Electronic Colloquium on Computational Complexity, Report No. 204 (2026)

Good Quantum Locally Testable Codes from Product Expansion

Mitali Bafna[1*] Anqi Li[1†] Quynh T. Nguyen[2‡]

[1]University of Washington [2]UC Berkeley

Abstract

We construct quantum locally testable codes (LTCs) with constant rate, distance, soundness and locality under a variant of a product expansion conjecture of Bafna and Vyas [BV26] about Reed-Solomon codes. In particular, we use the high-dimensional expansion framework of Dinur, Lin and Vidick [DLV24] for constructing quantum LTCs, instantiated with the non-Abelian cubical complexes of [RSV19]. Our code is obtained by equipping the complex with carefully chosen Reed-Solomon local codes whose symmetries are compatible with those of the complex.

Statement of AI Use: Motivated by the new conjectural construction of product expanding Reed-Solomon codes, the authors prompted GPT 6 to construct quantum LTCs using the framework of [DLV24], along with the non-Abelian cubical complexes considered in [HLM$^+$25], assuming the product expansion conjecture of [BV26]. The model provided a construction that uses using a variant of the conjecture of [BV26]. We spent significant effort in understanding the draft, extracting out the main ideas for our exposition and tracing many of the ideas to their origin in the literature. Upon further prompts, GPT 6 also provided a proof of the product expansion conjecture, but we did not verify its correctness, and hence have not included it in our manuscript. We are happy to share it with anyone who is interested.

1 Introduction

Locally testable codes are error correcting codes that have a local tester that can probabilistically check whether a word is in the code using only few queries.

Definition 1.1 (Locally testable codes)**.** *For $\kappa > 0$ and $q \in \mathbb{N}$, a linear code $C \subseteq \mathbb{F}^n$ is* κ-locally testable with q queries *if it is defined by a collection of parity checks, each involving at most q coordinates, such that every codeword is accepted with probability 1 and every word $f \in \mathbb{F}^n$ at relative distance Δ from C, violates at least a $\kappa\,\Delta$-fraction of the checks.*

These originated in the celebrated works on program checking [BLR93] and probabilistically checkable proofs [BFLS91, ALM$^+$98, AS98], in the form of low-degree testing of polynomials. Goldreich and Sudan [GS02] initiated a systematic study of LTCs, following which a long line of work [GS02, BSVW03,

*mitalib@cs.washington.edu
†aqli@cs.washington.edu
‡thequynhqb@pm.me

ISSN 1433-8092

BGH$^+$06, BS08, Din07, KMRZS17, GKO$^+$18] yielded constructions with *nearly optimal* parameters. Although, achieving 'good' LTCs, those with constant rate, relative distance, and locality simultaneously, remained open.

In a parallel vein, quantum low-density parity-check (qLDPC) codes with good rate and distance have long been sought for their potential in performing efficient and fault-tolerant quantum computation [Got14]. A long line of work [TZ14,EKZ20,HHO21,PK22b,BE21] developed constructions with increasingly strong parameters.

The seminal works of [DEL$^+$22] and [PK22a] resolved the questions of constructing good LTCs and good quantum LDPC codes respectively, using a novel method based on high-dimensional expansion. These build upon the expander codes of [SS94], that constructed good classical LDPC codes by defining a codespace on the edges of an expander graph by adding parity checks using good **'local'** codes at every vertex (found by brute force). They showed how the distance of the local codes translates to distance of the global code using graph expansion. These new works instead used a *cubical complex*, which can be thought of as a graph that contains not just edges, but also squares (or 2-dimensional faces). They define a global code on the squares, by imposing local codes on the edges in a way so that the local view at a vertex is a tensor codeword. Here they used special local codes so that the resulting tensor codes has not only have good rate and distance, but also is 'robustly' locally testable. They then lifted this local property to local testability of the full code building upon the local-to-global theorems of [KKL14, EK16]. We will refer to this property of the local codes as **product expansion**, as coined by [PK22a, KP25] (see Definitions 2.3 and 5.5).

This brings us to quantum LTCs that were introduced by Aharonov and Eldar [AE15] in connection with the quantum PCP conjecture [AN02, AAV13], formally defined as,

Definition 1.2 (Quantum locally testable CSS codes)**.** *A CSS code $\mathcal{Q} = \mathrm{CSS}(C_X, C_Z)$ is specified by two linear codes $C_X, C_Z \subseteq \mathbb{F}^n$ satisfying $C_X^\perp \subseteq C_Z$. The parity checks defining C_X and C_Z give its X- and Z-type checks, respectively. For $\kappa > 0$ and $q \in \mathbb{N}$, we call the code $\mathcal{Q}$* κ-locally testable with q queries *if both C_X and C_Z are κ-locally testable with q queries.*

Constructions of qLTCs with progressively improved parameters were obtained in [Has16, LLZ22, CHN$^+$24], and the currently best known parameters are by Dinur, Lin, and Vidick [DLV24] who constructed almost-good qLTCs, achieving constant rate and locality with inverse-polylogarithmic relative distance and soundness. Their construction uses the product expansion of *random codes* proved by [KP25]. These almost-good qLTCs have been used in low-overhead fault-tolerant quantum computation [NP25]. Analogous to classical LTCs, qLTCs have implications for the quantum PCP conjecture and related problems: they imply the NLTS theorem [EH17, ABN23], quantum interactive oracle proofs [SV26] and quantum multiparty computation as well as a 'gap amplification' procedure for quantum CSPs in recent work by two of the authors [BNZ26]. In all of these applications, good qLTCs yield better parameters than almost-good qLTCs.

1.1 Our Work

Motivated by the quantum PCP conjecture, a recent work by [BV26] constructed 'private' PCPs and enroute to that gave a construction of improved quantum codes supporting multiplication. This result relies on a conjecture about the product expansion of carefully chosen Reed-Solomon (RS) codes (see Section 2.4.1),

that if true would be the first structured codes apart from random codes that would satisfy this property. They used this conjecture to construct tensor product RS codes (see Corollary 4.5 therein) which give rise to quantum codes with near-linear rate and distance, sparse X checks giving local testability, but having dense Z checks.

A natural question that arises is whether one can use these structured RS codes as local codes instead in the high-dimensional framework of [DLV24] to obtain better parameters. Indeed, doing so we are able to build good qLTCs under a variant of the product expansion conjecture of [BV26],

Theorem 1.3 (Informal (see Theorem 5.9))**.** *Assuming Conjecture 5.6, there exists an explicit family of quantum locally testable CSS codes over a binary alphabet with constant rate, quantum distance, soundness and locality.*

We cross the polylog-bottleneck in [DLV24] by using *non-Abelian* cubical complexes instead of Abelian ones. Using these introduces the difficulty that the local views are tensor codewords only if the symmetries of the local code play well with the complex used, in particular, one can no longer use random codes. This fact was already observed by [KP25], where they suggested that one might be able to use the non-Abelian complexes of Rungtanapirom–Stix–Vdovina [RSV19] (RSV), along with local RS codes to build quantum LTCs. It is also very similar to the approach that is discussed in [DEL$^+$22], which they did not carry out for their final result. Our main work goes into showing that an appropriate choice of RS codes that satisfies our product expansion conjecture ensures that every local view in the RSV complex is a tensor codeword. The DLV framework then applies automatically, and one can lift the product expansion of the local codes, to local testability of the global quantum code defined on the complex.

Future Directions: Beyond codes, the HDX framework has led to many other applications in TCS such as sampling algorithms [ALG20], property testing [BLM24, DDL24], PCPs [BMVY25] and the construction of the first explicit lossless vertex expanders using the RSV complexes [HLM$^+$25], which served as an inspiration for this work. We believe our work opens up exciting avenues for further applications. We use certain properties of algebraic HDX constructions (see Lemma 4.1, property (4)) that to our knowledge have not yet seen applications. An immediate open question is be to obtain better qLTC and qLDPC codes that support multiplication [DLZ25, GL25], and potentially lead to advances in classical and quantum PCP constructions.

2 Technical Overview

We start by explaining the key ideas in the framework of [DEL$^+$22] for constructing LTCs from cubical complexes[1]. We then discuss the abstract framework of [DLV24] that obtains quantum LTCs from 4-dimensional cubical complexes by generalizing their ideas, albeit runs into a 'polylog-bottleneck'. We then discuss how to overcome this bottleneck using codes with symmetries that are 'compatible' with the symmetries of the complex used. Finally we explain our construction based on the RSV complexes and Reed-Solomon codes as local codes over them, as well as some intuition for their conjectured product expansion.

[1]These ideas are quite similar to the quantum LDPC codes construction of [PK22a], but we choose to focus on classical LTCs.

2.1 LTCs from Left-Right Cayley Complexes

We start by giving an exposition of the classical LTCs of [DEL$^+$22] that used the *Left-Right* Cayley complexes.

Definition 2.1 (2D-Left-Right Cayley Complex)**.** *Let G be a finite group and let $A, B \subseteq G$ be nonempty sets closed under inverses. The vertex set is $V = G \times \{0,1\}^2$ and for $g \in G$ and $i, j \in \{0,1\}$, an A-edge joins (g, i, j) to $(ga, i, 1-j)$ for each $a \in A$, i.e. it acts by right multiplication and a B-edge joins it to $(bg, 1-i, j)$ for each $b \in B$, i.e. acts by left multiplication. Define the set of squares by*

$$\mathcal{Q} = \left\{ \begin{array}{ccc} (bg,10) & \xrightarrow{\quad a \quad} & (bga,11) \\ {\scriptstyle b}\uparrow & & \uparrow{\scriptstyle b} \\ (g,00) & \xrightarrow[\quad a \quad]{} & (ga,01), \end{array} \;\middle|\; g \in G,\ a \in A,\ b \in B \right\}.$$

Defining the Code on the Complex: Fix orderings $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_n)$, a finite field $\mathbb{E}$, and let linear codes $C_A \subseteq \mathbb{E}^A$, $C_B \subseteq \mathbb{E}^B$ with the specified ordering over A, B. A word $z \in \mathbb{E}^{\mathcal{Q}}$ assigns one field element $z(Q)$ to every square. We then define the code $\mathcal{C}$ by demanding that the squares around every A-edge $((g,00),(ga,01))$ or $((g,10),(ga,11))$, obtained by varying b in the specified order, lie in C_B and similarly the squares around any B-edge lie in C_A:

$$\mathcal{C} = \left\{ z \in \mathbb{E}^{\mathcal{Q}} : \begin{array}{ll} (z(Q_e(b_1)), \ldots, z(Q_e(b_n))) \in C_B & \text{for every } A\text{-edge } e, \\ (z(Q_e(a_1)), \ldots, z(Q_e(a_m))) \in C_A & \text{for every } B\text{-edge } e \end{array} \right\},$$

where $Q_e(\cdot)$ denotes the square obtained on the edge e by moving via the specified element.

