightarrow (u, v)$.

The left-right Cayley complex is the quotient of the Cartesian product of $G_A = (G, X^A(1))$ and $G_B = (G, X^B(1))$ obtained by identifying the vertex $(g, g')$ with $(gh^{-1}, hg')$ for all $h \in G$. One can check that the map $(g, g') \mapsto gg'$ gives a homomorphism of graphs from $G_A \times G_B$ to $\text{Cay}^2(A, G, B)$. 


Remark 3.4. Left-right Cayley complexes are examples of two-dimensional cubical complexes. Cubical complexes are well-studied, and in particular there are constructions of Ramanujan cubical complexes [JL99] with bounded degree and any dimension, whose walk dynamics was studied in [Moz91]. The left-right Cayley complexes have an additional matching labels feature that other complexes are not known to have.

Definition 3.5 (Links). For each \( g \in G \), the link of \( g \) is \( X_g = \{ [a, g, b] \mid a \in A, b \in B \} \). There is a natural map \( (a, b) \mapsto [a, g, b] \).

For every edge \( e = \{ g, ag \} \), the link of \( e \) is denoted \( X_e = \{ [a, g, b] \mid b \in B \} \). Similarly, if \( e = \{ g, gb \} \) we let \( X_e = \{ [a, g, b] \mid a \in A \} \).

Definition 3.6. A left-right Cayley complex satisfies the total no-conjugacy condition if

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

Here are a few easy properties of left-right Cayley complexes.

Claim 3.7. Assuming (TNC), each vertex has exactly \( |A| + |B| \) distinct neighbors; and each square contains exactly four distinct vertices; and the map \( (a, b) \mapsto [a, g, b] \) is a bijection from \( A \times B \) to \( X_g \) for each \( g \in G \).

Proof. Let \( a \neq a' \in A \) and \( b \neq b' \in B \). Clearly \( ag \neq a'g \) and \( gb \neq gb' \). If \( ag = gb \) then \( g^{-1}ag = b \) which contradicts (TNC). So \( g \) has \( |A| + |B| \) distinct neighbors. Next we show that each square \([a, g, b] \in X(2)\) must have four distinct vertices. As \( 1 \notin A \cup B \), \( g \neq ag \) and \( g \neq gb \), and we already saw that \( ag \neq gb \). Now, if \( g = agb \) we would contradict (TNC) because it implies \( g^{-1}a^{-1}g = b \) making \( a^{-1} \in A \) and \( b \in B \) conjugates.

Finally, let us see that the map \( (a, b) \mapsto [a, g, b] \) is a bijection between \( A \times B \) and \( X_g \) for all \( g \). Assume that \([a, g, b] = [a', g, b']\) for some \((a, b), (a', b') \in A \times B\). This implies that \((a', g, b') \in \{(a, g, b), (a^{-1}, ag, b), (a^{-1}, agb, b^{-1}), (a, gb, b^{-1})\}\). We have seen that \( g \neq ag, gb, agb \) hence \((a, g, b) = (a', g, b')\) which means that \((a, b) = (a', b')\). \(\Box\)

Remark 3.8. It follows that assuming (TNC)

\[
|X(1)| = \frac{|A| + |B|}{2} \cdot |G| \quad \text{and} \quad |X(2)| = \frac{|A||B|}{4} \cdot |G|.
\]

It will be useful to consider a weighted version of the 1-skeleton of \( X \), where the weight is distributed evenly between the \( A \) and the \( B \) edges. When \(|A| = |B|\) this is the usual unweighted graph.

Definition 3.9. Let \( \mathcal{D}_1 \) be the distribution over \( X(1) \) given by selecting with probability half a uniform edge in \( X^A(1) \), and with probability half a uniform edge in \( X^B(1) \). (In case \( A, B \) have equal size \( \mathcal{D}_1 \) is the uniform distribution over \( X(1) \).)

We define an inner product on functions over \( X(1) \). Let \( f, f' : X(1) \to \mathbb{R} \) and define

\[
\langle f, f' \rangle_{\mathcal{D}_1} = \mathbb{E}_{e \sim \mathcal{D}_1} \left[ f(e) f'(e) \right] = \frac{1}{2} \mathbb{E}_{e \in X^A(1)} \left[ f(e) f'(e) \right] + \frac{1}{2} \mathbb{E}_{e \in X^B(1)} \left[ f(e) f'(e) \right]. \tag{3.1}
\]

This will be the only inner product we consider for functions over \( X(1) \) so we sometimes omit the subscript and simply write \( \langle f, f' \rangle = \langle f, f' \rangle_{\mathcal{D}_1} \). As usual we let \( \|f\| = \langle f, f \rangle \).
**Parallel Random Walk.** In addition to the standard random walk on the 1-skeleton of $X$, we will be interested in a random walk on the edges called the parallel walk, which takes an edge $e$ to a random edge $e'$ that is “parallel” to it.

**Definition 3.10 (Labels).** For each $s \in A \cup B$ let $[s] = \{s, s^{-1}\}$. Let $\hat{A} = \{[a] \mid a \in A\}$ and let $\hat{B} = \{[b] \mid b \in B\}$. The label of an edge $\{g, ag\}$ is defined to be $[a] = \{a, a^{-1}\}$, and this is independent of the presentation of the edge as $\{g, ag\}$ or $\{(ag), a^{-1}(ag)\}$. Similarly, the label of an edge $\{g, gb\}$ is defined to be $[b] = \{b, b^{-1}\}$.

Let $\hat{A} \cup \hat{B}$ denote the set of labels of the edges in the complex. For any $\sigma \in \hat{A} \cup \hat{B}$, denote by $X^\sigma(1)$ the set of edges labelled $\sigma$.

**Claim 3.11.** If $\sigma = \{c, c^{-1}\} \in \hat{A} \cup \hat{B}$ and $c \neq c^{-1}$, then $X^\sigma(1)$ has size $|G|$, otherwise it has size $|G|/2$.

**Proof.** We shall prove the claim for $\sigma = \{a, a^{-1}\} \in \hat{A}$, the claim for $\sigma \in \hat{B}$ is proven analogously. Observe that every vertex $g$ participates in two edges labelled $\sigma = \{a, a^{-1}\}$, namely $\{g, ag\}$ and $\{g, a^{-1}g\}$. Since every edge is counted twice, from each of its two endpoints, we get that $|\hat{A}| = |X^\sigma(1)|$.

In case $a = a^{-1}$ each vertex participates in only a single edge labelled $[a]$, but still every edge has two endpoints so after accounting for the double counting we get $|X^\sigma(1)| = |\hat{A}|/2$.

Let us define a Markov operator $M[\sigma] : \mathbb{R}^{X^\sigma(1)} \to \mathbb{R}^{X^\sigma(1)}$ on the space of functions on $X^\sigma(1)$. Fix $f : X^\sigma(1) \to \mathbb{R}$. If $\sigma = [a] \in \hat{A}$ we set

$$M[\sigma]f(\{g, ag\}) = \mathbb{E}_b f(\{gb, agb\}),$$

and if $\sigma = [b] \in \hat{B}$ we set

$$M[\sigma]f(\{g, gb\}) = \mathbb{E}_a f(\{ag, ag\}).$$

Note that when $a \neq a^{-1}$, the operator $M[\sigma]$ on $X[\sigma](1)$ is isomorphic to the standard random walk on $Cay(G, B)$ by sending the edge $\{g, ag\}$ to the vertex $g$. Similarly if $b \neq b^{-1}$ then $M[\sigma]$ is isomorphic to the random walk on $Cay(G, A)$.

We define a Markov operator $M[1] : \mathbb{R}^{X(1)} \to \mathbb{R}^{X(1)}$ on the space of functions on the entire set of edges $X(1)$ by letting, for any $f : X(1) \to \mathbb{R}$,

$$M[1]f = \sum_{\sigma} M[\sigma]f|_{X^\sigma(1)}.$$

**Definition 3.12 (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 $e$. (If (TNC) does not hold the square might not contain an edge $e' \neq e$ with the same label, in which case the walk will stay in place).

The Markov operator corresponding to this walk is exactly $M[1]$, because starting at an edge $e = \{g, ag\}$, a random square containing $e$ is $[a, g, b]$ for a uniformly chosen $b \in B$, and then the only other $[a]$-labeled edge in this square is the edge $e' = \{gb, ag\}$.

**Lemma 3.13.** Assume both $Cay(G, A)$ and $Cay(G, B)$ are $\lambda$-expanders. Suppose $R \subseteq X(1)$ and assume $f = \mathbb{1}_R : X(1) \to \mathbb{R}$ satisfies $(f, M[1]f) \geq c \cdot (f, f)$. Then there exists some $\sigma \in \hat{A} \cup \hat{B}$ such that

$$|R \cap X^\sigma(1)| \geq (c - \lambda)|G|/2.$$
Proof. The weight of $R$ is given by $\langle f, f \rangle$, and the probability to start in $R$ and stay in $R$ after one step of the parallel random walk is $\langle f, M^{|f|} f \rangle$. We expand $\langle f, M^{|f|} f \rangle$ according to (3.2), and get

$$\langle f, M^{|f|} f \rangle = \mathbb{E}_{\sigma \in X^*(1)} [f(e) M^{|f|}_\sigma f(e)],$$

where the expectation over $\sigma$ is obtained by choosing, with probability half, a random $a \in A$ and outputting $[a]$; and with probability half, a random $b \in B$ and outputting $[b]$. Clearly then

$$\langle f, f \rangle = \mathbb{E}_{e \sim D_1} [f(e)^2] = \mathbb{E}_{\sigma \in X^*(1)} [f(e)^2].$$

Plugging these into the inequality $\langle f, M^{|f|} f \rangle - c \cdot \langle f, f \rangle \geq 0$ we get

$$\mathbb{E}_{\sigma \in X^*(1)} \left[ f(e) M^{|f|}_\sigma f(e) - c \cdot f(e)^2 \right] \geq 0$$

so there must be at least one $\sigma$ for which

$$\mathbb{E}_{\sigma \in X^*(1)} [f(e) M^{|f|}_\sigma f(e)] \geq c \cdot \mathbb{E}_{\sigma \in X^*(1)} [f(e)^2]. \quad (3.3)$$

Fix, say, $\sigma = [a]$ and define $h_a : G \to \mathbb{R}$ by $h_a(g) = f([a, ag])$. (The case $\sigma = [b]$ is analogous and omitted). Now,

$$c \cdot \langle h_a, h_a \rangle = c \cdot \mathbb{E}_g [h_a(g)^2] = c \cdot \mathbb{E}_g [f([a, ag])^2] = c \cdot \mathbb{E}_{e \in X^{[a]}(1)} [f(e)^2] \overset{(3.3)}{\leq} \mathbb{E}_{e \in X^{[a]}(1)} [f(e) M^{|f|}_\sigma f(e)]$$

$$= \mathbb{E}_{g \in G} [f([g, ag])] \mathbb{E}_{b \in B} [f([gb, agb])] = \mathbb{E}_{g \in G} \left[ h_a(g) \mathbb{E}_{b \in B} h_a(gb) \right] = \langle h_a, M_B h_a \rangle, \quad (3.4)$$

where $M_B$ is the random walk operator on $\text{Cay}(G, B)$. We relied here on the fact that choosing a uniform edge in $X^{[a]}(1)$ can be done by choosing a uniform $g \in G$ and looking at $[g, ag]$. Observe now that $h_a$ indicates the set $T = \{ g \in G \mid f([g, ag]) \neq 0 \}$, so by Lemma 2.2 applied on the graph $\text{Cay}(G, B)$ with the operator $M_B$ we deduce that $|T| \gg (c - \lambda)|G|$. Since every non-zero value for $f$ can cause at most two non-zero values in $h_a$, we get that $|R \cap X^*(1)| = |f^{-1}(1) \cap X^*(1)| \gg |h_a^{-1}(1)|/2 = |T|/2 \gg (c - \lambda)|G|/2$. \hfill \Box

4 Error Correcting Code on a Left-Right Cayley Complex

Let $G$, $A$, $B$ and $X = \text{Cay}^2(G, A, B)$ as in the previous section. Recall that for any vertex $g \in X(0)$ (resp. any edge $e \in X(1)$) we denote by $X_g \subset X(2)$ (resp. $X_e \subset X(2)$) the set of squares in $X$ containing the vertex $g$ (resp. the edge $e$). Let $C_A \subset \mathbb{F}_2^A$ and let $C_B \subset \mathbb{F}_2^B$ be two fixed linear error correcting codes with rates $\rho_A = \text{Rate}(C_A), \rho_B = \text{Rate}(C_B)$ and distances $\delta_A = \text{dist}(C_A), \delta_B = \text{dist}(C_B)$, respectively.