Local Views at Vertices: Take any codeword $z \in \mathcal{C}$ and fix a vertex v. Let $Q_v(a,b) \in \mathcal{Q}$ be the unique square containing the A-edge and B-edge incident to v with labels a, b at v. The squares around v can be arranged into an $m \times n$ matrix Z_v with $Z_v(a,b) := z(Q_v(a,b)), a \in A, b \in B$, which we will refer to as the *local view* of v. The row labeled by a contains exactly the symbols on squares containing the A-edge e incident to v with label a and the same holds for columns labeled by B. Thus the word z restricted to this local view belongs to $C_A \otimes C_B$.

Deriving the rate and local testability of $\mathcal{C}$ for an appropriate choice of C_A, C_B: A simple constraint counting argument implies that the rate of $\mathcal{C}$ equals $2\mathrm{rate}(C_A) + 2\mathrm{rate}(C_B) - 3$. Thus to get non-zero rate for $\mathcal{C}$ via counting, the sum of their rates must be larger than $3/2$.

Secondly, consider the set of parity checks given by picking a vertex, and testing whether the squares around it belong to the code $C_A \otimes C_B$. We have shown above that all codewords satisfy this property. A tensor code is locally testable via the natural row and column checks, but if it is additionally 'robustly' testable via these checks, or equivalently has product expansion, the works [DEL$^+$22, PK22a] show that one can lift this property to the full code, i.e. $\mathcal{C}$ itself is locally testable via these vertex-based parity checks! We discuss product expansion later on, but for now it suffices to think of this as an abstract property we will want our local codes to satisfy in order to build LTCs from this framework.

2.2 Generalizing to Higher Dimensions

The work of [DLV24] (DLV) generalized this framework to show that '4-dimensional' Cayley complexes yield quantum LTCs, in a similar way by defining local codes on faces. Although there seems to be no way of generalizing Left-Right non-Abelian Cayley complexes to higher dimensions, they obtained higher dimensional complexes by moving to Abelian Cayley complexes, where we have 4 generating sets, and each of them acts by right multiplication. At a high level, they require the following properties from a 4-dimensional Cayley complex,

1. The underlying Cayley graphs in each direction should be a good spectral expander, i.e. the normalized adjacency matrix should have second largest eigenvalue at most some small constant.
2. A 4-tuple of codes $(C_i)_{i\in 4}$, which will serve as our constant-sized local codes. We require the rates of these codes to be close to 1 and that they satisfy 4-dimensional product expansion[2].
3. The property that defining a code $\mathcal{C}$ by adding constraints on the (local views of the co-dimensional 1 faces of the) complex according to the codes C_i (or their duals) satisfies that the local view of every face in the complex is a tensor codeword; this is the higher-dimensional analogue of the local view at a vertex being the tensor codeword $C_A \otimes C_B$.

Expansion Bottleneck: It is a well-known fact that Abelian Cayley graphs on n vertices need $\Theta(\log n)$ generators to achieve constant spectral gap. This parameter is essentially responsible for the $\log n$-losses in the soundness/query-complexity of the quantum LTCs of DLV.

2.3 Constructing LTCs over Right-Right Non-Abelian Cayley Complexes

Thus to construct good quantum LTCs we need to move to non-Abelian Cayley complexes under *right* multiplication, which also has a natural analogue in higher dimensions (see Definition 3.1). The main difficulty is already present in using these complexes in 2 dimensions for constructing good classical LTCs, thus we only focus on this for the overview.

Definition 2.2 (2D-Right-Right-Cayley Complex)**.** *Again we have a set of vertices $V = G \times \{0,1\}^2$ and two generating sets A, B that are closed under inverse but unlike the Left-Right Cayley complex, both the set of edges act by right multiplication: an A-edge joins (g,i,j) to $(ga, i, 1-j)$ for each $a \in A$ and a B-edge joins it to $(gb, 1-i, j)$ for each $b \in B$.*

To get a well-defined set of squares, we additionally require that $AB = BA$ and $|AB| = |A||B|$. This implies that, for each $a \in A, b \in B$, there are unique $b' \in B, a' \in A$ with $ab' = ba'$, which can be represented by the permutations $\beta_a(b) = b'$ and $\beta_b(a) = a'$, so that we have the set of squares,

$$\mathcal{Q} = \left\{ \begin{array}{ccc} (gb,10) & \xrightarrow{\beta_b(a)} & (ga\beta_a(b),11) \\ {\scriptstyle b}\uparrow & & \uparrow{\scriptstyle \beta_a(b)} \\ (g,00) & \xrightarrow[a]{} & (ga,01), \end{array} \;\middle|\; g \in G,\ a \in A,\ b \in B \right\}.$$

[2]Technically, one pair needs to have rates close to 1, and the other needs to have rates close to 0, and we also need $(C_i^{\perp})$ to satisfy product expansion.

The global code: Defining a code $\mathcal{C}$ on this complex is more subtle than in the Left-Right case – given a fixed edge, say $((g, 00), (ga, 01))$ enumerating the squares touching it using $(g, 00)$ in the fixed order over B, is not the same as enumerating them from $(ga, 01)$. Thus, for an A-edge e, we choose the endpoint whose second bit is 0 and enumerate the squares in the prescribed order $b_1, \ldots, b_n$; denote the resulting squares by $Q_e(b_1), \ldots, Q_e(b_n)$. For a B-edge e, we enumerate them from the endpoint whose first bit is 0, using the A-edges in the order $a_1, \ldots, a_m$.

Local Views at a Vertex under Symmetry Assumptions on the Local Codes: Again we fix $z \in \mathcal{C}$ and a vertex v, and look at its local view given by the matrix,

$$Z_v(a, b) := z(Q_v(a, b)), \qquad a \in A, b \in B.$$

One can check that if v has type 00, then just like the case of the Left-Right Cayley construction, each row of Z_v belongs to C_B and column belongs to C_A, i.e. $Z_v \in C_A \otimes C_B$. But consider a vertex v of type 01. Each column of Z_v will belong to C_A, but one can check that each row a will be the permuted codeword,

$$u \circ \beta_a = \big(u_{\beta_a(b_1)}, \ldots, u_{\beta_a(b_n)}\big),$$

for some $u \in C_B$. The analogous property holds for vertices of types 10 and 11.

If our codes C_A, C_B are invariant under these permutations, i.e. $u \circ \beta_a \in C_B$ for all $u \in C_B$, and $u \circ \beta_b \in C_A$ for all $u \in C_A$, then we will get that each vertex view belongs to $C_A \otimes C_B$.

2.4 Our Construction in 2-Dimensions: Complex and Local Codes

Thus our task reduces to choose local codes that are invariant under the maps β of the complex. Our construction builds on the ideas suggested in [DEL$^+$22], in their abandoned approach, and [PK24, footnote 36]. They noticed that projective Reed-Solomon codes are natural choices for local codes, along with the complexes of [RSV19] (or other similar algebraic complexes). In more detail, the vertices of the RSV complexes is $G = \mathrm{PGL}_2(\mathbb{F}_{2^r})$, the group of invertible matrices over an extension field $\mathbb{E} = \mathbb{F}_{2^r}$[3]. As in the constructions of Ramanujan graphs [LPS88] (and complexes [LSV05]), the A-edge (analogously for B-edges) around each vertex can be labeled by the set of *projective lines* or elements of $\mathbb{P}^1(\mathbb{F}_{s_A}) = \mathbb{F}_{s_A} \cup \infty$, where s_A is some power of 2. This set can alternatively be viewed as a set of 2-dimensional vectors: $a \to (a, 1)^T$ for $a \in \mathbb{F}_{s_A}$ and $a \to (1, 0)^T$ if $a = \infty$, which we refer to as r_a. For each fixed b, each β_b permutation over the labels of A can then be implemented using a 2×2 invertible matrix B_b: $B_b r_a = \lambda_a r_{\beta_b(a)}$ for a nonzero scalar λ_a. This symmetry property plays well with projective RS (PRS) codes, which is the set of homogenous bivariate degree d polynomials with coefficients in $\mathbb{E}$, evaluated over $\mathbb{P}^1(\mathbb{F})$. For any such polynomial p, we have that,

$$p \circ \beta_b(r_a) = p(r_{\beta_b(a)}) = p(\lambda_a^{-1} B_b r_a) = \lambda_a^{-d} q(r_a),$$

for some degree d polynomial $q = p \circ B_b$. Note that if the scaling factor in front of q did not exist, we would get that $p \circ \beta_b$ belongs to the code (and the same holds for the β_a maps), and then by the argument above each vertex view would be a tensor PRS codeword.

[3]Precisely, the complex from [RSV19] turns out to have the vertex set being cosets of some G with respect to the subgroup $\mathrm{PGL}_2(\mathbb{F}_2)$, but we ignore this distinction for the overview.

That is, the main difficulty is how to handle these scaling factors. We deal with this issue using the fact that one can set up certain matrices on faces of the complex carefully, such that they are 'compatible' with the B_a, B_b matrices (see property (4) in Lemma 4.1). Then, using these matrices, we choose a scaling factor for each edge and define the constraint on it as a scaled PRS codes (thus each edge has a different constraint). Finally, we show that one can choose appropriate scalars per (v, Q) pair, so that the scaled code satisfies the fact that each local view around a vertex v belongs to a tensor PRS code (with coefficients in $\mathbb{E}$) evaluated over the domain $\mathbb{P}^1(\mathbb{F}_{s_A}) \times \mathbb{P}^1(\mathbb{F}_{s_B})$. We defer a more detailed discussion for the intuition here to Section 4.1.

2.4.1 Product Expansion of Reed-Solomon Codes

The second property we need from our pair of local codes (which in our case are PRS codes evaluated over $\mathbb{P}^1(\mathbb{F}_s)$), is that they satisfy product expansion (PE) in the high-rate regime. Let us define 2d-PE (see Definition 5.5 for the high-dimensional analogue): let $C_1 \subseteq \mathbb{E}^{S_1}$ and $C_2 \subseteq \mathbb{E}^{S_2}$ be linear codes, where $n_i = |S_i|$, and view a word on $S_1 \times S_2$ as a matrix. For a matrix M, let $\mathrm{supp}(M)$ be the set of its nonzero entries, and let $|M|_{\mathrm{col}}$ and $|M|_{\mathrm{row}}$ denote the number of its nonzero columns and rows, respectively.