Define the code $C = [G, A, B, C_A, C_B]$ as follows. For an edge $e = [g, ag] \in X^A(1)$, we define a local code

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

Similarly, for an edge $e = [g, gb] \in X^B(1)$, we define a local code

$$C_e = \{ f : X_e \to \mathbb{F}_2 \mid f([\cdot, g, b]) \in C_A \}.$$
Note that this definition appears to depend on the choice of \( g \in e \) but it does not. Finally, we define a global code
\[
C = \{ f : X(2) \to \mathbb{F}_2 | \forall e \in X(1), f|_{X_e} \in C_e \}.
\]
For each vertex \( g \in X(0) \), define the local tensor code around the vertex \( g \) to be
\[
C_g = \{ f : X_g \to \mathbb{F}_2 | f(\cdot, g, \cdot) \in C_A \otimes C_B \}.
\]

**Lemma 4.1** (\( C \) is a lifted tensor-code).
\[
C = \{ f : X(2) \to \mathbb{F}_2 | \forall g \in X(0), f|_{X_g} \in C_g \}.
\]

**Proof.** Immediate from the fact that \( f(\cdot, g, \cdot) \in C_A \otimes C_B \) for any \( g \in X(0) \) if and only if \( f(\cdot, g, \cdot) \in C_B \) and \( f(\cdot, g, \cdot) \in C_A \) for any \( g \in X(0) \), \( a \in A \) and \( b \in B \). □

Observe that for the local code at each vertex to be a tensor code, we must make sure that around every \( A \) edge we have the same code \( C_A \), and similarly for \( B \). If we choose different base codes at different edges we might still get a code with rate and distance, but local testability will probably fail, because we lose the local tensor structure. This is in contrast to the case of expander codes where the local base code can be chosen arbitrarily and differently at each vertex.

### 4.1 Properties of The Code

We now look at the rate, distance and local testability of the code \( C = C[G, A, B, C_A, C_B] \). Recall \( \rho_A = \text{Rate}(C_A) \), \( \rho_B = \text{Rate}(C_B) \) and \( \delta_A = \text{dist}(C_A) \), \( \delta_B = \text{dist}(C_B) \).

**Lemma 4.2** (Rate). The rate of the code \( C \) is bounded from below by
\[
\text{Rate}(C) \geq 2(\rho_A + \rho_B) - 3.
\]

**Proof.** For each \( e \in X^A(1) \), \( \text{codim}(C_e) = \text{codim}(C_B) = |B| \cdot (1 - \rho_B) \). Similarly for each \( e \in X^B(1) \), \( \text{codim}(C_e) = \text{codim}(C_A) = |A| \cdot (1 - \rho_A) \). The number of linearly independent constraints on \( f \in C \) is at most
\[
|X^A(1)| \cdot |B| (1 - \rho_B) + |X^B(1)| \cdot |A| (1 - \rho_A) = |G||A||B|(1 - \rho_A + \rho_B).
\]

On the other hand, the dimension of the ambient space is the number of squares \( |X(2)| = |G||A||B|/4 \), see Remark 3.8. Subtracting the number of constraints from the number of bits we get a lower bound on the dimension of the code,
\[
\text{dim}(C) \geq \frac{1}{4}|G||A||B|(1 - (4 - 2(\rho_A + \rho_B))) = \frac{1}{4}|G||A||B|(2(\rho_A + \rho_B) - 3).
\]

In fact, we can do a little better. Recall that a *vertex cover* of a graph is a set of vertices that touch all of the edges. For example, if the graph is bipartite, then it has a vertex cover whose size is half the size of the graph.
Lemma 4.3 (Rate - better bound). Suppose the underlying graph of $X$ has a vertex cover of size $\nu |G|$. Then the rate of the code is at least $4\nu \rho_A \rho_B + 1 - 4\nu$. In particular, if the graph is bipartite, then $\nu = \frac{1}{2}$ and we get that

$$\text{Rate}(C) \geq 2\rho_A \rho_B - 1.$$ 

It is interesting to mention that in the expander codes of Tanner [Tan81], (whose distance and decoding were later analyzed in [SS96]), if the local code $C_0$ has rate $\rho_0$ then the global rate is shown to be at least $2\rho_0 - 1$. In our code the rate of the local code is $\text{Rate}(C_g) = \text{Rate}(C_A \otimes C_B) = \rho_A \rho_B$, and in case the graph is bipartite, we get the same bound of $2(\rho_A \rho_B) - 1$ on the rate of the global code.

**Proof.** Let $V^* \subset G$ be a vertex cover, namely, a set of vertices that touches every edge. Then $f \in C$ if and only if for every $g \in V^*$, $f|_{X_g} \in C_g$. The reason is that every edge $e$ touches some $g \in V^*$ and the constraint $f|_{X_e} \in C_e$ is implied by $f|_{X_g} \in C_g$.

Since $C_g$ is isomorphic to $C_A \otimes C_B$, it has $|A| \cdot |B| (1 - \rho_A \rho_B)$ linearly independent constraints. The dimension of the code is at least

$$\dim(C) \geq |G||A||B| \cdot \frac{1}{4} - |V^*| \cdot |A| \cdot |B| (1 - \rho_A \rho_B)$$

$$\geq \frac{1}{4} |G||A||B| \cdot (1 - 4\nu (1 - \rho_A \rho_B)) = \frac{1}{4} |G||A||B| \cdot (4\nu \rho_A \rho_B + 1 - 4\nu). \quad (4.1)$$

**Lemma 4.4** (Distance). Suppose that both $\text{Cay}(G, A)$ and $\text{Cay}(G, B)$ are $\lambda$-expanders for $\lambda < 1$. Then the distance of the code $C$ is bounded from below by

$$\text{dist}(C) \geq \delta_A \delta_B \cdot (\max(\delta_A, \delta_B) - \lambda).$$

**Proof.** Let $0 \neq f \in C$. Let $g_0 \in X(0)$ be some vertex such that $w_{g_0} = f|_{X_{g_0}} \neq 0$ (if they are all zero then $f = 0$). Observe that since $0 \neq w_{g_0} \in C_A \otimes C_B$ then $w_{g_0}$ has at least $\delta_A |A|$ non-zero columns and at least $\delta_B |B|$ non-zero rows. Let $A_1 \subset A$ be the labels of these columns, and fix $a_1 \in A_1$. We first show that

$$\mathbb{P}_{g,b} [f([a_1, g, b]) \neq 0] \geq \delta_B (\delta_B - \lambda). \quad (4.2)$$

To prove (4.2) consider the graph $\text{Cay}(G, B)$ whose vertices are $X(0)$ and the edges are $X^B(1)$, and define a function $f_{a_1} : X^B(1) \rightarrow \mathbb{F}_2$ by $f_{a_1}(\{g,b\}) = f([a_1, g, b])$. Observe that $f_{a_1}$ is well defined because for $g' = gb$, $f_{a_1}(\{g,gb\}) = f([a_1, g, b]) = f([a_1, g', b^{-1}]) = f_{a_1}(\{g', gb^{-1}\}) = f_{a_1}(\{g', g\})$.

Since $f_{a_1} \neq 0$, it must have large weight because it belongs to the expander code defined on $\text{Cay}(G, B)$ with local code $C_B$. More elaborately, for every vertex $g$ that touches an edge where $f_{a_1} \neq 0$, there must be at least $\delta_B |B|$ non-zero edges touching $g$. By Lemma 2.1 we get at least $\delta_B (\delta_B - \lambda) |X^B(1)|$ edges on which $f_{a_1} \neq 0$, which proves (4.2).

For every $a \in A_1$, the weight of $f_{a_1}$ is at least $\delta_B (\delta_B - \lambda)$, so if we choose a random $a \in A$ and then a random edge in $X^B(1)$, the probability that $a \in A_1$ is at least $\delta_A$, and conditioned on this, the probability that $f_a(e) \neq 0$ is at least $\delta_B (\delta_B - \lambda)$, so altogether

$$\mathbb{P}_{a,g,b} [f([a, g, b]) \neq 0] \geq \mathbb{P}_a [a \in A_1] \cdot \mathbb{P}_{g,b} [f_{a_1}(\{g,gb\}) \neq 0 \mid a \in A_1] \geq \delta_A \delta_B (\delta_B - \lambda).$$

Symmetrically, the weight of $f$ is also at least $\delta_B \delta_A (\delta_A - \lambda)$, and the lemma follows. □
\textbf{Theorem 4.5} (Local Testability). Suppose $X = \text{Cay}^2(A,G,B)$ is a left-right Cayley complex such that both \text{Cay}(G,A)$ and \text{Cay}(G,B)$ are $\lambda$-expanders, and such that (TNC) holds. Assume $C_A \subset \mathbb{F}_2^A$ and $C_B \subset \mathbb{F}_2^B$ are error correcting codes with relative distances $\delta_A, \delta_B > 0$ respectively and such that $C_A \otimes C_B$ is $\kappa_0$-agreement testable. If
text{c = \frac{\kappa_0}{8 + \kappa_0} \cdot \min(\delta_A, \delta_B) > \lambda} \hfill \text{(4.3)}
then $C = C(G,A,B,C_A,C_B)$ is $\kappa$-locally testable with $|A| \cdot |B|$ queries, where
$$\kappa = \min \left( \frac{1}{4(1 + |A| + |B|)} \cdot \frac{c - \lambda}{2(|A| + |B|)} \right).$$

Namely, for every $f : X(2) \rightarrow \mathbb{F}_2$,
$$\mathbb{P}_{g \in X(0)} [f|_{X_g} \notin C_g] \geq \kappa \cdot \text{dist}(f,C).$$

In words, given some potential codeword $f$, each vertex $g$ is associated with a local test that reads $f$ at all of the $|A| \cdot |B|$ squares touching $g$ and checks that these values form a codeword in the base code $C_g \cong C_A \otimes C_B$. The theorem says that the distance of $f$ to the code is upper bounded by a constant multiple of the fraction of violated local tests.

Given a base code with distance $\delta_0$ and agreement testability $\kappa_0$, the testability of the entire code is about $\Omega(\delta_0^2 \kappa_0/(|A| + |B|))$.

We prove the theorem in the next section, by describing an iterative correction algorithm that finds a codeword close to $f$ if the probability that the test rejects is not too large.

\section{Local Self-Correction Algorithm}

In this section we describe a local self-correction algorithm, see Algorithm 1, that starts with a given string $f : X(2) \rightarrow \mathbb{F}_2$ and either finds a codeword $f_0 \in C$ or gives up. We denote
$$\zeta(f) = \mathbb{P}_{g \in X} [f|_{X_g} \notin C_g],$$
the fraction of rejecting local tests. We will show that if $\zeta(f) \leq \zeta_0$ for some constant $\zeta_0 > 0$, then the algorithm finds $f_0 \in C$ such that $\text{dist}(f_0,f) \leq O(\zeta(f))$.