Definition 2.3 (Two-dimensional product expansion)**.** *The pair (C_1, C_2) is ρ-product-expanding if every matrix M that is the sum of a matrix whose columns belong to C_1 and a matrix whose rows belong to C_2 admits such a decomposition $M = M_1 + M_2$ satisfying*

$$|\operatorname{supp}(M)| \geqslant \rho\big(n_1|M_1|_{\mathrm{col}} + n_2|M_2|_{\mathrm{row}}\big).^4$$

Polishchuk and Spielman [PS94] proved that any pair of Reed-Solomon codes evaluated over $\mathbb{F}_q^\star$ is product expanding with $\rho = \Omega(1)$ when the *sum* of their rates is *less than one*, which is a tight condition, and thus these codes cannot be used as local codes for LTCs.

To see why their sum-of-rates condition is tight, consider a pair of RS codes C_1, C_2 evaluated on S_1, S_2, respectively. Intuitively, the main geometric obstruction to product expansion without such a rate restriction is a low-degree bivariate curve $F(X, Y)$ containing too many points of $S_1 \times S_2$. For example, when $S_1 = S_2 = \mathbb{F}_q^\star$, the diagonal $X - Y = 0$ contains $q - 1$ grid points – an unbounded number despite its constant degree. This diagonal yields an obstruction to constant PE when the sum of the two rates reaches one.

The work of [BV26] avoids this issue by choosing S_1, S_2 carefully: they show that if the S_i are *multiplicative subgroups of coprime orders* of $\mathbb{F}_q^\star$, for prime q, not only does the above obstruction disappear (since $|S_1 \cap S_2| = 1$), but the pair has constant PE in the high-rate regime (with an analogous conjecture in higher dimensions). We need such a PE result for PRS codes; fortunately, one can show that a PRS code evaluated over $\mathbb{P}^1(\mathbb{F}_s)$ is equivalent to an RS code (of similar degree) evaluated over a multiplicative subgroup of $\mathbb{F}_{s^2}^\star$ of size $s + 1$ (see Lemma 5.7). We want our ambient field to contain $\mathbb{F}_{s^2}^\star$, which means we cannot take it to be of prime order. Thus we move to extension fields, and use the RSV complexes over $\mathbb{E} = \mathbb{F}_{2^r}$. One could hope that the same conjecture also holds for arbitrary coprime order multiplicative subgroups S_i of $\mathbb{E}^\star$. That turns out to be false due to low-degree curves that intersect $S_1 \times S_2$ in too many points. But restricting to a pair of coprime order subgroups whose order is of the form $2^m + 1$ seems to avoid such obstructions.

[4] This two-dimensional notion is equivalent to robust testability of the tensor code $C_1 \otimes C_2$. In higher dimensions (see Definition 5.5) product expansion is a stronger condition.

In conclusion, we conjecture this modified PE statement (see Conjecture 5.6) in high-dimensions and choose the 4-dimensional complex so that in each direction, the edge labels lie in $\mathbb{P}^1(\mathbb{F}_{s_i})$ where s_i is a power of 2. The local views at a vertex are then shown to be tensor products of PRS codes evaluated over $\mathbb{P}^1(\mathbb{F}_{s_i})$, which satisfy PE under our conjecture. This gives us all the properties required to get good qLTCs using DLV's framework.

3 Preliminaries

3.1 Cubical Complexes

Our complex will be obtained from the construction of [RSV19]. Instead of the Cayley complexes defined above, we will get a complex whose set of vertices contains the cosets of a group G with respect to a subgroup H, just like in the case of Schreier graphs that generalize Cayley graphs (the setting of $H = \{1\}$).

Definition 3.1 (A k-dimensional complex on cosets)**.** *Let H be a subgroup of a finite group G. A coset is a set $gH = \{gh : h \in H\}$, and G/H denotes the set of these cosets. Let $D_1, \ldots, D_k \subseteq G$ be nonempty subsets satisfy $D_i^{-1} = D_i$ and $HD_iH = D_i$. We write $D/H = \{dH : d \in D\}$, let $n_i = |D_i/H|$ and assume*

$$D_iD_j = D_jD_i \quad (i \neq j), \qquad |D_S/H| = \prod_{i \in S} n_i \quad (\varnothing \neq S \subseteq [k]), \tag{1}$$

where $D_S = \prod_{i \in S} D_i$. The vertices of the complex are given by $(gH, \mathbf{b}) \in (G/H) \times \{0,1\}^k$. The direction-$i$ edges are

$$\{(gH, \mathbf{b}), (gdH, \mathbf{b} + \mathbf{e}_i)\} \qquad (d \in D_i),$$

where $\mathbf{e}_i$ is the ith unit vector and addition is modulo two. For each nonempty $S \subseteq [k]$, the cubes in directions S are the $2^{|S|}$-sized vertex sets of the form,

$$\left\{ \left(g_T H, \mathbf{b} + \sum_{i \in T} \mathbf{e}_i \right) : T \subseteq S \right\}$$

where $\mathbf{b} \in \{0,1\}^k$, $g_T \in G$, and $g_T^{-1} g_{T \cup \{i\}} \in D_i$ for $T \subseteq S$ and $i \in S \setminus T$. Each such cube has dimension $|S|$, and its faces arise by fixing some of the type bits in S.

Note that the neighbor sets are independent of the representative g of gH, because $hD_i = D_i$ for every $h \in H$ and $i \in [k]$. Each vertex has n_i incident edges in direction i. The size condition in (1) says that each two-edge path in two distinct directions i then j is uniquely determined by its endpoints, and we also have a unique path in the reversed order. That is, by considering $gd_iH \to gH \to gd_jH$ and its reverse, we can identify fourth corner of the unique square containing two given edges at gH.

Definition 3.2. *Let X be a cubical complex with vertex type classes V_{type}, indexed by $\mathrm{type} \in \{0,1\}^4$. Every vertex has $n_i \geqslant 1$ neighbors in direction i, and direction-i edges flip only the ith type bit. For $\mathrm{type}_i = 0$, we write $X_{i,\mathrm{type}}$ for the n_i-regular bipartite graph consisting of these edges between V_{type} and $V_{\mathrm{type}+\mathbf{e}_i}$, where $\mathbf{e}_i$ is the ith unit vector and addition is modulo two.*

For $0 \leqslant \lambda < 1$, we say that X is λ-expanding *if $\lambda_2(X_{i,\mathrm{type}}) \leqslant \lambda$ for every direction i and every type with $\mathrm{type}_i = 0$.*

3.2 Projective Reed-Solomon Codes

We define projective Reed–Solomon codes, identify their duals, and prove their symmetry under Möbius transformations.

Projective coordinates. Let s be a prime power and let $\mathbb{E}$ be a finite field containing $\mathbb{F}_s$. The projective line $\mathbb{P}^1(\mathbb{F}_s)$ is the set of one-dimensional subspaces of $\mathbb{F}_s^2$. Write $\langle r\rangle$ for the line spanned by a nonzero vector r. We identify its $s+1$ elements with $\mathbb{F}_s \cup \{\infty\}$ using the representatives

$$r_x = (x,1)^T \quad (x \in \mathbb{F}_s), \qquad r_\infty = (1,0)^T.$$

Write $\mathrm{GL}_2(\mathbb{F}_s)$ for the invertible 2×2 matrices, $\mathrm{SL}_2(\mathbb{F}_s)$ for its determinant-one subgroup, and $\mathrm{PGL}_2(\mathbb{F}_s)$ by identifying two matrices which differ by a scalar multiple.

Definition 3.3 (Projective Reed–Solomon code)**.** *For $0 \leqslant d \leqslant s-1$, define*

$$\mathrm{PRS}_d(\mathbb{F}_s;\mathbb{E}) = \left\{ (f(r_x))_{x\in\mathbb{P}^1(\mathbb{F}_s)} : f(X,Y) = \sum_{j=0}^{d} f_j X^j Y^{d-j}, \quad f_j \in \mathbb{E} \right\}.$$

We abbreviate this code by PRS_d when the fields are clear.

Thus finite coordinates are $f(x,1)$, and the coordinate at infinity is $f(1,0) = f_d$. It is well-known that the code has parameters $[s+1, d+1, s+1-d]_{\mathbb{E}}$.

Claim 3.4 (Dual of projective Reed-Solomon codes)**.** *For $0 \leqslant d \leqslant s-1$, we have $\mathrm{PRS}_d(\mathbb{F}_s;\mathbb{E})^\perp = \mathrm{PRS}_{s-1-d}(\mathbb{F}_s;\mathbb{E})$.*

Proof. Let f, g be homogeneous polynomials of degrees $d, s-1-d$. Their product has degree $s-1$, so the finite-field power sums give

$$\sum_{x\in\mathbb{F}_s} f(x,1)g(x,1) = -f(1,0)g(1,0).$$

Thus their evaluation vectors are orthogonal, including the coordinate at infinity. The two codes have dimensions $d+1$ and $s-d$, whose sum is $s+1$, proving equality. □

Möbius transformations. For $B = \begin{pmatrix} a & b \\ c & e \end{pmatrix} \in \mathrm{GL}_2(\mathbb{F}_s)$, let β_B be the permutation of projective coordinates induced by B,

$$\beta_B(x) = \langle Br_x\rangle, \qquad Br_x = \lambda_B(x) r_{\beta_B(x)}.$$

In scalar notation, $\beta_B(x) = (ax+b)/(cx+e)$ for finite x, with a zero denominator giving ∞; also $\beta_B(\infty) = a/c$ if $c \neq 0$, and ∞ otherwise. For finite x with $cx+e \neq 0$, the scalar is $\lambda_B(x) = cx+e$.

Claim 3.5 (Möbius symmetry)**.** *For every $B \in \mathrm{GL}_2(\mathbb{F}_s)$,*

$$\left\{ \left(\lambda_B(x)^d (u\circ\beta_B)_x\right)_x : u \in \mathrm{PRS}_d \right\} = \mathrm{PRS}_d,$$

where $(u\circ\beta_B)_x = u_{\beta_B(x)}$.

Proof. For $u = (f(r_x))_x$, we have $\lambda_B(x)^d (u\circ\beta_B)_x = f(\lambda_B(x) r_{\beta_B(x)}) = f(Br_x) =: (f\circ B)(r_x)$. Since B is invertible, $f \mapsto f\circ B$ is a bijection on homogeneous degree-d polynomials, proving the claim. □

4 The Cubical Complex and Local Codes

As discussed in the overview, the main contribution of our work is to set up local Reed-Solomon codes on the non-Abelian cubical complex of RSV, in a way such that all the local views are tensor codewords. We start by defining the properties of the complex that we will use, then describe how the local codes are assigned and finally show that all local views are tensor codewords.

4.1 The Complex and Its Symmetries

The following lemma describes the cubical complex and its symmetries. The proof is deferred to Section 6.