For each vertex $g$, let $w_g \in C_g$ be a closest codeword to $f|_{X_g}$ (breaking ties arbitrarily). We focus on the collection of local views $W = \{w_g\}$ and whether the local views of neighboring vertices agree on the common squares.

\textbf{Definition 4.6.} Given a collection of local views $W = \{w_g \in C_g \mid g \in G\}$, we define the disagreement of the collection to be
$$\Delta(W) = \mathbb{P}_{e \in X(1)} [w_g|_{X_e} \neq w_g'|_{X_e}] \hfill \text{(4.4)}$$
where $e$ is a uniformly random edge in $X(1)$.

Observe that $\Delta(W)|X(1)|$ is a non-negative integer, and this value decreases by at least 1 every step of the algorithm, so the algorithm must halt.

\textbf{Proposition 4.7.} If the algorithm outputs $f_0$ then $f_0 \in C$ and
$$\text{dist}(f,C) \leq \text{dist}(f,f_0) \leq 4(1 + |A| + |B|) \cdot \zeta(f).$$
Proposition 4.8. If the algorithm outputs “far” then the algorithm will then find the nearest codeword.

Proof of Theorem 4.5 assuming Propositions 4.7 and 4.8. Given \( f : X(2) \to \mathbb{F}_2 \), run the algorithm above. The output is either a function \( f_0 : X(2) \to \mathbb{F}_2 \) by choosing for each square \( s \in X(2) \) an arbitrary vertex \( g \in s \) and setting \( f_0(s) = w_g(s) \). Output \( f_0 \).

Algorithm 1: Iterative decoding algorithm. (input: \( f : X(2) \to \mathbb{F}_2 \))

1. (Initialization:) For each vertex \( g \), let \( w_0^g \in C_g \) be a closest codeword to \( f|_{X_g} \) (breaking ties arbitrarily).

\[
  w_0^g = \arg \min_{w \in C_g} \text{dist}(w, f|_{X_g}).
\]

Let \( w_g \leftarrow w_0^g \) for all \( g \in G \), and let \( W = \{w_g\} \).

2. (Main loop:) If there is a vertex \( g \) and a choice \( w \in C_g \) that reduces \( \Delta(W) \) then replace \( w_g \) by \( w \) and repeat.

3. (End:) If \( \Delta(W) > 0 \) output “far”. Otherwise, \( \Delta(W) = 0 \), define \( f_0 : X(2) \to \mathbb{F}_2 \) by setting \( f_0(s) = w_g(s) \). Output \( f_0 \).

Let \( W^0 = \{w_0^g\} \) be the collection of local views defined in the initialization step of the algorithm, and let \( W = \{w_g\} \) be the final collection, at the end of the algorithm.

Proposition 4.8. If the algorithm outputs “far” then \( \Delta(W) \geq \varepsilon_0 = \frac{c - \lambda}{|A| + |B|} \), where \( c = \frac{c_0}{\pi + \kappa_0} \cdot \min(\delta_A, \delta_B) \) is defined in (4.3).

Proof of Theorem 4.5 assuming Propositions 4.7 and 4.8. Given \( f : X(2) \to \mathbb{F}_2 \), run the algorithm above. The output is either a function \( f_0 \), which by Proposition 4.7, satisfies \( \text{dist}(f, C) \leq \text{dist}(f, f_0) \leq 4(1 + |A| + |B|) \cdot \zeta(f) \); or the output is “far”, in which case \( \Delta(W) \geq \varepsilon_0 \) by Proposition 4.8. We observe that

\[
  \Delta(W^0) \leq 2\zeta(f) \tag{4.5}
\]

for the following reason. For each edge \( \{g, g'\} \) that contributes to \( \Delta(W^0) \) either \( f|_{X_g} \neq w_0^g \) or \( f|_{X_{g'}} \neq w_0^{g'} \), otherwise

\[
  w_0^g|_{X_{gg'}} = (f|_{X_g})|_{X_{gg'}} = f|_{X_{gg'}} = (f|_{X_{g'}})|_{X_{gg'}} = w_0^{g'}|_{X_{gg'}}.
\]

Therefore, the process of selecting an edge uniformly and then a random endpoint of it will lead to a rejecting vertex with probability at least \( \Delta(W^0)/2 \), proving (4.5).

Now \( \zeta(f) \geq \Delta(W^0)/2 \geq \Delta(W)/2 \geq \varepsilon_0/2 = \frac{c - \lambda}{2(|A| + |B|)} \), so we can write

\[
  \text{dist}(f, C) \leq 1 \leq \frac{2(|A| + |B|)}{c - \lambda} \cdot \zeta(f).
\]

All in all we get,

\[
  \text{dist}(f, C) \leq \max(4(1 + |A| + |B|), \frac{2(|A| + |B|)}{c - \lambda}) \cdot \zeta(f) = \kappa \cdot \mathbb{P}(f|_{X_g} \notin C_g)
\]

as needed. \( \square \)

Remark 4.9. Algorithm 1 is clearly also a decoding algorithm in the standard sense: if we know that the given word \( f \) is close enough to the code, then the regular structure of the tester (each square affects exactly four vertices) implies that it will be rejected with probability proportional to \( \text{dist}(f, C) \). The analysis herein shows that for small enough (constant) distance, the algorithm will then find the nearest codeword.
We now turn to prove the two propositions.

**Proof of Proposition 4.7.** By assumption, $\Delta(W) = 0$. We first observe that the value of $f_0(s)$ does not depend on the choice of $g \in s$ because $\Delta(W) = 0$ implies that $w_g(s) = w_{g'}(s)$ for any $g, g' \in s$. (Suppose $g_1, g_2 \in s$ disagree. If they are adjacent this means that $w_{g_1}$ disagrees with $w_{g_2}$ contradicting $\Delta(W) = 0$. If they are non-adjacent, they have a common neighbor which cannot agree with both of them). It follows that $f_0 \in C$, because for each $g$, $f_0|_{V_g} = w_g \in C_g$. To bound $\text{dist}(f, f_0)$, let

$$
V_0 = \{g \in X(0) \mid f|_{X_g} \neq w^0_g\}, \quad V_1 = \{g \in X(0) \mid w^0_g \neq w_g\}.
$$

So $V_0$ is the set of vertices whose local view does not perfectly satisfy the constraints of the code, and $V_1$ is the set of vertices $g$ for which $w_g$ at the end of the algorithm differs from its initial value.

Observe that $g \in V_0$ iff $f|_{X_g} \notin C_g$, so by definition,

$$
|V_0| = \zeta(f) \cdot |X(0)|. 
$$

(4.6)

Any square $s$ that does not touch $V_0 \cup V_1$ must have for every $g \in s$

$$
f_0(s) = w_g(s) = w^0_g(s) = f(s),
$$

where the second equality is because $g \notin V_1$ and the third is because $g \notin V_0$. We bound $|V_1|$ by the number of iterations of the algorithm, which is at most $|V_1| \leq \Delta(W^0) \cdot |X(1)|$. We recall from (4.5) that $\Delta(W^0) \leq 2\zeta(f)$. Thus, we have,

$$
|V_1| \leq \Delta(W^0) \cdot |X(1)| \leq 2\zeta(f) \cdot \frac{|A| + |B|}{2} |X(0)|.
$$

(4.7)

Altogether, since every vertex touches $|A||B|$ squares, and since $|X(2)| = |A||B||X(0)|/4$, and using (4.6) and (4.7), we get

$$
\text{dist}(f, f_0) \leq \frac{|A||B| \cdot |V_0 \cup V_1|}{|X(2)|} = \frac{4 \cdot |V_0 \cup V_1|}{|X(0)|} \leq 4(1 + |A| + |B|)\zeta(f).
$$

\square

The interesting part of the proof is to show that if $\Delta(W) > 0$ after the algorithm ends, then $\Delta(W) > \varepsilon_0 = \frac{c-\lambda}{|A| + |B|}$. 

**Proof of Proposition 4.8.** Let

$$
R = \{e = \{g, g'\} \in X(1) \mid w_g|_{X_g} \neq w_{g'}|_{X_x}\}
$$

be the set of “dispute” edges. The rest of the proof is aimed towards showing $\Delta(W) \geq \varepsilon_0$ or equivalently, since $\Delta(W) = |R|/|X(1)|$, that

$$
|R| \geq \frac{c-\lambda}{|A| + |B|} \cdot |X(1)| = \frac{c-\lambda}{2} \cdot |G|.
$$

(4.8)

First, some more notations. For an edge $\{g, ag\} \in X^A(1)$ let

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

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

$$
E^\parallel(\{g, gb\}) = \{\{ag, agb\} \in X^B(1) \mid a \in A\}.
$$
For a vertex \( g \), let
\[
E^A(g) = \{ \{ a, g \} \mid a \in A \} , \quad E^B(g) = \{ \{ g, b \} \mid b \in B \}.
\]

We now make two claims on the local structure of \( R \). The first is due to the local
distance, and the second is due to the local testability of our tensor code.

**Claim 4.10.** Suppose \( \{ a, g \} \in R \), then
\[
|R \cap E^B(g)| + |R \cap E^B(\{ g, b \})| + |R \cap \{ \{ a, g \} \}| \geq \delta_B |B|.
\]

Similarly, suppose \( \{ g, b \} \in R \), then
\[
|R \cap E^A(g)| + |R \cap E^A(\{ g, a \})| + |R \cap \{ \{ g, b \} \}| \geq \delta_A |A|.
\]

**Proof.** Let \( e = \{ a, g \} \in R \), so \( w_{g|X_e} \neq w_{ag|X_e} \). Since \( w_{g|X_e}, w_{ag|X_e} \in C_e \), these are two
distinct codewords of \( C_e \), and must disagree on at least \( \delta_B |B| \) squares. Let \([ a, g, b \]
be such a square, and look at the three edges of the square that are not
\([ a, gb, agb \]. Observe that \( a, gb, agb \)
completes the “opinion” of \( a \). At least one of the three edges must be in \( R \), because \( w_g, w_{gb}, w_{ag} \)
cannot all agree on the value of \([ a, g, b \]) without contradicting \( w_g(\{ a, g, b \}) \neq w_{ag}(\{ a, g, b \}) \).
This implies the first part of the claim, and the second part is proven similarly. \( \square \)

Recall that we assume \( C_A \otimes C_B \) is agreement testable, as per Definition 2.8.

**Claim 4.11.** Assume \( C_A \otimes C_B \) is \( \kappa_0 \)-agreement testable. For every \( g \in G \),
\[
\mathbf{P}_a[\{ g, a \} \in R] + \mathbf{P}_b[\{ g, b \} \in R] \leq \kappa_0^{-1} \cdot \mathbf{P}_a \{ \{ g, ag \} \in R \text{ or } \{ gb, agb \} \in R \} , \quad (4.9)
\]

**Proof.** Define \( w_0, w_1, w_2 : A \times B \rightarrow \mathbb{F}_2 \) as follows. First, let \( w_0(a, b) = w_g(\{ a, g, b \}) \). Next,
let \( w_1(a, b) = w_{ag}(\{ a^{-1}, ag, b \}) \). Similarly let \( w_2(a, b) = w_{gb}(\{ a, gb, b^{-1} \}) \). In words,
the \( a \)th row of \( w_1 \) comes from the “opinion” of \( w_{ag} \), and the \( b \)th column of \( w_2 \) comes from
the “opinion” of \( w_{gb} \). Observe that \( w_0 \in C_A \otimes C_B, w_1 \in \mathbf{F}_2^A \otimes C_B \), and \( w_2 \in C_A \otimes \mathbf{F}_2^B \).
Now observe that \( w_1(\cdot, \cdot) \neq w_0(\cdot, \cdot) \) iff \( \{ g, ag \} \in R \), and \( w_2(\cdot, \cdot) \neq w_0(\cdot, \cdot) \) iff \( \{ gb, agb \} \in R \).
Finally, \( w_1(a, b) \neq w_2(a, b) \) implies that the event on the RHS of (4.9) holds, namely,
\( \{ ag, agb \} \in R \) or \( \{ gb, agb \} \in R \).