Lemma 4.1 (Cubical Complex Construction)**.** *Fix positive integers $m_1, \ldots, m_4$ divisible by 8, and set $s_i = 2^{m_i}$ for $i \in [4]$. For each such choice and arbitrarily large integers r, there is a finite cubical complex $X = \Gamma \backslash \mathcal{T}$, where $\Gamma \leqslant \mathcal{G}$ acts freely on vertices, with the following properties. Set $G = \mathrm{PGL}_2(\mathbb{F}_{2^r})$ and $H = \mathrm{PGL}_2(\mathbb{F}_2) \leqslant G$.*

1. *The vertices are $(G/H) \times \{0,1\}^4$. There are subsets $D_i \subset G$ satisfying $HD_i = D_iH = D_i$, $D_i^{-1} = D_i$, and $|D_i/H| = s_i + 1$. Writing $\mathbf{e}_i$ for the ith unit vector, the direction-i edges are*

$$(gH, \vec{b}) \ \sim_i \ (gdH, \vec{b} + \mathbf{e}_i), \qquad dH \in D_i/H. \tag{2}$$

   Here addition is modulo two. At each vertex, edges in distinct directions determine a unique cube. Squares and higher cubes are the images of product cubes in $\mathcal{T}$. Moreover, $D_iD_j = D_jD_i$ for $i \neq j$.

2. *At every vertex, the $s_i + 1$ incident direction-i edges are labeled by $A_i = \mathbb{P}^1(\mathbb{F}_{s_i})$. The type of a vertex $v = (gH, \vec{b})$ is $\mathrm{type}(v) = \vec{b}$.*

3. *Fix nonzero representatives $r_a^{(j)} \in \mathbb{F}_{s_j}^2$ for $a \in A_j$. For each directed direction-i edge $e : v \to w$ and each $j \neq i$, let $\beta_e^{(j)} : A_j \to A_j$ pair the direction-j edges opposite each other across e. There is a matrix $B_e^{(j)} \in \mathrm{SL}_2(\mathbb{F}_{s_j})$ such that*

$$\beta_e^{(j)}(a) = \langle B_e^{(j)} r_a^{(j)} \rangle.$$

   i

4. *For every vertex v of every 4-cube Q of X, one can choose matrices $F_i(v, Q) \in \mathrm{SL}_2(\mathbb{F}_{s_i})$, for $1 \leqslant i \leqslant 4$, such that:*

   (a) *If a_i labels the direction-i edge of Q at v, then column $\mathrm{type}_i(v) + 1$ of $F_i(v, Q)$ spans a_i.*

   (b) *For every direction-i edge $e : v \to w$, oriented so that $\mathrm{type}_i(v) = 0$ and $\mathrm{type}_i(w) = 1$, there are $P_{v,e}, P_{w,e} \in \mathrm{SL}_2(\mathbb{F}_{s_i})$, independent of Q, such that every $Q \supset e$ satisfies*

$$\begin{aligned} F_i(v, Q) &= P_{v,e} \begin{pmatrix} \eta & * \\ 0 & \eta^{-1} \end{pmatrix}, \quad F_i(w, Q) = P_{w,e} \begin{pmatrix} \eta & 0 \\ * & \eta^{-1} \end{pmatrix}, \\ F_j(w, Q) &= B_e^{(j)} F_j(v, Q) \qquad\qquad (j \neq i), \end{aligned} \tag{3}$$

   where $\eta \in \mathbb{F}_{s_i}^\times$ and the starred entries may depend on e, Q, and $B_e^{(j)}$ is the matrix from item 3.

The symmetry properties of the complex are crucial to ensure that the local views of vertices are consistent, thereby yielding a well-defined global code. In the rest of this section, we will show how to set up local codes on the complex, and then prove that the local views of vertices are tensor codewords.

4.2 Setting up Local Codes on the 2d Complex

The main idea of our construction can be described in two dimensions, with the higher-dimensional case following essentially the same calculations. Thus, we will focus on the 2d version of the complex in Lemma 4.1.

We associated each square with a scalar in a finite field $\mathbb{E}$. We equip this complex with a set of local codes, one for every edge in $X(2)$. The global code is defined as the set of all assignments of field elements to squares that satisfy the local constraints on edges. The edges' local codes are rescaled and permuted versions of two base projective Reed–Solomon codes, one for each direction.

Definition 4.2 (The Base Codes)**.** *Let $s_i = 2^{m_i}$, where $m_i \geqslant 1$ for $i \in \{1, 2\}$, and let $\mathbb{E}$ be a finite field containing both $\mathbb{F}_{s_1}$ and $\mathbb{F}_{s_2}$. Fix an integer $h \geqslant 0$ and, for each $i \in \{1, 2\}$, an integer d_i satisfying*

$$0 \leqslant d_i \leqslant s_i - 1, \qquad s_i - 1 \mid (2^h + 1)d_i.$$

Let $A_i = \mathbb{P}^1(\mathbb{F}_{s_i})$, with a fixed coordinate order, and define

$$C_i = \mathrm{PRS}_{d_i}(\mathbb{F}_{s_i}; \mathbb{E}) \subseteq \mathbb{E}^{A_i}, \qquad i \in \{1, 2\}.$$

We note that the divisibility condition on the degrees will be crucial.

Our goal is to design the edge local codes so that the local view around each vertex belongs to a tensor code. The main challenge for doing so is that, in our non-Abelian cubical complex, the two endpoints of an edge index its incident squares using their respective local labels that are not necessarily the same. Instead, the labels are related by the permutations β_e as specified by Lemma 4.1(3). If the base codes were invariant under the permutations β_e, these differences in ordering would be harmless. Projective Reed–Solomon codes instead have the *scaled* permutation symmetry of Claim 3.5. We therefore introduce rescaling weights, chosen so that one rescaling of each square symbol works simultaneously for its row and its column. Making these weights compatible at both endpoints of an edge requires an additional modification, which we resolve by introducing an edge-independent permutation. Fortunately, rescaling and permutation do not affect the product expansion of tensor codes, which is a property concerning Hamming weights only.

We now describe the permutations and rescaling coefficients that will be used to define the local codes on edges.

Fact 4.3. *On a finite field of characteristic two, the map $x \mapsto x^2$ is a field automorphism known as the Frobenius map.*

Edge-independent permutations. Let P be the h-fold iterate of the Frobenius automorphism: $x \mapsto x^{2^h}$. This induces a permutation of the projective coordinates A_i by permuting the field elements $\mathbb{F}_{s_i}$ and fixes ∞. We extend this naturally to define the permuted codes PC_i. So, e.g., PC_1 consists of the words $(f(r_{a^{2^h}}))_{a \in A_1}$, with f a polynomial of homogeneous degree d_1 (see Definition 3.3). The $\mathbb{F}_2$-linearity of the Frobenius map, $(x + y)^{2^h} = x^{2^h} + y^{2^h}$, will be crucial. Later, we will sometimes raise a vector or matrix to the 2^h-th power entrywise, we will be explicit when doing so.

Vertex weights. The rescaling coefficients are defined in terms of the matrices from Lemma 4.1(4) and depend on the type of the vertex and the square.

First, each vertex associates a weight to each incident square. We use labels $a \in A_1$ and $b \in A_2$. For a vertex v of type $t_1 t_2$ and a square $Q = Q_v(a, b)$, define $c_1(v, Q) \in \mathbb{F}_{s_1}^{\times}$ and $c_2(v, Q) \in \mathbb{F}_{s_2}^{\times}$ by

$$\begin{aligned} &\text{column } t_1 + 1 \text{ of } F_1(v, Q) = c_1(v, Q) r_a, \\ &\text{column } t_2 + 1 \text{ of } F_2(v, Q) = c_2(v, Q) r_b. \end{aligned} \tag{4}$$

These scalars can depend on the whole square Q, hence on both a and b. Define

$$\nu_v(Q) = \begin{cases} c_1(v,Q)^{d_1} c_2(v,Q)^{d_2}, & v \text{ of type } 00, \\ c_1(v,Q)^{d_1} c_2(v,Q)^{2^h d_2}, & v \text{ of type } 01, \\ c_1(v,Q)^{2^h d_1} c_2(v,Q)^{d_2}, & v \text{ of type } 10, \\ c_1(v,Q)^{2^h d_1} c_2(v,Q)^{2^h d_2}, & v \text{ of type } 11. \end{cases} \tag{5}$$

Local codes. We can now define the local codes on edges. Assign a symbol $z(Q) \in \mathbb{E}$ to every square Q. For a direction-i edge e, let its *designated endpoint* be the endpoint w with $\mathrm{type}(w)_i = 0$. The squares incident to e are denoted by $Q_e(b)$ where b is the label of the direction-j edge ($j \neq i$) specified by w that completes the square. The squares are ordered by the labels given by the designated endpoint. The local code enforced by edge e is C_j, but rescaled and permuted according to the rules specified below. Let D_e be the operation that multiplies the coordinate for each square $Q_e(b)$ by $\nu_w(Q_e(b))$. Write $D_e C = \{D_e c : c \in C\}$.

The following table specifies the local codes imposed on edges.

Direction	Endpoint types	Designated endpoint	Required edge code
1	$00 \leftrightarrow 10$	00	$D_e C_2$
1	$01 \leftrightarrow 11$	01	$D_e P C_2$
2	$00 \leftrightarrow 01$	00	$D_e C_1$
2	$10 \leftrightarrow 11$	10	$D_e P C_1$

Table 1: Local codes imposed on edges. Each direction-i edge has an designated endpoint of type $t_i = 0$. The rescaling weights D_e are specified by the designated endpoint. The permutaion P is applied when the designated endpoint has type $t_j = 1$ for the other direction $j \neq i$.

For example, on a direction-1 edge e between types $0t_2$ and $1t_2$, the check at e requires that there exists a homogeneous degree-d_2 polynomial f satisfies

$$z(Q_e(b)) = \nu_w(Q_e(b)) f(r_{b 2^h t_2}) \qquad (b \in A_2),$$

where w is the type-$0t_2$ endpoint of e.

Remark 4.4. We give some intuition for choosing the weights $\nu_v(Q)$ in (5) and the permutation P.

First consider the construction without the permutation P. Consider again an edge $e = (w, v)$ of type $00 \leftrightarrow 10$ and a square $Q = Q_e(b)$. The matrices in Lemma 4.1(4) provide the vectors $c_2(v, Q) r_{\beta_e(b)}$ and

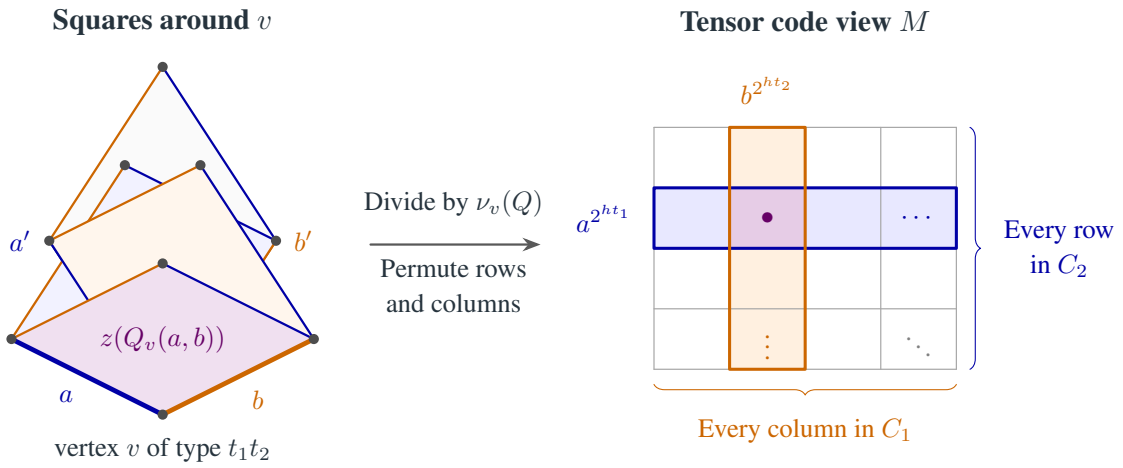


Figure 1: The local view of vertex v is a tensor code up to rescaling and permutations.

$c_2(w,Q)r_b$. Since a homogeneous degree-d polynomial satisfies $f(cr) = c^d f(r)$ for any scalar c, using the matrix identity in Lemma 4.1(4) gives

$$f(r_{\beta_e(b)}) = \left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{d_2} f(B_e r_b).$$

This suggests a weight contribution of the form $c_2(v,Q)^{d_2}$ at every vertex v. Applying the same intuition to direction-2 edges suggests another weight contribution of the form $c_1(v,Q)^{d_1}$, and thus an overall weight of $\nu_v(Q) = c_1(v,Q)^{d_1} c_2(v,Q)^{d_2}$ at every vertex v. However, this does not work because the direction-2 contribution $\frac{c_1(w,Q)^{d_1}}{c_1(v,Q)^{d_1}}$ can depend on Q and affect the direction-1 rescaling guarantee. We therefore want to modify the exponents so that the two direction rules agree.

The key observation is that, using the triangular forms of the F matrices in Lemma 4.1(4) we can infer $\frac{c_1(w,Q)^{d_1}}{c_1(v,Q)^{md_1}} \propto$ (some Q-dependent quantity)$^{(m+1)d_1}$. The latter becomes independent of Q if $(s_1 - 1) \mid (m+1)d_1$ by Fermat's little theorem. At vertex v, replacing an evaluation label x by x^m changes the associated scalar factor from $c_1(v,Q)^{d_1}$ to $c_1(v,Q)^{md_1}$. It turns out that we also want the map $x \mapsto x^m$ to be a field automorphism, and the Frobenius map provides such a map. Thus applying the permutation P at vertices where $t_i = 1$ together with rescaling weights $\nu_v(Q) = \prod_{i=1}^{2} c_i(v,Q)^{2^{ht_i} d_i}$ works.

4.3 Local Views of Vertices in the 2d Construction

In this section, we show that the edge constraints from the preceding subsection enforce that the local view of every vertex is a tensor code, up to rescaling and permuting coordinates.

Lemma 4.5 (Local views are Tensor Codes)**.** *Fix a vertex v of type t_1t_2. An assignment z to the squares containing v satisfies the checks in Table 1 on every edge incident to v if and only if there exists $M \in C_1 \otimes C_2$*

such that

$$z(Q_v(a,b)) = \nu_v(Q_v(a,b))\, M\big(a^{2^{ht_1}}, b^{2^{ht_2}}\big) \qquad (a \in A_1,\ b \in A_2).$$

Proof. We need to show that division by $\nu_v(Q)$ turns each row into a codeword in $P^{t_2}C_2$, and each column into a codeword in $P^{t_1}C_1$. See Figure 1.

First consider the rows. If $t_1 = 0$, the direction-1 edge checks are defined at v, so containment in $P^{t_2}C_2$ follows directly from Table 1. Suppose $t_1 = 1$, and consider a fixed edge $e = (w, v)$, so w has type $0t_2$ and is the designated endpoint of e. Let $B_e^{(2)}$ be the matrix from Lemma 4.1(3) and $\beta_e^{(2)}$ be the associated edge label permutation. Let us omit the direction superscript because there is only one direction transverse to e, i.e., we will write B_e, β_e. Consider a label $b \in A_2$ and the square completed by an edge labeled by b at w, $Q = Q_e(b)$. Let a_w, a_v be the labels of e at w, v, respectively, so $Q = Q_v(a_v, \beta_e(b)) = Q_w(a_w, b)$.

The edge check at e requires

$$z(Q) = \nu_w(Q) f(r_{b^{2^{ht_2}}}),$$

where f is homogeneous of degree d_2 independent of b.

Since w has type $0t_2$ and v has type $1t_2$, the weights according to (4), (5) are

$$\nu_w(Q) = c_1(w,Q)^{d_1} c_2(w,Q)^{2^{ht_2} d_2}, \qquad \nu_v(Q) = c_1(v,Q)^{2^h d_1} c_2(v,Q)^{2^{ht_2} d_2}.$$

So the rescaled entry is

$$\frac{z(Q)}{\nu_v(Q)} = \frac{c_1(w,Q)^{d_1}}{c_1(v,Q)^{2^h d_1}} \left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{2^{ht_2} d_2} f(r_{b^{2^{ht_2}}}) \tag{6}$$

We will show the following two claims:

Claim 4.6 (Direction-1 Contribution is b-independent)**.** $\frac{c_1(w,Q)^{d_1}}{c_1(v,Q)^{2^h d_1}} = \alpha_e$ *independent of* Q.

Claim 4.7 (Direction-2 Contribution is a Permuted Codeword)**.** $\left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{2^{ht_2} d_2} f(r_{b^{2^{ht_2}}}) = f'(r_{\beta_e(b)^{2^{ht_2}}})$, *where* f' *is a homogeneous degree-*d_2 *polynomial.*

We first use the claims to conclude the lemma before proving the claims.

Substituting the claims into (6) gives

$$\frac{z(Q)}{\nu_v(Q)} = \alpha_e f'(r_{\beta_e(b)^{2^{ht_2}}}) =: g(r_{\beta_e(b)^{2^{ht_2}}}),$$

where g is a homogeneous degree-d_2 polynomial. Recall that $Q = Q_e(b) = Q_v(a_v, \beta_e(b))$, hence changing the variable name $\beta_e(b) \mapsto b$ gives

$$\frac{z(Q_v(a_v,b))}{\nu_v(Q_v(a_v,b))} = g(r_{b^{2^{ht_2}}}).$$

Therefore,

$$M\big(a_v^{2^{ht_1}}, b^{2^{ht_2}}\big) = \frac{z(Q_v(a_v,b))}{\nu_v(Q_v(a_v,b))} = g(r_{b^{2^{ht_2}}}).$$

Since $b \mapsto b^{2^{ht_2}}$ permutes A_2, this row of M belongs to C_2.

Applying the argument to every direction-1 edge gives all row constraints. Exchanging directions gives all column constraints. We conclude that $M \in C_1 \otimes C_2$.

We now prove the claims.

Proof of Claim 4.6. This proof is where our choice of permutation P and the degree divisibility condition in Definition 4.2 are used.

By (4) and Lemma 4.1(4),

$$c_1(w,Q)r_{a_w} = F_1(w,Q)\begin{pmatrix}1\\0\end{pmatrix} = P_{w,e}\begin{pmatrix}\eta & *\\ 0 & \eta^{-1}\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \eta P_{w,e}\begin{pmatrix}1\\0\end{pmatrix},$$
$$c_1(v,Q)r_{a_v} = F_1(v,Q)\begin{pmatrix}0\\1\end{pmatrix} = P_{v,e}\begin{pmatrix}\eta & 0\\ * & \eta^{-1}\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \eta^{-1}P_{v,e}\begin{pmatrix}0\\1\end{pmatrix}.$$

Rearranging gives

$$P_{w,e}\begin{pmatrix}1\\0\end{pmatrix} = \eta^{-1}c_1(w,Q)r_{a_w}, \qquad P_{v,e}\begin{pmatrix}0\\1\end{pmatrix} = \eta c_1(v,Q)r_{a_v}.$$

Thus the first column of $P_{w,e}$ is a scalar multiple of r_{a_w}, and the second column of $P_{v,e}$ is a scalar multiple of r_{a_v}. Denote these scalars by p_w and p_v. They are nonzero because the matrices are invertible. Moreover, the matrices $P_{w,e}$, $P_{v,e}$ and the chosen representatives r_{a_w}, r_{a_v} depend only on the edge and its endpoints, so p_w, p_v are independent of Q even though η and $c_1(w,Q), c_1(v,Q)$ may vary with Q. Consequently,

$$c_1(w,Q) = p_w\eta, \qquad c_1(v,Q) = p_v\eta^{-1}.$$

Since $s_1 - 1 \mid (2^h+1)d_1$, we have $\eta^{(2^h+1)d_1} = 1$ by Fermat's little theorem, and hence,

$$\frac{c_1(w,Q)^{d_1}}{c_1(v,Q)^{2^h d_1}} = \frac{p_w^{d_1}}{p_v^{2^h d_1}}\eta^{(2^h+1)d_1} = \frac{p_w^{d_1}}{p_v^{2^h d_1}} =: \alpha_e.$$

□

Proof of Claim 4.7. The idea of this claim is to use the scaled permutation symmetry of projective Reed-Solomon codes from Claim 3.5.

By Lemma 4.1(4),

$$F_2(v,Q) = B_e F_2(w,Q).$$

Both vertices v, w have second type bit t_2. Comparing column t_2+1 on both sides and using (4) gives

$$c_2(v,Q)r_{\beta_e(b)} = c_2(w,Q)B_e r_b.$$

Rearranging,

$$\frac{c_2(w,Q)}{c_2(v,Q)}r_b = B_e^{-1}r_{\beta_e(b)}.$$

Because $(x+y)^{2^h} = x^{2^h} + y^{2^h}$ for any field elements x, y in characteristic 2, we can take entrywise 2^{ht_2}-th powers both sides and get

$$\left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{2^{ht_2}} r_{b^{2^{ht_2}}} = (B_e^{-1})^{2^{ht_2}} r_{\beta_e(b)^{2^{ht_2}}}.$$

Hence,

$$\left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{2^{ht_2}d_2} f(r_{b^{2^{ht_2}}}) = f\left(\left(\frac{c_2(w,Q)}{c_2(v,Q)}\right)^{2^{ht_2}} r_{b^{2^{ht_2}}}\right) = f\left((B_e^{-1})^{(2^{ht_2})} r_{\beta_e(b)^{2^{ht_2}}}\right).$$

Letting $f'(r_x) := f((B_e^{-1})^{(2^{ht_2})} r_x)$ proves the claim. □

□

4.4 Generalization to Higher Dimensions

We now extend the 2-dimensional construction in the previous subsection to the 4-dimensional complex of Lemma 4.1 by placing local codes on 3-cubes. This essentially a straightforward extension.