By the \( \kappa_0 \)-agreement testability of \( C_A \otimes C_B \), there is a word \( w \in C_A \otimes C_B \) such that
\[
\mathbf{P}_a[\{ w(a, \cdot) \neq w_1(a, \cdot) \}] + \mathbf{P}_b[\{ w(\cdot, b) \neq w_2(\cdot, b) \}] \leq \kappa_0^{-1} \cdot \mathbf{P}_a \{ w_1(a, b) \neq w_2(a, b) \} .
\]

Since the iterative algorithm has terminated, we know that
\[
\mathbf{P}_a[\{ w_0(a, \cdot) \neq w_1(a, \cdot) \}] + \mathbf{P}_b[\{ w_0(\cdot, b) \neq w_2(\cdot, b) \}] \leq \mathbf{P}_a[\{ w(a, \cdot) \neq w_1(a, \cdot) \}] + \mathbf{P}_b[\{ w(\cdot, b) \neq w_2(\cdot, b) \}]
\]
otherwise the algorithm would have flipped from \( w_g = w_0 \) to \( w_g = w \). Combining the
inequalities the claim follows,
\[
\mathbf{P}_a[\{ g, ag \} \in R] + \mathbf{P}_b[\{ g, gb \} \in R] = \mathbf{P}_a[\{ w_0(a, \cdot) \neq w_1(a, \cdot) \}] + \mathbf{P}_b[\{ w_0(\cdot, b) \neq w_2(\cdot, b) \}]
\leq \kappa_0^{-1} \cdot \mathbf{P}_a \{ w_1(a, b) \neq w_2(a, b) \}
\leq \kappa_0^{-1} \cdot \mathbf{P}_a \{ \{ ag, agb \} \in R \text{ or } \{ gb, agb \} \in R \} .
\]

\( \square \)
Lemma 4.13. Fix \( \text{Lemma 4.13.} \) We give a combinatorial interpretation to 
\[ \langle f, \gamma M \rangle \leq \gamma \cdot \min(\delta_A, \delta_B) \cdot \langle f, f \rangle_{\mathcal{D}_1}. \]

\text{Proof.} We rely on the fact that \( \mathcal{D}_1 \) can be described by first choosing a uniform vertex \( g \) and then a random edge containing \( g \) such that with probability half we choose an \( A \) edge and with probability half a \( B \) edge. For any \( f_1 : X(1) \rightarrow \mathbb{R} \) and \( f_0 : X(0) \rightarrow \mathbb{R} \) we have
\[ \langle f_1, f_0 \rangle = \mathbb{E} \left[ \mathbb{E}_{e \sim \mathcal{D}_1} \langle f_1(e), f_0(g) \rangle \right] = \mathbb{E} \left[ \mathbb{E}_{e \sim \mathcal{D}_1} \mathbb{E}_{g \sim e} \langle f_1(e), f_0(g) \rangle \right] = \langle f_1, U f_0 \rangle_{\mathcal{D}_1}. \]

Now, if \( \langle f_1, 1 \rangle = 0 \) then \( \langle D f_1, 1 \rangle = 0 \), so
\[ \langle f_1, M f_1 \rangle = \langle f_1, U M_0 D f_1 \rangle = \langle D f_1, M_0 D f_1 \rangle \leq \lambda \langle D f_1, D f_1 \rangle \leq \lambda \langle f_1, f_1 \rangle. \]

The following lemma is based on Claims 4.10 and 4.11.

Lemma 4.13. Fix \( \gamma = \frac{\delta_a}{8 + \delta_B} \). Let \( M = U M_0 D \) and let \( f = 1_R : X(1) \rightarrow \mathbb{R} \) be the indicator function of the edge set \( R \). Then
\[ \langle f, (\gamma M) f \rangle_{\mathcal{D}_1} \geq \gamma \cdot \min(\delta_A, \delta_B) \cdot \langle f, f \rangle_{\mathcal{D}_1}. \]

\text{Proof.} We give a combinatorial interpretation to \( \gamma M f \) by observing that for a fixed \( e \in X(1) \), \( (\gamma M)f(e) \) is the probability that \( e' \in R \) in the following random process.

1. Start from an edge \( e \in X(1) \).
2. With probability \( \gamma \), output a uniformly random edge \( e' \in E(||e||) \) and halt. With probability \( 1 - \gamma \), continue.
3. Choose at random one of the endpoints of the edge, \( g_1 \in e \).
4. 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 \( b_1 \in B \).
5. 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 \( g_2 = g_2 b_2 \) for a random \( b_2 \in B \). Output \( e' \).
We will prove the lemma by showing that for every \( e \in R \),
\[
(\gamma M^\| + (1 - \gamma) M) f(e) \geq \gamma \cdot \min(\delta_A, \delta_B) .
\]
(4.10)
So fix some \( e \in R \), and for convenience assume \( e = \{g, ag\} \) for some \( g \in G, a \in A \) (if \( e = \{g, gb\} \) the argument is symmetric). Let
\[
r_0 = |R \cap E^\| (e)|, \quad r_1 = |R \cap E^B (g)|, \quad r_2 = |R \cap E^B (ag)| .
\]
By Claim 4.10, \( r_0 + r_1 + r_2 \geq \delta_B |B| \). With probability \( \gamma \) step 2 outputs a random \( e' \in E^\| (e) \), and the probability it is in \( R \) is \( r_0/|B| \).
\[
P[e' \in R] = \gamma \cdot r_0/|B| + (1 - \gamma) \cdot P[e' \in R \mid \text{the process entered step 3}]
\]
(4.11)
Assume we entered step 3. Due to Claim 4.11,
\[
P[e_{a,b} = \{ag_1, ag_1 b\} \in R \mid \{g_1 b, ag_1 b\} \in R] \geq \kappa_0 \cdot r_i/|B| \]
(4.12)
where \( i \in \{1, 2\} \) depending on whether \( g_1 = g \) or \( g_1 = ag \) as chosen in step 3. What is the probability that \( e' \) is one of the edges \( \{ag_1, ag_1 b\} \) and \( \{g_1 b, ag_1 b\} \) considered in the LHS of (4.12)? This happens exactly if in steps 4 and 5 we will walk in alternating colors \((A, B) \text{ or } (B, A)\). Let \( E_{AB} \) be the event that in step 4 we choose an \( A \)-edge, i.e. \( g_2 = a_1 g_1 \) for some \( a_1 \in A \) and then in step 5 we set \( e' \) to be a \( B \)-edge, i.e. \( e' = \{a_1 g_1, a_1 g_1 b_2\} \) for some \( b_2 \in B \). Similarly let \( E_{BA} \) be the event that \( g_2 = g_1 b_1 \) and \( e' = \{g_1 b_1, a_2 g_1 b_1\} \). Clearly
\[
P[E_{AB}] = P[E_{BA}] = \frac{1}{4} .
\]
Now,
\[
P[E_{AB} \text{ and } e' \in R] = \frac{1}{4} \cdot P[a_{1, b_2} \{a_1 g_1, a_1 g_1 b_2\} \in R], \quad (4.13)
\]
and
\[
P[E_{BA} \text{ and } e' \in R] = \frac{1}{4} \cdot P[b_{2, b_1} \{g_1 b_1, a_2 g_1 b_1\} \in R]. \quad (4.14)
\]
where the probability is taken over the randomness of the random process above conditioned on having entered step 3. Since \( E_{AB} \) and \( E_{BA} \) are disjoint events,
\[
P[e' \in R] \geq P[E_{AB} \text{ and } e' \in R] + P[E_{BA} \text{ and } e' \in R]
\[
\geq \frac{1}{4} \cdot P[a_{1, b_2} \{ag_1, ag_1 b\} \in R] + P[a_{1, b_2} \{g_1 b, ag_1 b\} \in R]
\[
\geq \frac{1}{4} \cdot \frac{r_i \kappa_0}{4|B|}\]
where in the last inequality we have used (4.12). We conclude that if in step 3 we choose \( g_1 = g \), then \( P[e' \in R] \geq \frac{r_i \kappa_0}{4|B|} \), whereas if in step 3 we choose \( g_1 = ag \), then \( P[e' \in R] \geq \frac{r_i \kappa_0}{4|B|} \).
Altogether, recalling (4.11),
\[
P[e' \in R] \geq \gamma \cdot \frac{r_0}{|B|} + (1 - \gamma) \cdot \frac{\kappa_0}{4|B|}(r_1 + r_2)/2.
\]
Plugging in $\gamma = \frac{\kappa_0}{\delta + \kappa_0}$, we get $1 - \gamma = 8\gamma/\kappa_0$, and recalling that $r_0 + r_1 + r_2 \geq \delta |B|$, we have
\[
P[e' \in R] \geq \gamma (r_0 + r_1 + r_2)/|B| \geq \gamma \delta_B.
\]
We have seen that if $e = \{a,ag\}$ for some $a,g$ is in $R$, then $e' \in R$ with probability at least $\gamma \delta_B$. Symmetrically, if $e = \{g,gb\}$ for some $g,b$ is in $R$ then we would get that $e' \in R$ with probability at least $\gamma \delta_A$. Together this proves (4.10) and completes the proof of Lemma 4.13.

Recall from (4.3) that $c = \frac{\kappa_0}{\delta + \kappa_0} \cdot \min(\delta_A, \delta_B)$. By Lemma 4.13, $(f, (\gamma M|| + (1 - \gamma)M)f) \geq c \cdot (f, f)$ so either
\[
(f, Mf) \geq c(f, f) \tag{4.15}
\]
or
\[
(f, M||f) \geq c(f, f). \tag{4.16}
\]

If (4.15) holds, then by Lemma 2.2, applied with the operator $M$, whose vertex set $X(1)$ is endowed with the distribution $D_1$, we get $P_{D_1}[R] \geq c - \lambda$ which means that $|R| \geq \frac{\kappa_0}{\delta + \kappa_0} \cdot \min(|A|, |B|)(c - \lambda)$.

Otherwise, assume that (4.16) holds. By Lemma 3.13 there exists some $\sigma \in \hat{A} \cup \hat{B}$ such that $|R \cap X^\sigma(1)| \geq |G|(c - \lambda)/2$.

This completes the proof of Proposition 4.8 showing that if $\Delta(W) > 0$ then $\Delta(W) > \frac{2(c - \lambda)}{|\hat{A} \cup \hat{B}|}$. \hfill \square

## 5 A Concrete Construction

In the previous section we have described a code scheme: Given a left-right Cayley complex $\text{Cay}^2(A, G, B)$ together with two base codes $C_A \subseteq F_2^D$ and $C_B \subseteq F_2^D$, we get an error correcting code $C[G, A, B, C_A, C_B]$.

In this section we prove our main theorem by showing how to find an infinite family of left-right Cayley complexes and base codes that yield locally testable codes.

**Theorem** (Restatement of Theorem 1.1). For every $0 < r < 1$, there exist $\delta, \kappa > 0$, $q \in \mathbb{N}$ and an explicit construction of an infinite family of error correcting codes $\{C_n\}_n$, such that for each $n$, $\text{Rate}(C_n) \geq r$, $\text{dist}(C_n) \geq \delta$ and $C_n$ is $\kappa$-locally testable with $q$ queries.

The proof of the theorem relies on the following two lemmas.

**Lemma 5.1** (Good base code). For every $0 < r_0 < 1$, there exist $\delta_0, \kappa_0 > 0$ and $d_0, D_0 \in \mathbb{N}$, such that for every integer $D > D_0$ that is divisible by $d_0$, there exists a linear error correcting code $C_0 \subseteq \mathbb{F}_2^D$ with rate at least $r_0$, distance at least $\delta_0$, and such that the tensor code $C_0 \otimes C_0$ is $\kappa_0$-agreement testable.