Definition 4.8 (Four-dimensional local codes)**.** *We assign a local code to each cube (3-face) in the 4-dimensional complex as follows:*

1. ***Base codes.*** *Let $s_i = 2^{m_i}$ with $m_i \geqslant 1$, let $\mathbb{E}$ be a finite field containing every $\mathbb{F}_{s_i}$, and fix an integer $h \geqslant 0$. Choose*

$$0 \leqslant d_i \leqslant s_i - 1, \qquad s_i - 1 \mid (2^h + 1)d_i, \qquad i \in [4],$$

and set $A_i = \mathbb{P}^1(\mathbb{F}_{s_i})$ and $C_i = \mathrm{PRS}_{d_i}(\mathbb{F}_{s_i}; \mathbb{E}) \subseteq \mathbb{E}^{A_i}$. As before, P permutes the evaluation labels by $x \mapsto x^{2^h}$ while fixing ∞.

2. ***Vertex weights.*** *For a vertex v of type $t_1t_2t_3t_4$, write $Q = Q_v(a_1, a_2, a_3, a_4)$ for the 4-cube with these edge labels at v, and define $c_i(v, Q) \in \mathbb{F}_{s_i}^{\times}$ and $\nu_v(Q)$ by*

$$\text{column } t_i + 1 \text{ of } F_i(v, Q) = c_i(v, Q) r_{a_i} \qquad (i \in [4]),$$

$$\nu_v(Q) = \prod_{i=1}^{4} c_i(v, Q)^{2^{ht_i} d_i}.$$

3. ***Local codes.*** *Assign a symbol $z(Q) \in \mathbb{E}$ to every 4-face (hypercube) Q. For a 3-face (cube) τ missing direction i, its* designated vertex *is the unique vertex $w \in \tau$ whose type bits in directions $j \neq i$ are zero. Write $Q_\tau(b)$ for the 4-face obtained by adjoining the direction-i edge labeled $b \in A_i$ at w. In this coordinate order, let D_τ multiply the coordinate for $Q_\tau(b)$ by $\nu_w(Q_\tau(b))$. The local code is $C_\tau = D_\tau P^{t_i} C_i$, where t_i is the constant ith type bit on τ, as listed in Table 2.*

Type bit t_i on τ	Required code C_τ
0	$D_\tau C_i$
1	$D_\tau P C_i$

Table 2: Local codes on 3-faces missing direction i, for every $i \in [4]$.

Thus the constraint at τ requires a homogeneous degree-d_i polynomial f such that

$$z(Q_\tau(b)) = \nu_w(Q_\tau(b)) f(r_{b^{2^{ht_i}}}) \qquad (b \in A_i).$$

The global code consists of assignments satisfying these constraints on every 3-face.

The next lemma extends Lemma 4.5 to the local view of any face.

Lemma 4.9 (Local Views are Tensor Codes)**.** *Let f be any face with direction set $S \subseteq [4]$, and fix a vertex $v \in f$ of type $t_1t_2t_3t_4$. An assignment z to the 4-faces containing f satisfies the constraints C_τ for every 3-face τ containing f if and only if there exists $M \in \bigotimes_{i \notin S} C_i$ such that*

$$z(Q) = \nu_v(Q)\, M\big((a_i^{2^{ht_i}})_{i \notin S}\big), \qquad Q = Q_v(a_1, a_2, a_3, a_4) \supseteq f.$$

Proof. Fix a 3-face τ missing direction i. We first show that its constraint, divided coordinatewise by $\nu_v(Q)$ and expressed in the labels at any $v \in \tau$, is exactly $P^{t_i}C_i$. Let w be the designated vertex of τ. At w, dividing the constraint coordinatewise by $\nu_w(Q)$ gives $P^{t_i}C_i$ by definition.

For any other vertex $v \in \tau$, choose a path from w to v along edges of τ. We show that the desired description is equivalent at the two endpoints of each edge, so it holds at v. Consider on this path an edge $e : u \to u'$ in τ of direction $k \neq i$, oriented from type bit $t_k = 0$ to $t_k = 1$. Claim 4.6, with direction 1 replaced by k, shows that the direction-k factor in $\nu_u(Q)/\nu_{u'}(Q)$ is independent of Q. For $j \notin \{i, k\}$, the direction-j edge labels at u, u' are fixed by τ. Comparing the corresponding columns in Lemma 4.1(4) shows that $c_j(u, Q)/c_j(u', Q)$ is also independent of Q. The only remaining factor is in direction i, where the proof of Claim 4.7 applies with matrix $B_e^{(i)}$. As in Claim 4.7, the invertible linear substitution and the nonzero constant factors give an equivalence between the rescaled codeword symbols at u and u'. Applying this equivalence successively along the chosen path proves the assertion at v.

Now fix f and v as in the statement. The 4-faces containing f are indexed by the labels $(a_i)_{i \notin S}$ at v. Divide their symbols by $\nu_v(Q)$ and reindex each coordinate by $a_i \mapsto a_i^{2^{ht_i}}$, then by the preceding paragraph, the constraints on the 3-faces containing f become exactly the C_i constraints on every direction-i line for each $i \notin S$. This concludes the proof. □

5 Rate, Distance, and Soundness Analysis

In this section, we define the quantum code using local constraint systems on the cubes described in the previous section. We instantiate the parameters and then establish its rate, distance, and soundness by adapting the corresponding arguments of Dinur–Lin–Vidick [DLV24], under the product expansion conjecture.

5.1 Defining the Quantum Code

We define a cochain complex and quantum code using the cubical complex of Lemma 4.1 and the local codes from Definition 4.8.

Recall from Lemma 4.1 that the cubical complex X has vertex set

$$X(0) = (G/H) \times \{0, 1\}^4.$$

For $i \in [4]$, let B_i be the $(s_i + 1)$-regular graph on G/H with edges $gH \sim gdH$ for $dH \in D_i/H$. The direction-i edges of X follow B_i and flip the ith type bit, as in (2). Write $X(j)$ for the j-faces and $S(f) \subseteq [4]$ for the direction set of a face f.

For the local code system, we use the field $\mathbb{E}$, base codes C_i, and local codes C_τ as in Definition 4.8. Write $Q(f) = \{Q \in X(4) : f \subseteq Q\}$. For every face f, define the set of valid local views

$$F_f = \left\{ z \in \mathbb{E}^{Q(f)} : z|_{Q(\tau)} \in C_\tau \text{ for every } \tau \in X(3) \text{ containing } f \right\}.$$

Thus $F_\tau = C_\tau$ on 3-faces and $F_Q = \mathbb{E}$ on 4-faces. For faces $f \subseteq g$, restriction to $Q(g)$ maps F_f into F_g. The cochain spacces are

$$\mathcal{C}^j = \bigoplus_{f \in X(j)} F_f.$$

The coboundary maps $\delta^j : \mathcal{C}^j \to \mathcal{C}^{j+1}$, for $0 \leqslant j < 4$, are defined by

$$(\delta^j x)(g) = \sum_{\substack{f \subseteq g \\ \dim f = j}} x(f)|_{Q(g)}, \qquad \text{for } x \in \mathcal{C}^j,\ g \in X(j+1)$$

If $f \subseteq g$ and $\dim g = \dim f + 2$, there are exactly two intermediate faces. The two compositions both restrict to $Q(g)$, so they cancel in characteristic two. Thus $\delta^{j+1}\delta^j = 0$, giving the cochain complex

$$\mathcal{C}^\bullet = \mathcal{C}^0 \xrightarrow{\delta^0} \mathcal{C}^1 \xrightarrow{\delta^1} \mathcal{C}^2 \xrightarrow{\delta^2} \mathcal{C}^3 \xrightarrow{\delta^3} \mathcal{C}^4.$$

By Lemma 4.9, at each vertex v, under the weights $\nu_v(Q)$ and label permutations $a_i \mapsto a_i^{2^{ht_i}}$ we can identify

$$F_f \cong \bigotimes_{i \notin S(f)} C_i \qquad (f \ni v).$$

Choose an $\mathbb{E}$-basis for each F_f. In these bases, we define check matrices

$$H_Z = \delta^2, \qquad H_X = (\delta^1)^{\mathsf{T}}$$

and place the physical qudits on the coordinates of $\mathcal{C}^2$ in the chosen bases.

Since

$$H_X H_Z^{\mathsf{T}} = (\delta^2\delta^1)^{\mathsf{T}} = 0,$$

these matrices define a CSS code over $\mathbb{E}$, encoding K qudits into N qudits of alphabet $\mathbb{E}$, where

$$N = \dim_{\mathbb{E}} \mathcal{C}^2, \qquad K = \dim_{\mathbb{E}}(\ker \delta^2 / \operatorname{im} \delta^1).$$

The following claim is immediate.

Claim 5.1. *The row and column weights of H_Z, H_X are bounded whenever $s_i,\ i \in [4]$ are bounded.*

5.2 Rate

We bound the quantum code rate using the argument of [DLV24, Section 4].

Lemma 5.2 (Rate)**.** *Write $R_i = \dim_{\mathbb{E}} C_i/(s_i+1)$. If $R_1, R_2 < \mu$ and $R_3, R_4 > 1-\mu$, then $\frac{K}{N} \geqslant \frac{2-(1+2\mu)^4}{24}$.*

Proof. It is straightforward to extend the proof of [DLV24, Theorem 4.1] to our setting. They assumed the same local codes with the same block length are used throughout their cubical complex. However, their proof only relies on the local views being tensor codes, which our construction satisfies. Adapting the dimension count in [DLV24, Lemma 4.2 and the proof of Corollary 3.7] gives $K \geqslant |X(4)| \left(2 - (1+2\mu)^4\right)$. On the other hand, $N = 4|X(4)| \sum_{1 \leqslant i < j \leqslant 4} R_i R_j \leqslant 24|X(4)|$. Combining proves the claim. □

5.3 Distance and Soundness

We next apply [DLV24]'s expansion analysis to the cubical complex defined in Section 5.1. They use the expansion of the base graphs of the complex along with the product expansion of the local tensor codes and their duals to conclude the (co)systolic distance and (co)cycle expansion of the complex via a local-to-global argument, which then translates to the distance and local testability of the associated quantum code.

In Section 6, we establish the following:

Lemma 5.3 (Spectral expansion)**.** *For our cubical complex X from Lemma 4.1, each B_i for $i \in [4]$ is connected and X is λ-expanding for $\lambda = \max_{i\in[4]} \frac{2\sqrt{s_i}}{s_i+1}$.*

Lemma 5.4 (Distance and soundness)**.** *Fix the field $\mathbb{E}$ and the parameters $s_1, \dots, s_4$ bounded independently of N. For every $\rho > 0$ and $\beta \geqslant 1$, there is $\lambda = \lambda(\rho, \beta) > 0$ with the following property. Suppose $\max_i(s_i + 1)/\min_i(s_i + 1) \leqslant \beta$, every nonempty subtuple of (C_i) and of $(C_i^\perp)$ is ρ-product-expanding, and X has spectral expansion λ. Then the resulting qudit CSS code has distance $D = \Omega(N)$ and soundness $\kappa > 0$.*

Proof. Again, by Lemma 4.9, the local views and incidence maps of our cochain complex have tensor code structures. This suffices to extend the proof of in [DLV24, Corollary 3.7], giving linear distance and constant soundness in the coboundary direction, provided λ is sufficiently small in terms of ρ, β ([DLV24] assume the directions have equal degrees, but unequal degrees only affect the constants only through β). To prove the boundary direction's distance and soundness, [DLV24, Propostion 8.1] performs an argument on a 'dual' cochain complex where the local codes at cubes (C_τ) are replaced by their duals $(C_\tau^\perp)$. Thus, we only need to verify that the local views in this dual cochain complex are tensor codes. Indeed, $\mathrm{PRS}_{d_i}(\mathbb{F}_{s_i}; \mathbb{E})^\perp = \mathrm{PRS}_{d_i'}(\mathbb{F}_{s_i}; \mathbb{E})$ with $d_i' = s_i - 1 - d_i$, and the dual codes $(\mathrm{PRS}_{d_i'}(\mathbb{F}_{s_i}; \mathbb{E}))$ satisfy the requirements in Definition 4.8. Thus Lemma 4.9 applies. □

5.4 Product Expansion of Reed-Solomon Codes

In this section, we state our conjecture on the product expansion of Reed-Solomon codes. For a finite set $S \subseteq \mathbb{E}$ and $0 \leqslant d < |S|$, write

$$\mathrm{RS}_d(S; \mathbb{E}) = \{(f(x))_{x\in S} : f \in \mathbb{E}[T],\ \deg f \leqslant d\}.$$

Thus the subscript denotes the maximum polynomial degree, as in the PRS notation.

Fix $k \geqslant 2$, finite sets $S_1, \dots, S_k$, and write

$$\Omega = \prod_{i=1}^{k} S_i, \qquad N = |\Omega| = \prod_{i=1}^{k} |S_i|.$$

An i-axis line is obtained by fixing every coordinate except the ith. For a code $C_i \subseteq \mathbb{F}_q^{S_i}$, let

$$L_i(C_i) = \mathbb{F}_q^{S_1} \otimes \cdots \otimes \mathbb{F}_q^{S_{i-1}} \otimes C_i \otimes \mathbb{F}_q^{S_{i+1}} \otimes \cdots \otimes \mathbb{F}_q^{S_k}.$$

For $M_i \in L_i(C_i)$, let $\ell_i(M_i)$ be the number of nonzero i-axis lines.

Definition 5.5 (Product expansion)**.** *A tuple $(C_i)_{i\in I}$, where $I \subseteq [k]$, is ρ-product-expanding if every $M \in \sum_{i\in I} L_i(C_i)$ has a decomposition $M = \sum_{i\in I} M_i$, with $M_i \in L_i(C_i)$, such that*

$$\mathrm{wt}(M) \geqslant \rho \sum_{i\in I} |S_i| \ell_i(M_i).$$

We now state our conjecture for product expansion of Reed-Solomon codes over binary extension fields,

Conjecture 5.6 (Product expansion over Binary Extension Fields)**.** *For every $r \geqslant 2$, $\eta \in (0,1)$, and $\beta \geqslant 1$, there exists $\rho = \rho(r,\eta,\beta) > 0$ with the following property. Let $\mathbb{E} = \mathbb{F}_{2^e}$, and let $m_1, \ldots, m_r$ be positive integers such that $2m_i \mid e$ for every i. Set $s_i = 2^{m_i}$ and define*[5]

$$S_i \coloneqq \{x \in \mathbb{E}^\times : x^{s_i+1} = 1\}, \qquad |S_i| = s_i + 1.$$

Suppose that $s_1 + 1, \ldots, s_r + 1$ are pairwise coprime and satisfy $\max_{i\in[r]}(s_i+1) \leqslant \beta \min_{i\in[r]}(s_i+1)$. Then, for every choice of integers $0 \leqslant d_i$ with $d_i + 1 \leqslant (1-\eta)(s_i+1)$, the tuple $\left(\mathrm{RS}_{d_i}(S_i;\mathbb{E})\right)_{i=1}^r$ is ρ-product-expanding.

Lemma 5.7 (Product expansion of projective Reed–Solomon codes)**.** *Let $\mathbb{E}, s_i, S_i$ be as in Conjecture 5.6, and let $C_i = \mathrm{PRS}_{d_i}(\mathbb{F}_{s_i};\mathbb{E})$, where $0 \leqslant d_i \leqslant s_i - 1$. Assuming Conjecture 5.6, if $\eta \leqslant \frac{d_i+1}{s_i+1} \leqslant 1-\eta$ for every i, then all nonempty subtuples of (C_i) and of $(C_i^\perp)$ are ρ-product-expanding for a common $\rho > 0$ depending only on r, η, β.*

Proof. We will show that C_i and $C_i^\perp$ are equivalent, by coordinate permutations and nonzero coordinate scalings, to $\mathrm{RS}_{d_i}(S_i;\mathbb{E})$ and $\mathrm{RS}_{s_i-1-d_i}(S_i;\mathbb{E})$ respectively. Provided that $\eta \leqslant \frac{d_i+1}{s_i+1} \leqslant 1-\eta$, the latter Reed-Solomon code tuples have product expansion from the conjecture. The equivalence then transfers this product expansion to that of the projective RS code tuples because product expansion only concerns Hamming weights.

We now show the equivalence between C_i and $\mathrm{RS}_{d_i}(S_i;\mathbb{E})$. Fix i and write $s = s_i$, $d = d_i$, and $S = S_i$. First, we identify the evaluation sets of PRS and RS codes. Choose $\theta \in \mathbb{F}_{s^2} \setminus \mathbb{F}_s$ and define

$$\beta(x) = \frac{x-\theta}{x-\theta^s}, \qquad \beta(\infty) = 1.$$

For $x \in \mathbb{F}_s$, we have $\beta(x)^s = \frac{x-\theta^s}{x-\theta}$ (this follows from simple algebra, using s is a power of characteristic and $x^s = x$, $\theta^{s^2} = \theta$), so $\beta(x)^{s+1} = 1 \implies \beta(x) \in S$. Since $\theta \neq \theta^s$, this is an invertible Möbius transformation. It therefore maps $\mathbb{P}^1(\mathbb{F}_s)$ bijectively onto S, both sets having size $s+1$.

Next, we change polynomial variables using the linear transformation

$$B = \begin{pmatrix} 1 & -\theta \\ 1 & -\theta^s \end{pmatrix}.$$

For a homogeneous degree-d polynomial f, set $F = f \circ B^{-1}$ and $g(T) = F(T,1)$. The matrix B is invertible, thus this gives a bijection between homogeneous degree-d polynomials f over $\mathbb{E}$ and univariate polynomials g of degree at most d.

[5] Since $2m_i \mid e$, we have $s_i + 1 \mid s_i^2 - 1 \mid 2^e - 1$. Thus $|S_i| = s_i + 1$ because $\mathbb{E}^\times$ is cyclic.

Finally, let $\lambda_x = x - \theta^s$ for finite x and $\lambda_\infty = 1$. These scalars are nonzero, and $Br_x = \lambda_x r_{\beta(x)}$. Hence

$$f(r_x) = F(Br_x) = \lambda_x^d F(r_{\beta(x)}) = \lambda_x^d g(\beta(x)).$$

Thus dividing coordinate x by λ_x^d and relabeling it by $\beta(x)$ transforms the PRS codeword into $(g(z))_{z\in S}$. Since every polynomial g of degree at most d occurs, the resulting code is exactly $\mathrm{RS}_d(S;\mathbb{E})$.

The equivalence of $C_i^\perp$ and $\mathrm{RS}_{s_i-1-d_i}(S_i;\mathbb{E})$ follows exactly the same argument by using Claim 3.4. □

5.5 Proof of Main Theorem

We prove the main theorem in this section by combining the results in preceding sections and choosing parameters.

We first choose the four base local codes satisfying the conditions required in the preceding lemmas.

Definition 5.8 (Parameters for local codes)**.** *Let L be a sufficiently large positive multiple of 128, and set*

$$(m_1, m_2, m_3, m_4) = (L+8, L+16, L+32, L+64), \qquad s_i = 2^{m_i}.$$

Over the common field $\mathbb{E} = \mathbb{F}_{2^{2\,\mathrm{lcm}(m_1,m_2,m_3,m_4)}}$, take $C_i = \mathrm{PRS}_{d_i}(\mathbb{F}_{s_i};\mathbb{E})$ where

$$d_i = \begin{cases} (s_i-1)/17, & i = 1,2, \\ 16(s_i-1)/17, & i = 3,4. \end{cases}$$

These base codes satisfy Definition 4.8 with $h = 4$.

We explain the choices. Since $8 \mid m_i$, we have $17 \mid s_i - 1$, and thus the degrees are integral and satisfy $s_i - 1 \mid 17d_i$ as required for the divisibility condition. We pick $h = 4$ the smallest choice, so that the first two code rates are below $1/16$ and the last two are above $15/16$, giving the separation needed for the rate bound of Lemma 5.2 to be nontrivial.

All four rates lie in $[1/17, 16/17]$, so both the codes and their duals have rates bounded away from one. The largest powers of two dividing m_1, m_2, m_3, m_4 are $8, 16, 32, 64$, respectively. Moreover, $\frac{\max_i(s_i+1)}{\min_i(s_i+1)} < 2^{56}$. For Conjecture 5.6 and Lemma 5.7 to apply, we finally need $s_1+1, \ldots, s_r+1$ to be pairwise coprime. Indeed, consider a pair $1 \leqslant i < j \leqslant 4$ and the equality $s_i^{m_j} = s_j^{m_i}$. Taking 2^{i+2}-root both sides give $s_i^{L/2^{i+2}+2^{j-i}} = s_j^{L/2^{i+2}+1}$. Because $128 \mid L$, the left exponent is even while the right exponent is odd. Now suppose for contradiction there existed an odd prime p such that p divides both $s_i + 1$ and $s_j + 1$, then $s_i = -1 \mod p$ and $s_j = -1 \mod p$. Then $s_i^{L/2^{i+2}+2^{j-i}} = s_j^{L/2^{i+2}+1}$ implied $1 = -1 \mod p$, a contradiction.