**Lemma 5.2** (Good left-right Cayley complexes). Let $d_0, D_0 \in \mathbb{N}$. Let $q$ be any odd prime power such that $q \geq \max\{2d_0^2, D_0, 17\}$ and define $D = d_0 \cdot \lceil \frac{q+1}{4d_0} \rceil$. Then there exist an explicit construction of an infinite family of finite groups $G_i = \text{PSL}_2(q^i)$, with two symmetric generating subsets $A_i, B_i \subseteq G_i$, such that for every $i$, both $A_i$ and $B_i$ are of size $D$ hence divisible by $d_0$, $A_i$ and $B_i$ satisfy condition (TNC), and the Cayley graphs $\text{Cay}(G_i, A_i)$ and $\text{Cay}(G_i, B_i)$ are $\lambda$-expanders where $\lambda \leq 8D^{-1/2}$.

We prove Lemma 5.1 in Subsection 5.1 by showing that random LDPC codes are smooth. We prove Lemma 5.2 in Section 6 using the known constructions of Ramanujan graphs by Lubotzky, Samuels and Vishne [LSV05b] and Morgenstern [Mor94].

Let us now deduce Theorem 1.1 from Lemmas 5.1 and 5.2.
Proof of Theorem 1.1. Fix $0 < r < 1$ and set $r_0 = \frac{r+3}{4}$ so that $r = 4r_0 - 3$. By Lemma 5.1, given $r_0$, there exist $\delta_0, \kappa_0 > 0$ and $d_0, D_0 \in \mathbb{N}$, such that for any $D > D_0$ divisible by $d_0$, there exists a code $C_0 \subset \mathbb{F}_q^D$ with $\text{Rate}(C_0) \geq r_0$, $\text{dist}(C_0) \geq \delta_0$ and such that $C_0 \otimes C_0$ is $\kappa_0$-agreement testable.

Define $q_0 = \max\{2D_0, 2q_0^2, 27\left(\frac{\alpha_0 + \beta_0}{\kappa_0 \delta_0}\right)^2\}$. For any $q > q_0$ odd prime power denote $D = d_0 \cdot \left\lfloor \frac{q+1}{d_0} \right\rfloor$. Note that $q + 1 \geq D \geq q + 1 - d_0 > q - \sqrt{q} > \frac{1}{2}q$. In particular $D > \frac{1}{2}q_0$, hence $8D^{-1/2} < 2^{7/2}q_0^{-1/2} \leq \frac{\kappa_0 \delta_0}{8+\kappa_0}$, which also implies $8D^{-1/2} < \delta_0$.

By Lemma 5.2 there exists an explicit construction of an infinite family of groups $G_i = PSL_2(q^i)$ together with generating sets $A_i, B_i$ such that for each $i \in \mathbb{N}$, $|A_i| = |B_i| = D$, conditions (TNC) holds, and both $Cay(G_i, A_i)$ and $Cay(G_i, B_i)$ are $\lambda = 8D^{-1/2}$ expanders. In particular, from our choice of $D$, equation (4.3) holds and $\lambda < \delta_0$.

By Lemma 5.1 there exists a code $C_0$ of length $D$, with rate at least $r_0$, distance at least $\delta_0$ and such that the tensor code $C_0 \otimes C_0$ is $\kappa_0$-agreement testable. Since $D$ is a constant we can, theoretically, enumerate over all possible codes in search of a good one.

Define our family of global codes to be $C_i = C[G_i, A_i, B_i, C_0, C_0], i \in \mathbb{N}$, and by the above choices it has the following parameters:

- Block-length $\frac{1}{4}|G_i|D^2$, where $|G_i| = \frac{1}{2}(q^i - q^i)$.
- Distance at least $\delta = \delta_0^2(\delta_0 - 4D^{-1/2}) > 0$, by Lemma 4.4.
- Rate at least $r = 4r_0 - 3 > 0$, by Lemma 4.2.
- It is $\kappa$-locally testable with $D^2$ queries, by Theorem 4.5, for

$$\kappa = \min\left\{\frac{1}{4 + 8D}, \frac{1}{4D} \left(\frac{\delta_0 \kappa_0}{8 + \kappa_0} - 4D^{-1/2}\right)\right\}.$$  \hspace{1cm} (5.1)

Assuming that $\delta_0, \kappa_0$ are large with respect to $D$ we get distance $\approx \delta_0^3$ and detection probability $\kappa \approx \theta(\delta_0 \kappa_0 / D)$. \hfill \Box

5.1 Good Base Codes

In this section we prove Lemma 5.1 by relying on the notion of smooth codes from [DSW06], which was consequently broadened to weakly-smooth codes in [BV09]. These works showed that the tensor product of a smooth code and any other code is robustly testable and therefore, by Lemma 2.9, also agreement testable.

Definition 5.3 (Smooth Code). Let $c, d, n \in \mathbb{N}, \alpha, \beta, \delta > 0$. A $(c, d, n)$-LDPC code $C \subset \mathbb{F}_q^n$ is $(\alpha, \beta, \delta)$-smooth if for every $Y_0 \subseteq Y$ with $|Y_0| \leq \alpha |Y|$ there is some $X_0 \subseteq X$ with $|X_0| \leq \beta |X|$ such that the code $C(Y_0)|_{X_0}$ has distance at least $\delta$. Here the code $C(Y_0)|_{X_0}$ is the code obtained by removing the constraints in $Y_0$ and then removing the coordinates in $X_0$.

5.1.1 Random LDPC Codes

We will next show that random LDPC codes satisfies w.h.p. the requirements of Lemma 5.1.

Random LDPC codes, see Definition 2.6, were famously introduced by Gallager in his PhD thesis [Gal63]. A random $(c, d, n)$-code is given by selecting a random $(c, d)$-regular bipartite graph (called the factor graph of the code), which in turn is done by taking a
random matching between the $nc$ “half-edges” on the left and the $md$ “half-edges” on the right, where we assume that $nc/d$ is an integer.

Spielman described in his thesis [Spi96] the following expansion property,

**Definition 5.4.** A $(c,d)$-regular bipartite graph $([n],[m],E)$ is a $(\delta,\gamma)$-expander if every set of left vertices $A \subset [n]$ whose size is at most $\delta n$, has at least $c|A|(1-\gamma)$ neighbors.

**Claim 5.5** (Claim 6.4 in [BHR05]). For any $d > c > 2$, any $\gamma > \frac{1}{2}$ and any $n$ such that $nc/d \in \mathbb{N}$, then w.h.p. a random $(c,d)$-biregular graph $([n],[m],E)$ is a $(\delta,\gamma)$-expander for any $\delta$ satisfying

$$\delta \leq \left(2 \cdot \epsilon^{c+1-c\gamma} \cdot d^{c\gamma} \cdot (1-\gamma)^{c\gamma}\right)^{-1/(c\gamma-1)}. \tag{5.2}$$

Moreover, there exists at least one such graph for any $n \geq n_0$ divisible by $d$, where $n_0$ is the minimal integer satisfying

$$c^{c+1-c\gamma} \cdot d^{c\gamma} \cdot (1-\gamma)^{c\gamma} \cdot n_0^{-1/9} + n_0 \cdot 2^{-n_0 \min\left(\frac{1}{c\gamma-1}\right)} < 1. \tag{5.3}$$

The proof is similar to Gallager’s proof [Gal63] that a random LDPC code has constant distance with high probability. Note that the upper bound on $n_0$ follows from showing that the last equation in the proof in [BHR05], which upper bound the probability that a random graph is not a $(\delta,\gamma)$-expander, is smaller then 1.

**Claim 5.6.** A $(c,d,n)$-LDPC code whose factor graph is $(\delta,\gamma)$-expander graph, with $\gamma < \frac{1}{2}$, has rate at least $1 - \frac{\delta}{c}$ and distance at least $\delta$.

**Proof.** The number of constraints of the code is $m = nc/d$, hence the dimension of the code is at least $n - m = n(1 - \frac{\delta}{c})$, so the rate is at least $1 - \frac{\delta}{c}$. An LDPC code whose factor graph is a $(\delta,\gamma)$-expander with $\gamma < \frac{1}{2}$, has the unique neighbor expansion property [Spi96], i.e. that for each subset $A \subset [n]$, $|A| \leq \delta n$, there exists $u \in [m]$ with a unique neighbor in $A$, which implies that the distance of the code is at least $\delta$. \qed

In [DSW06] and [BV09] it is shown that tensors of random LDPC codes are robustly testable,

**Theorem 5.7** (Robust testability of expander codes). Let $C$ be a $(c,d,n)$-code whose factor graph is a $(c,d)$-regular $(\delta,\gamma)$-expander. Let $C'$ be any linear code with distance $\delta'$. Then $C \otimes C'$ is $\tau$-robustly testable for

$$-\tau \geq \frac{8\delta' \cdot \frac{1}{2} - \gamma}{2d} \text{ when } \gamma < 1/6 \text{ [DSW06], and}$$

$$-\tau \geq \frac{8\delta'}{d \min\left(\frac{1}{c\gamma-1}, \frac{1}{3c\gamma-1}\right)} \text{ for all } \gamma < 1/2 \text{ [BV09].}$$

Finally, we can prove Lemma 5.1, which we restate for convenience,

**Lemma** (Restatement of Lemma 5.1). For all $0 < r_0 < 1$, there exist $d_0$, $D_0 \in \mathbb{N}$, such that for every integer $D > D_0$ that is divisible by $d_0$, there exists a linear error correcting code $C_0 \subseteq \mathbb{F}_2^D$ with rate at least $r_0$, distance at least $\delta_0$, and such that the tensor code $C_0 \otimes C_0$ is $\kappa_0$-agreement testable.

---

1 A $(\delta,\gamma)$-expander here is called a $(c(1-\gamma),\delta)$-left-expander in [BHR05].
Proof. We fix $\gamma_0 = 0.15 < 1/6$ and set $c_0 = 7$ so that $\gamma_0 > 1/c_0$. We choose $d_0 = \lceil \frac{7}{1 - r_0} \rceil$ such that $\frac{d_0}{4} \leq 1 - r_0$. Claim 5.5 guarantees existence of $\delta_0 > 0$ and $D_0 = n_0$ such that for all $D > D_0$ divisible by $d_0$, a random $(c_0, d_0)$-regular bipartite graph with $D$ left vertices is a $(\delta_0, \gamma_0)$-expander with high probability. Moreover, we can take $D_0$ by Claim A.1 these codes are $\kappa_0$-agreement testable for $\kappa_0 = \frac{\delta_0^3 (\frac{1}{2} - \gamma_0)}{4d_0}$. □

Remark 5.8. In Lemma 5.1, the parameters $\delta_0$, $\kappa_0$ and $D_0$ depend on the parameter $\varepsilon = 1 - r_0$, as follows (ignoring absolute constants)

\[
\delta_0 = (2 \cdot e^{6.95 \cdot ([7\varepsilon^{-1}])^{1.05} \cdot 0.85^{1.05}})^{-20} = \Omega(\varepsilon^{21}),
\]

which follows directly from equation (5.2),

\[
\kappa_0 = \frac{\delta_0^3 \cdot 0.01666}{4d_0} = \Omega(\varepsilon^{64}),
\]

and by bounding each of the two right terms in equation (5.3) by $\frac{1}{2}$, we get

\[
D_0 \leq \max\{(2 \cdot e^{6.95 \cdot ([7\varepsilon^{-1}])^{1.05} \cdot 0.85^{1.05}})^9, 2^{400}\} = O(\varepsilon^{-9}).
\]

This shows that the parameters $\delta_0$, $\kappa_0$ and $D_0$ are polynomials in $\varepsilon = 1 - r_0$.

6 Good Left-Right Cayley Complexes

In the previous section we showed how to construct good locally testable codes on left-right Cayley complexes provided the latter have sufficiently large spectral gap. To finish the proof of the main result of the paper, we need to show that such complexes indeed exist and to give an explicit construction. Namely, in this section we prove Lemma 5.2.

More generally, we show that for every $\lambda > 0$, there exist $k_1, k_2 \in \mathbb{N}$ and an infinite family of finite groups $G_i$, with two symmetric subsets of generators $A_i, B_i$, such that for each $i$, $|A_i| = k_1$ and $|B_i| = k_2$, the two sets $A_i$ and $B_i$ satisfies (TNC), and the second largest eigenvalues of the normalized adjacency matrices of Cay($G_i, A_i$) and Cay($G_i, B_i$), denoted $\lambda($Cay($G_i, A_i$)) $\lambda($Cay($G_i, B_i$)), are bounded from above by $\lambda$. Moreover, we can take $\lambda = \Theta(k_1^{-1/2}) = \Theta(k_2^{-1/2})$, making both Cayley graphs quasi-Ramanujan.

There are a number of ways in the literature to find Cayley graphs with small $\lambda($Cay($G, S$)). There are even various methods to give different sets of generators for the same group (see [Lub94], [LSV05b]). The difficulty is to ensure that condition (TNC) is satisfied. We will show two (actually three) ways to do so. In all of our constructions, the elements in the sets $B_i$ will be of order 2, while all the elements in $A_i$ will be of order greater than 2. This ensures that (TNC) is automatically satisfied.

6.1 The Morgenstern Generators, $q = 2^\ell$

In [Mor94], Morgenstern presented for every prime power $q$, infinitely many groups $G_i = \text{PGL}_2(q^\ell)$ or $G_i = \text{PSL}_2(q^\ell)$ each with a symmetric set $B_i$ of $q + 1$ generators such that $\lambda($Cay($G_i, B_i$)) are Ramanujan, i.e., $\lambda($Cay($G_i, B_i$)) $\leq \frac{2\sqrt{q}}{q+1}$. 

\[\text{Page 25}\]
The case of \( q \) even, i.e., \( q = 2^k \), is special in two ways. First of all, here \( PGL_2(q') = PSL_2(q') \), so this is always a simple group. But more importantly, in this case all the elements of \( B_i \) are of order 2 (see Remark 6.3 below). Assume \( q \) is even from now on.

Morgenstern constructed an explicit arithmetic lattice \( \Gamma \) in the group \( PSL_2(F_q((t)))) \) which is isomorphic to the free product \( \langle b_0 \rangle * ... * \langle b_q \rangle \), where \( B = \{ b_0, \ldots, b_q \} \) is a set of elements of order 2 (see [Mor94, Section 5]). The above mentioned Cayley graphs \( Cay(G_i, B_i) \) are identified as quotients of this \( \Gamma \) by normal congruence subgroups, where \( B_i = \phi_i(B) \) is the image of \( B \) under an epimorphism \( \phi_i : \Gamma \to G_i \). Note that by [Mor94] these Cayley graphs are all Ramanujan.

Let us now show how to get another symmetric set of generators \( A_i \) for \( G_i = PSL_2(q') \) with \( \lambda(Cay(G_i, A_i)) \) small, and such that \( A_i \) and \( B_i \) satisfy (TNC).

Let \( \Lambda \) be the index 2 subgroup of \( \Gamma \) - the kernel of the homomorphism \( \phi : \Gamma \to C_2 \) (= the cyclic group of order 2) where \( \phi \) sends each \( b_j \) to the unique non-trivial element of \( C_2 \). One can see easily that \( \Lambda \) is exactly the subgroup of all elements of \( \Gamma \) of even length w.r.t. \( B \). It is generated by the set \( A = \{ b_t b_s \mid b_t, b_s \in B, t \neq s \} \) which is of size \( k_1 = q^2 + q \). We claim

Claim 6.1. (i) For \( i > 1 \), the image \( A_i = \phi_i(A) \) of \( A \) in \( G_i \) generates \( G_i = PSL_2(q') \).

(ii) \( \lambda(Cay(G_i, A_i)) < \frac{3q-1}{q^2+q} \leq \frac{3\sqrt{q} - 1}{k_1} \).

(iii) For \( i > 4 \), the images of the elements of \( A \) in \( G_i \) are distinct from one another, and each element in \( A_i \) has order \( > 2 \).

Proof. (i) Since \( \Lambda = \langle A \rangle \) is of index two in \( \Gamma \) then \( A_i = \phi_i(A) \) is of index at most two in \( G_i \). But \( G_i = PSL_2(q') \) is simple, hence it has no index 2 subgroup (a subgroup of index 2 must be normal), which implies \( A_i = G_i \).

(ii) Let \( T_B \) and \( T_A \) be the (non-normalized) adjacency matrices of \( Cay(G_i, B_i) \) and \( Cay(G_i, A_i) \), respectively. Note that \( T_B^2 = T_A + (q + 1)I \). Hence if \( \mu \) is an eigenvalue of \( T_A \), then \( \mu = \lambda^2 - (q + 1) \) for some eigenvalue \( \lambda \) of \( T_B \). Since \( Cay(G_i, B_i) \) is Ramanujan, \( |\lambda| = q + 1 \) or \( |\lambda| < 2\sqrt{q} \). Therefore \( \mu = q^2 + q \) or \( \mu < (2\sqrt{q})^2 - (q + 1) = 3q - 1 \).

(iii) It suffices to show that each reduced word which is a product of length at most 4 in \( B \) is not in the kernel of \( \phi_i \), which is equivalent to the girth of \( Cay(G_i, B_i) \) being greater than 4. By [Mor94, Theorem 5.13 (3)] the girth of \( Cay(G_i, B_i) \) is at least \( \frac{3}{2} \log_q |G_i| \geq i \), which completes the proof. \( \Box \)

Thus, given \( \lambda > 0 \) by taking \( q \) large enough so that \( \frac{3\sqrt{q} - 1}{q^2+q} < \lambda \), we get the desired \( \lambda \)-expanding left-right Cayley complexes with \( k_1 = q^2 + q \) and \( k_2 = q + 1 \).

We can do slightly better. Note that \( \Lambda \) above, being a normal subgroup of a free product of finite groups, with trivial intersection with each factor is a free group (see Section 34 in [Kur55]). In fact, by the Reidemeister-Schreier algorithm applied to the transversal set \( \{ 1, b_0 \} \) of \( \Lambda \) in \( \Gamma \) (or by inspection) one can see that \( \Lambda \) is a free group on the \( q \) generators \( \{ b_0 b_j : j = 1, \ldots, q \} \). As \( \{ b_0 b_j \}^{-1} = b_j b_0 \) we deduce that \( \Lambda' = \{ b_0 b_j, b_j b_0 : j = 1, \ldots, q \} \) is a symmetric set of generators of \( \Lambda \).

We can now look at the image \( A_i' = \phi_i(A') \) under the epimorphism \( \phi_i : \Gamma \to G_i \).

Arguing similarly to the proof of Claim 6.1 (i), \( A_i' \) generates \( G_i \), and by the proof of (iii) above, the images are all different. Finally:

Claim 6.2. \( \lambda(Cay(G_i, A_i')) < \frac{3\sqrt{2q+1}}{2q} \).

Proof. Let \( V_i = \{ f : G_i \to \mathbb{C} \} \) and for any element \( s \in G_i \), define the \( s \)-adjacency \( T_s : V_i \to V_i, T_s f(g) = f(gs) \), and for any multiset \( S \) of \( G_i \), define the \( S \)-adjacency operator \( T_S : V_i \to V_i, T_S = \sum_{s \in S} T_s \). Note that for any two multisets \( S, S' \) of \( G_i \), \( T_{S \cup S'} = T_S + T_{S'} - T_{S \cap S'} \) and \( T_{SS'} = T_{S'} T_S \), where \( SS' = \{ ss' : s \in S, s' \in S' \} \) counted
with multiplicities. Therefore \( T_{A^j} = T_bT_{B^j} + T_{B^j}T_b - 2I \), where \( b = \phi_i(b_i) \). Let \( f \in V_i \) be such that \( f \perp 1_{G_i} \), i.e. \( \sum_{g \in G_i} f(g) = 0 \). Note that for any \( s \in G_i \), then \( T_s f \perp 1_{G_i} \) and \( \|T_s f\| = \|f\| \). By [Mor94, Theorem 5.11], we have \( \|T_{B^j} f\| \leq \sqrt{q}\|f\| \) for any \( f \perp 1_{G_i} \).

Then

\[
\|T_{A^j} f\| \leq \|T_bT_{B^j} f\| + \|T_{B^j}T_b f\| + 2\|f\| \leq \|T_{B^j} f\| + 2\|f\| = 2\sqrt{q}\|f\| \leq 3\sqrt{2q - 1}\|f\|
\]

which completes the proof. \( \square \)

So this time we have a family of \( \lambda \)-expanders left-right Cayley complexes with \( k_1 = 2q \) and \( k_2 = q + 1 \), for any \( \lambda \geq \frac{3\sqrt{2q - 1}}{2q} \).

**Remark 6.3.** Everything said above is explicit. In fact the generator set \( B_i \) of \( PSL_2(q^i) \) are given explicitly in [Mor94, equation (21)]. Assume for simplicity that \( i \) is even. Let \( i \in \mathbb{F}_{q^i} \) be such that \( i \notin \mathbb{F}_q \) and \( \varepsilon = i^2 + i \in \mathbb{F}_{q^i} \). Let \( x \in \mathbb{F}_{q^i} \) be such that \( 1, x, \ldots, x^{e_i - 1} \) form a basis for \( \mathbb{F}_{q^i} \) over \( \mathbb{F}_q \). Then the \( q + 1 \) elements of \( B_i \) are

\[
\phi_i(b_j) = \left( \begin{array}{ccc}
1 & \gamma_j + \delta_j i \\
x(\gamma_j + \delta_j i) & 1
\end{array} \right), \quad j = 0, \ldots, q,
\]

(6.1)

where \( (\gamma_j, \delta_j) \in \mathbb{F}_{q^2}^2 \) are the \( q + 1 \) solutions in \( \mathbb{F}_q \) for \( \gamma^2 + \gamma \delta + \delta^2 \varepsilon = 1 \). One indeed sees that each of the elements of \( B_i \) is of order 2.

We will pass now to a different construction, which will give us Cayley graphs of \( G_i \) w.r.t. \( A_i \) and \( B_i \) of the same size: \( |A_i| = |B_i| = q + 1 \), and both are Ramanujan.

### 6.2 The LSV Generators, \( q \) odd

In [LSV05b], Lubotzky, Samuels and Vishne constructed Ramanujan complexes, based on an arithmetic lattice \( \Gamma \), discovered by Cartwright and Steger [CS98], which acts simply transitively on the Bruhat-Tits building of \( PGL_d(\mathbb{F}_q((t))) \). The special case \( d = 2 \) gave some new Ramanujan graphs. These Ramanujan graphs were highlighted in [KL12], as edge-transitive Ramanujan graphs which have been used there to construct symmetric LDPC codes.

The arithmetic group \( \Gamma \), acting simply transitively on the Bruhat-Tits tree of \( PGL_2(\mathbb{F}_q((t))) \) \( (q \) any odd prime power) is obtained there as a the group generated by \( q + 1 \) conjugates of a specific element \( b \), conjugated by the non-split torus \( T \) of order \( q + 1 \) in \( PGL_2(\mathbb{F}_q) \). This is a symmetric set of generators \( A \) for \( \Gamma \) which generates a free group on \( \frac{q + 1}{2} \) generators. We will present below a different choice for \( b \), this time \( b' \) - an element of order 2, whose conjugation under \( T \) forms a symmetric set \( B \) of size \( q + 1 \) and generate a group \( \Gamma' \) which also acts simply transitively on the Bruhat-Tits tree. Moreover, \( \Gamma \) and \( \Gamma' \) are both finite index subgroups of an arithmetic group \( G(R) \) - to be defined below.

In [LSV05b] (see also [KL12]) it was shown that \( G(R) \) has infinitely many finite congruence quotients \( G_i \), under the maps \( \phi_i : G(R) \to G_i \), where \( G_i = PGL_2(\mathbb{F}_{q^i}) \) or \( G_i = PSL_2(\mathbb{F}_{q^i}) \), for which \( \text{Cay}(G_i, \phi_i(A)) \) are Ramanujan \( (q + 1) \)-regular graphs. We will observe below that the holds for \( \text{Cay}(G_i, \phi_i(B)) \). For \( i \) large enough (see Claim 6.5) the elements of \( \phi_i(A) \) are of order \( > 2 \) while \( \phi_i(B) \) contains only elements of order 2. Hence we will get two-sided Cayley square complexes with \( k_1 = k_2 = q + 1 \) and \( \lambda < \frac{2\sqrt{2}}{q + 1} \). By choosing \( q \) large enough, they will be \( \lambda \)-expanders for arbitrarily small \( \lambda > 0 \).
Now, in more details: Let $0 \neq \varepsilon \in \mathbb{F}_q$ be a non-square element, let $R = \mathbb{F}_q[\frac{1}{y}, \frac{1}{1+y}]$ be the subring of $\mathbb{F}_q(y)$, generated by $y$, $\frac{1}{y}$ and $\frac{1}{1+y}$, and let $A(R)$ be the quaternion $R$-algebra,

$$A(R) = R + R\alpha + R\beta + R\gamma = \alpha^2 = \varepsilon, \quad z^2 = 1 + y, \quad az = -az. \quad (6.2)$$

**Remark 6.4.** We note that our choice of basis for $A(R)$, $\{1, \alpha, z, \alpha z\}$, is based on [KL12], while [LSV05b] used a different basis for $A(R)$, $\{\xi, \xi^q, \xi^2, \xi^q\}$, where $\{\xi, \xi^q\}$ forms an $\mathbb{F}_q$-basis for $\mathbb{F}_q^2 = \mathbb{F}_q[\alpha]$. The change of bases does not affect any of the following constructions.

For any ring $D$, denote by $D^*$ its group of units. Note that an element of $r(y) \in R$ belongs to $R^*$ if and only if it is of the form $r(y) = cy^n(1+y)^m$, $c \in \mathbb{F}_q^*$, $n, m \in \mathbb{Z}$, and that an element $a = a_1 + a_2\alpha + a_3\beta + a_4\gamma \in A(R)$ belongs to $A(R)^*$ if and only if its norm $N(a) := a_1^2 - \varepsilon a_2^2 - (1+y)a_3^2 - \varepsilon(1+y)a_4^2 \in R$ belongs to $R^*$. Note also that $R$ is the center of $A(R)$ and $R^*$ is the center of $A(R)^*$. Then the principal arithmetic group $G(R)$ is defined to be

$$G(R) = A(R)^*/R^* = \{a \in A(R) : N(a) \in R^*\}/R^*.$$

The Cartwright–Steger arithmetic lattice $\Gamma$, and the second arithmetic lattice $\Gamma'$, are defined to be the subgroups of $G(R)$, generated by the symmetric sets of size $q+1$, $A$ and $B$, which are the sets of $T$ conjugates of the elements, $b$ and $b'$, respectively, where $T = \mathbb{F}_q[\alpha]^*/\mathbb{F}_q^*$ is $G(R)$ is a non-split torus of order $q+1$, $b = \left(1 - \frac{1}{1+y}z\right) R^* \in G(R)$ and $b' = ab = \left(\alpha - \frac{1}{1+y}\alpha z\right) R^* \in G(R)$, namely,

$$\Gamma = \langle A \rangle \leq G(R), \quad A = \{tbt^{-1} : t \in T\}, \quad \Gamma' = \langle B \rangle \leq G(R), \quad B = \{tb't^{-1} : t \in T\},$$

$$T = \mathbb{F}_q[\alpha]^*/\mathbb{F}_q^*, \quad b = \left(1 - \frac{1}{1+y}z\right) R^*, \quad b' = ab = \left(\alpha - \frac{1}{1+y}\alpha z\right) R^*.$$

Note that $b$ and $b'$ belongs to $G(R)$, since $N\left(1 - \frac{1}{1+y}z\right) = 1 - (1+y)\frac{1}{(1+y)^2} = \frac{y}{1+y} \in R^*$ and $N\left(\alpha - \frac{1}{1+y}\alpha z\right) = N(\alpha) \cdot N\left(1 - \frac{1}{1+y}\gamma\right) = -\varepsilon \cdot \frac{y}{1+y} \in R^*$.

**Claim 6.5.** (i) Every element of $A$ is of infinite order, while every element of $B$ is of order 2.

(ii) For $i > 2$, every element of $A_i = \phi_i(A)$ is of order $> 2$, while every element of $B_i = \phi_i(B)$ is of order 2.

**Proof.** (i) The claim about the elements of $A$ follows from [LSV05b, Corollary 5.4]. For the claim about the elements of $B$, since they are all conjugate of one another, it suffice to show $b^2 = 1$, or equivalently, $\left(\alpha - \frac{1}{1+y}\alpha z\right)^2 \in R^*$. This follows from the following computations,

$$\left(\alpha - \frac{1}{1+y}\alpha z\right)^2 = \alpha^2 - \frac{1}{1+y}\alpha az - \frac{1}{1+y}\alpha z\alpha + \frac{1}{(1+y)^2}\alpha^2z\alpha z = *,$$

and by Equation 6.2, as $z\alpha = -\alpha z$, $\alpha^2 = \varepsilon$ and $z^2 = 1 + y$, we get

$$* = \alpha^2 - \frac{1}{(1+y)^2}\alpha^2z^2 = \varepsilon - \frac{\varepsilon}{1+y} = \frac{y}{1+y} \in R^*.$$

(ii) This follows from an injectivity radius argument for congruence subgroups, see for instance [LM07].
Let $\mathcal{B}$ be the Bruhat-Tits tree of $PGL_2(\mathbb{F}_q((t)))$, which is a $(q + 1)$-regular infinite tree. By [LSV05b, Section 3], $\Gamma$, $\Gamma'$ and $G(R)$ are subgroups of $PGL_2(\mathbb{F}_q((t)))$, hence acts on $\mathcal{B}$. In the notation of [LSV05b], let $v_0 = [L_0]$ be the fundamental vertex in $\mathcal{B}$, and let $\Omega$ be the set of its neighbors.

**Claim 6.6.** (i) For each set $X = A$ or $X = B$, the map $g \leftrightarrow g.v_0$ is a bijection between $X$ and $\Omega$.

(ii) The subgroups, $\Gamma$ and $\Gamma'$, acts simply transitively on the Bruhat-Tits tree.

(iii) Both subgroups, $\Gamma$ and $\Gamma'$, are normal in $G(R)$ and of index $2(q + 1)$.

(iv) If $\phi : G(R) \to PSL_2(q^e)$ is an epimorphism, then both subsets, $\phi(A)$ and $\phi(B)$, are symmetric set of generators for $PSL_2(q^e)$.

(v) If $\phi : G(R) \to PSL_2(q^e)$ is an epimorphism whose kernel is a congruence subgroup $G(R, \phi)$ of $G(R)$, then both Cayley graphs, $Cay(PSL_2(q^e), \phi(A))$ and $Cay(PSL_2(q^e), \phi(B))$, are Ramanujan $(q + 1)$-regular graphs.

**Proof.** (i) The claim for $A$ is [LSV05b, Proposition 4.3]. The claim for $B$ follows from the claim for $A$ and the identity $tb^{-1}.v_0 = t\alpha b^{-1}.v_0 = (\alpha b)(\alpha^{-1})(t\alpha^{-1}).v_0$. Now, $\alpha \in T$ and $T$ fixes $v_0$, so $\{tb^{-1}.v_0 | t \in T\} = \{t\alpha^{-1}.v_0 | t \in T\}$.

(ii) The transitivity claim for $\Gamma$ is [LSV05b, Proposition 4.5], which relies solely on the validity of claim (i) for the generating set $A$ of $\Gamma$, hence the same proof works also for $\Gamma'$. Moreover, the same proof can actually show that for any $n \in \mathbb{N}$, for any vertex $v$ of distance $n$ from $v_0$, there exists a reduced word $g = s_1 \cdots s_n \in \Gamma$ (resp. $\Gamma'$), $s_1, \ldots, s_n \in A$ (resp. $B$), such that $g.v_0 = v$. This proves that the action is also simply transitive since the number of vertices of distance $n$ is equal the number of reduced words of length $n$, for any $n \in \mathbb{N}$.

(iii) The claim for $\Gamma$ follows from [LSV05b, Propositions 4.9 and 3.5], and the same proof also works for $\Gamma'$. The fact that the index is $2(q + 1)$ follows also from the fact that $\Gamma'$ acts simply transitively on the Bruhat-Tits tree by (ii). Hence the index of $\Gamma'$ in $G(R)$ is equal to the order of the stabilizer of $v_0$ in $G(R)$, which by [LSV05b, Proposition 3.5], is of size $2(q + 1)$.

(iv) By (iii) both images, $\phi(\Gamma)$ and $\phi(\Gamma')$, are normal subgroups of index $\leq 2(q + 1)$ in $PSL_2(q^e)$, and since $PSL_2(q^e)$ is a simple group of size $\geq \frac{1}{2}(q+1)q(q-1) > 2(q+1)$, we get that $\phi(\Gamma) = PSL_2(q^e) = \phi(\Gamma')$.

(v) The claim for $Cay(PSL_2(q^e), \phi(A))$ is [LSV05b, Theorem 7.1], and the same proof holds also for $Cay(PSL_2(q^e), \phi(B))$. Another way to see this is to observe that both graphs are isomorphic to $G(R, \phi)\mathcal{B}$ and in particular they are isomorphic, so if one is Ramanujan so is the other.

\[\square\]

### 6.3 Proof of Lemma 5.2 and Degree Reduction

First we use the LSV generators constructed in the previous subsection to prove the following Lemma.

**Claim 6.7.** For any odd prime power $q$ there exist an explicit construction of an infinite family of finite groups $G_i = PSL_2(q^e)$, with two symmetric generating subsets $A_i, B_i$ of $G_i$, such that for each $i$, $|A_i| = |B_i| = q + 1$, condition (TNC) holds for $A_i$ and $B_i$, and
the Cayley graphs $\text{Cay}(G_i, A_i)$ and $\text{Cay}(G_i, B_i)$ are Ramanujan, in particular they are $\lambda$-expanders with $\lambda \leq 2(q + 1)^{-1/2}$.

Proof. From Claim 6.6 we get that for any $i$, there exists two symmetric generating subsets $A_i$ and $B_i$ of the finite group $G_i = \text{PSL}_2(q^i)$, both sets are of size $q + 1$, and the Cayley graphs $\text{Cay}(G_i, A_i)$ and $\text{Cay}(G_i, B_i)$ are both Ramanujan. By Claim 6.5, for any $i > 2$, the two sets $A_i$ and $B_i$ satisfy condition (TNC).

Next we prove the following degree reduction trick, which allows us to start with a $\lambda$-expander Cayley graph, and to remove a few elements from the generating set with only negligible effect on $\lambda$.

Claim 6.8. (i) Let $G$ be a finite group, let $S' \subset S$ be two symmetric subsets of $G$, and denote $\lambda = \lambda(\text{Cay}(G, S))$ and $\lambda' = \lambda(\text{Cay}(G, S'))$ the normalized second largest eigenvalues of the corresponding Cayley graphs. Then

$$\lambda' \leq \lambda + 2 \frac{|S \setminus S'|}{|S'|}.$$  

(ii) In particular, if $|S \setminus S'| \leq 2|S|^{1/2}$ and $2|S|^{-1/2} \leq \lambda \leq \frac{1}{3}$, then

$$\lambda' \leq 4\lambda.$$

Proof. (i) Let $M = M_{S'}$ and $M' = M_{S'}'$ be the adjacency matrices of $\text{Cay}(G, S)$ and $\text{Cay}(G, S')$, respectively. Since $\text{Cay}(G, S)$ is $|S|$-regular (resp. $\text{Cay}(G, S')$ is $|S'|$-regular), the largest eigenvalue of $M$ is $|S|$ (resp. $M'$ is $|S'|$), with corresponding eigenvector the constant function $1_G$. By the Courant-Fischer Formula we get that

$$\lambda \cdot |S| = \max_{0 \neq v \perp 1_G} \frac{v^t M v}{v^t v} \quad \text{and} \quad \lambda' \cdot |S'| = \max_{0 \neq v \perp 1_G} \frac{v^t M' v}{v^t v}.$$  

Now the matrix $M - M'$ can be considered as the adjacency matrix of $\text{Cay}(G, S \setminus S')$, which by the Perron-Frobenius Theorem, all of its eigenvalues are bounded in absolute value by $|S \setminus S'|$, and by the Courant-Fischer Formula $|S \setminus S'| = \max_{0 \neq v} \frac{v^t (M' - M) v}{v^t v}$. Therefore we get that

$$\lambda' \cdot |S'| = \max_{0 \neq v \perp 1_G} \frac{v^t M' v}{v^t v} \leq \max_{0 \neq v \perp 1_G} \frac{v^t M v}{v^t v} + \max_{0 \neq v \perp 1_G} \frac{v^t (M' - M) v}{v^t v} \leq \lambda \cdot |S| + |S \setminus S'|,$$

and after dividing by $|S'|$ we get claim (i),

$$\lambda' \leq \lambda \cdot \frac{|S|}{|S'|} + \frac{|S \setminus S'|}{|S'|} \leq \lambda + (1 + \lambda) \frac{|S \setminus S'|}{|S'|} \leq \lambda + 2 \frac{|S \setminus S'|}{|S'|}.$$

(ii) First note that

$$|S \setminus S'| \leq 2|S|^{1/2} \leq \lambda |S| \leq \lambda |S'| + \lambda |S \setminus S'| \quad \Rightarrow \quad \frac{|S \setminus S'|}{|S'|} \leq \frac{\lambda}{1 - \lambda} \leq \frac{3\lambda}{2}.$$  

Combined with claim (i) we get claim (ii).  

Finally we combine the above two Claims to prove Lemma 5.2.
Lemma (Restatement of Lemma 5.2). Let $d_0, D_0 \in \mathbb{N}$. Let $q$ be any odd prime power such that $q \geq \max\{2d_0^2, D_0, 17\}$ and define $D = d_0 \cdot \lfloor \frac{q+1}{d_0} \rfloor$. Then there exist an explicit construction of an infinite family of finite groups $G_i = \text{PSL}_2(q^i)$, with two symmetric generating subsets $A_i, B_i \subset G_i$, such that for each $i$, both $A_i$ and $B_i$ are of size $D$ hence divisible by $d_0$, $A_i$ and $B_i$ satisfy condition (TNC), and the Cayley graphs $\text{Cay}(G_i, A_i)$ and $\text{Cay}(G_i, B_i)$ are $\lambda$-expanders where $\lambda \leq 8D^{-1/2}$.

Proof of Lemma 5.2. First note that $D$ is by definition the largest integer $\leq q+1$ which is divisible by $d_0$, and that $q + 1 - D \leq d_0 \leq \frac{1}{2} \sqrt{D}$.

By Claim 6.7, for each $i$, there exist $\tilde{A}_i, \tilde{B}_i$ two symmetric generating subsets of $G_i = \text{PSL}_2(q^i)$, such that $\tilde{A}_i, \tilde{B}_i$ are both of size $q+1$, they satisfy (TNC) and such that the corresponding Cayley graphs are Ramanujan, i.e. $\lambda$-expanders for $\lambda \leq \frac{2\sqrt{q}}{q+1} \leq 2(q+1)^{-1/2}$. Since $q \geq 17$, then $\lambda \leq \frac{1}{3}$.

Let $A_i \subset \tilde{A}_i$ and $B_i \subset \tilde{B}_i$ be any two symmetric subsets of size $D$. Since $\tilde{A}_i$ and $\tilde{B}_i$ satisfy (TNC), any subsets of them must also satisfy (TNC).

By Claim 6.8, we get that for $G = G_i$, $S = A_i$ or $B_i$, and $S' = A_i$ or $B_i$, respectively, we get that

$$\lambda(\text{Cay}(G, S')) \leq 4\lambda(\text{Cay}(G, S)) \leq 8D^{-1/2},$$

which completes the proof of the Lemma. □

Acknowledgements

We wish to thank Prahladh Harsha and Avi Wigderson for many interesting discussions along the way of this project. We also wish to thank Tali Kaufman for her influential role in connecting LTCs and high dimensional expansion.

This work was presented by the first author on October 6, 2021 at the Simon’s Institute for the Theory of Computing [Din21a] as part of the lecture series on breakthroughs in computer science, and at the Institute for Advanced Study in Princeton on October 25-26, 2021 [Din21b]. It was also presented by the fourth author on October 27, 2021 at the Simon’s HDX21 workshop [Lub21]. The authors are very grateful to these institutions and for the remarks of the audience which improved the exposition of the paper.

Irit Dinur acknowledges support by ERC grant 772839 and ISF grant 2073/21. Shai Evra is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship. Alexander Lubotzky’s research is supported by a grant from the Institute for Advanced Study at Princeton and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 882751). Shahar Mozes acknowledges support by ISF-Moked grant 2019/19.

References


[DSW06] Irit Dinur, Madhu Sudan, and Avi Wigderson. Robust local testability of tensor products of LDPC codes. In Proc. 10th International Workshop on Randomization and Computation (RANDOM), 2006. 3, 8, 23, 24


[GM12] Oded Goldreich and Or Meir. The tensor product of two good codes is not necessarily robustly testable. Information Processing Letters, 112(8-9), 2012. 8


[Mum79] David Mumford. An algebraic surface with k ample, $(K^2) = 9, p_g = q = 0$. *American Journal of Mathematics*, 101, 02 1979. 6


[PK21b] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes, 2021. 7


A Robust Testability and Agreement Testability

In this section we show the equivalence between the two notions, proving Lemma 2.9

Claim A.1 (Robust testability implies agreement testability). Assume $\delta_i = \text{dist}(C_i)$ for $i = 1, 2$. If $C_1 \otimes C_2$ is $\tau$-robustly testable then $C_1 \otimes C_2$ is $\kappa$-agreement testable, for

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

Proof. Suppose $w_1 \in C_1 \otimes F_2^{n_1}$, and $w_2 \in F_2^{n_1} \otimes C_2$. Let $f = w_1$, so $\delta^{\text{col}}(f) = 0$, and observe that since $w_2(i, \cdot) \in C_2$ for each $j$,

$$\delta^{\text{row}}(f) = \mathbb{E}_{i \in [n_1]} \text{dist}(f(i, \cdot), C_2) \leq \mathbb{E}_{i \in [n_1]} \text{dist}(f(i, \cdot), w_2(i, \cdot)) = \text{dist}(w_1, w_2).$$

By the robust testability of $C_1 \otimes C_2$ there is some $w \in C_1 \otimes C_2$ such that

$$\text{dist}(w, w_1) = \text{dist}(w, f) \leq \frac{1}{\tau} \cdot \frac{\delta^{\text{row}}(f) + \delta^{\text{col}}(f)}{2} \leq \frac{1}{2\tau} \cdot (\text{dist}(w_1, w_2) + 0).$$

By the triangle inequality $\text{dist}(w, w_2) \leq \text{dist}(w, w_1) + \text{dist}(w_1, w_2) \leq (1 + \frac{1}{2\tau}) \text{dist}(w_1, w_2)$.

Next, observe that $P_j[w(i, \cdot) \neq w_1(i, \cdot)] \cdot \delta_1 \leq \text{dist}(w_1, w)$, and similarly $P_i[w(i, \cdot) \neq w_2(i, \cdot)] \cdot \delta_2 \leq \text{dist}(w, w_2)$. Altogether,

$$P_j[w(i, \cdot) \neq w_1(i, \cdot)] + P_i[w(i, \cdot) \neq w_2(i, \cdot)] \leq \frac{1}{\delta_1} \text{dist}(w, w_1) + \frac{1}{\delta_2} \text{dist}(w, w_2) \leq \left(\frac{1}{2\tau \delta_1} + \frac{1 + 1/(2\tau)}{\delta_2}\right) \cdot \text{dist}(w_1, w_2)$$

proving the claim with $\kappa^{-1} = \frac{1}{2\tau \delta_1} + \frac{1 + 1/(2\tau)}{\delta_2}$, or $\kappa = \frac{2\tau \delta_1 \delta_2}{\delta_2 + \delta_1 (1 + 2\tau)}$. \hfill \Box

Note that in case $\delta_1 = \delta_2 = \delta$ the statement simplifies slightly to $\kappa = \frac{\tau \delta}{\tau + 1}$. The other direction, that we do not need here, is even simpler.

Claim A.2 (Agreement testability implies robust testability). If $C_1 \otimes C_2$ is $\kappa$-agreement testable, then $C_1 \otimes C_2$ is $\tau$-robustly testable for $\tau = \frac{\kappa}{2\tau + 1}$.

36
Proof. Assume \( C_1 \otimes C_2 \) is \( \kappa \)-agreement testable. Let \( w \in F_2^{n_1 \times n_2} \) satisfy \( \delta(w) = \frac{\delta_{\text{col}}(w) + \delta_{\text{row}}(w)}{2} \). Let \( w_1 \in C_1 \otimes F_2^{n_2} \) be such that \( \delta_{\text{col}}(w_1) = \text{dist}(w, w_1) \). Let \( w_2 \in F_2^{n_1} \otimes C_2 \) be such that \( \delta_{\text{row}}(w_2) = \text{dist}(w, w_2) \). By the triangle inequality,

\[
\text{dist}(w_1, w_2) \leq \text{dist}(w_1, w) + \text{dist}(w, w_2) = \delta_{\text{col}}(w) + \delta_{\text{row}}(w) = 2\delta(w).
\]

By the \( \kappa \)-agreement testability there is some \( w' \in C_1 \otimes C_2 \) such that

\[
\kappa \cdot (\mathbb{P}_{i,j}[w_1(i, j) \neq w'(i, j)] + \mathbb{P}_{i,j}[w_2(i, j) \neq w'(i, j)]) \leq \mathbb{P}_{i,j}[w_1(i, j) \neq w_2(i, j)] = \text{dist}(w_1, w_2) \leq 2\delta(w).
\]

But clearly

\[
\text{dist}(w_1, w') + \text{dist}(w', w_2) \leq \mathbb{P}_{i,j}[w_1(i, j) \neq w'(i, j)] + \mathbb{P}_{i,j}[w_2(i, j) \neq w'(i, j)] \quad (A.1)
\]

so again by the triangle inequality,

\[
\text{dist}(w, w') \leq \frac{1}{2} (\text{dist}(w, w_1) + \text{dist}(w_1, w') + \text{dist}(w, w_2) + \text{dist}(w_2, w'))
\]

\[
= \frac{1}{2} (\text{dist}(w, w_1) + \text{dist}(w, w_2) + \text{dist}(w_1, w') + \text{dist}(w_2, w'))
\]

\[
\leq \delta(w) + \kappa^{-1} \cdot \delta(w) = \frac{\kappa + 1}{\kappa} \cdot \delta(w).
\]

\( \square \)