Note that increasing L makes the spectral bounds Lemma 5.3 tend to zero while keeping the rate bounds and the degree ratio fixed. After choosing L, we keep it and $\mathbb{E}$ fixed as the global cubical complex grows.

With the above parameter choices, we can establish the main theorem.

Theorem 5.9 (Main Theorem)**.** *Assuming Conjecture 5.6, there exists an infinite family of quantum CSS codes on N qubits with $K = \Omega(N)$, distance $D = \Omega(N)$, constant soundness, and constant check weights.*

Proof. A qudit CSS code over alphabet $\mathbb{E}$ with the stated properties directly follows from our construction. In particular, Lemma 5.7, with $\eta = 1/17$ and $\beta = 2^{56}$, gives a product-expansion constant $\rho > 0$ independent of L for all nonempty subtuples of the local codes and their duals. By Lemma 5.3, choosing L sufficiently large makes every $2\sqrt{s_i}/(s_i+1)$ smaller than the threshold in Lemma 5.4. Fix such an L and use the arbitrarily large complexes of Lemma 4.1 together with the local codes above to construct the cochain complex as described in Section 5.1. Then Lemma 4.9 and Claim 5.1 establishes CSS construction with bounded check weights and degrees. Lemma 5.2, with $\mu = 1/16$, gives $K = \Omega(N)$, and Lemma 5.4 gives $D = \Omega(N)$ and constant soundness.

To obtain a qubit code, we expand the cochain maps δ^2, δ^1 in an $\mathbb{F}_2$-basis of $\mathbb{E}$ and and define qubit check matrices using the expanded maps. This gives $N \log_2 |\mathbb{E}|$ physical and $K \log_2 |\mathbb{E}|$ logical qubits. The distance is at least D and soundness decreases by at most a factor $O(\log_2 |\mathbb{E}|)$. Each row and column weight of the check matrices increases by at most a factor $\log_2 |\mathbb{E}|$. Since $\mathbb{E}$ is fixed, all claimed parameters follow. □

6 The Non-Abelian Cubical Complex

The construction of the complexes below are due to Rungtanapirom, Stix, and Vdovina [RSV19, §2.1-2.4; §6.2]. We begin by describing the infinite trees underlying the cubical complex X and how to assign invertible 2×2 matrices to the edges incident to each vertex, which enables us to exhibit the matrices arising in property (4) of Lemma 4.1.

First, we set up some notation. We choose positive integers $m_1, \ldots, m_4$ divisible by 8 and let $s_i = 2^{m_i}$. The field $K_i = \mathbb{F}_{s_i}((\pi_i))$ consists of formal series $\sum_{n \geqslant N} a_n \pi_i^n$, where $N \in \mathbb{Z}$ and $a_n \in \mathbb{F}_{s_i}$. The ring $\mathcal{O}_i = \mathbb{F}_{s_i}[[\pi_i]]$ consists of series with only terms with nonnegative exponents. For a nonzero series a, its valuation $\mathrm{val}_i(a)$ is the smallest exponent with nonzero coefficient and reduction modulo π_i keeps only the constant coefficient of a series in $\mathcal{O}_i$; we denote this operation by $\overline{a} := a \pmod{\pi_i}$.

Bruhat-Tits tree and corresponding projective line edge labels. An $\mathcal{O}_i$*-lattice* is defined by $L = \mathcal{O}_i p \oplus \mathcal{O}_i q$, where $p, q \in K_i^2$ are linearly independent over K_i. We call (p, q) an *ordered basis* of L and write $L = U\mathcal{O}_i^2$ for the matrix $U = (p\ q)$. We identify lattices that differ by multiplication by a nonzero scalar in K_i, and write $[L]$ for the corresponding equivalence class.

The vertices of $\mathcal{T}_i$ are these classes. Two vertices are adjacent if they admit representatives L, L' satisfying

$$\pi_i L \subsetneq L' \subsetneq L. \tag{7}$$

This graph is the *Bruhat-Tits tree* [Ser80, Chapter II]. We define $\mathcal{T} = \prod_{i=1}^4 \mathcal{T}_i$, where here the $\prod$ denotes the Cartesian product of graphs.

For a fixed L, every neighboring class has a unique representative L' satisfying (7). The map $L' \mapsto L'/\pi_i L$ identifies these neighbors with the one-dimensional subspaces of $L/\pi_i L$. An ordered basis of L in turn identifies $L/\pi_i L$ with $\mathbb{F}_{s_i}^2$ by reducing the two coordinates modulo π_i, so we can think of the neighbors of L as corresponding to one-dimensional subspaces of this copy of $\mathbb{F}_{s_i}^2$. The set of one-dimensional subspaces of $\mathbb{F}_{s_i}^2$ is called the *projective line*, denoted $\mathbb{P}^1(\mathbb{F}_{s_i})$. It has $s_i + 1$ elements, and we can think of each

of these elements as indexing a neighbor of $[L]$. Henceforth when we refer to an edge label we mean the corresponding projective line.

A vertex $v \in \mathcal{T}$ is a tuple $([L_1], [L_2], [L_3], [L_4])$. We describe it in coordinates by choosing a representative lattice and an ordered basis in each factor. A direction-i edge changes only the ith coordinate to a neighboring lattice class. For this edge, we can consider its label as a change of basis matrices that correspond to a choice of basis $V = (p\ q)$ and $V\,\mathrm{diag}(1, \pi_i)$ for its respective endpoints. These basis may differ from the bases chosen independently at the vertices; we will soon introduce *comparison matrices* that describe how to relate these two choices.

Vertex types and group actions. For $L = U\mathcal{O}_i^2$ for some $U \in \mathrm{GL}_2(K_i)$, we define the *type* of $[L]$ to be $\mathrm{val}_i(\det U) \bmod 2$. A change of basis for the lattice multiplies $\det U$ by an element of valuation zero, while rescaling L changes its type by an even number. That is, the type is well-defined and independent of the choice of representative of L. For adjacent vertices, we can associate to the lattices at each of them the bases (p, q) and $(p, \pi_i q)$. Their determinants differ by a factor of π_i, so their determinant valuations differ by one. In particular, it follows that every edge joins vertices of opposite types.

We write $\mathrm{GL}_2(K_i)$ for the invertible 2×2 matrices over K_i. The group $\mathrm{PGL}_2(K_i)$ identifies matrices that differ by a nonzero scalar and acts on tree vertices by $[L] \mapsto [AL]$.

Choosing coordinate bases for the lattices at endpoints of an edge. We fix direction i and consider matrices $U, V \in \mathrm{GL}_2(K_i)$ whose corresponding lattices represent the same tree vertex. That is, there exists $a \in K_i^\times$ such that $aV\mathcal{O}_i^2 = U\mathcal{O}_i^2$. Then the matrix $M = U^{-1}aV$ and its inverse have entries in $\mathcal{O}_i$, since they compare two bases of the same lattice. We denote this group of matrices by $\mathrm{GL}_2(\mathcal{O}_i)$.

The reduction $\overline{M} \in \mathrm{GL}_2(\mathbb{F}_{s_i})$ expresses the edge labels obtained from V in the coordinates obtained from U. A convenient choice of normalization is to record this action using a determinant-one matrix: every nonzero element of $\mathbb{F}_{s_i}$ has a unique square root, so $\nu_i(R) = (\det R)^{-1/2} R$ is the unique determinant-one scalar multiple of R which acts on one-dimensional subspaces in the same way as R. To that end, we define the *comparison matrices*

$$\mathsf{C}_i(U, V) = \nu_i\left(\overline{U^{-1}aV}\right) \in \mathrm{SL}_2(\mathbb{F}_{s_i}). \tag{8}$$

To check that $\mathsf{C}_i(U, V)$ is well-defined, we note that any two admissible choices of a differ by an invertible element of $\mathcal{O}_i$. Their reduction $(\bmod \pi)_i$ differ by a nonzero scalar, which is then removed after the appropriate normalization. It follows that $\mathsf{C}_i(U, V)$ is independent of a. Furthermore, rescaling either basis matrix by an element of $K_i^\times$ also leaves this comparison unchanged.

Composition of change of basis and lifting from X into $\mathcal{T}$. For basis matrices U, V, W representing the same tree vertex, and $A \in \mathrm{GL}_2(K_i)$, by thinking of the $\mathsf{C}_i(U, V)$ as change of basis, we have the following composition laws of these comparison matrices

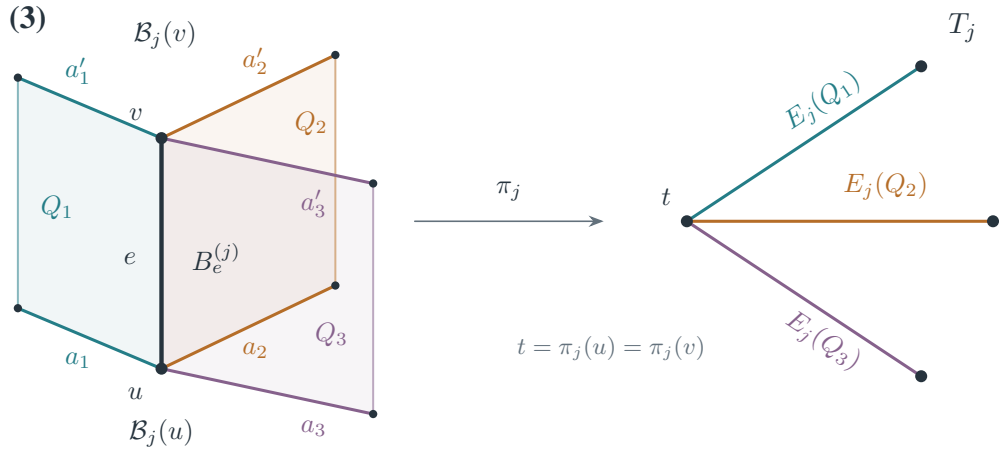
$$\begin{aligned} \mathsf{C}_i(U, V)\mathsf{C}_i(V, W) &= \mathsf{C}_i(U, W), \\ \mathsf{C}_i(AU, AV) &= \mathsf{C}_i(U, V). \end{aligned} \tag{9}$$

For $X = \Gamma\backslash\mathcal{T}$, a *lift* of a vertex, edge, or cube is a corresponding vertex, edge, or cube in $\mathcal{T}$ mapping to it. The construction ensures that $\Gamma \leqslant \mathcal{G}$ acts freely on vertices: only the identity fixes a vertex. Together

with type preservation, this makes the element of Γ relating two lifts of a face unique. We can transport the chosen lattice bases to the entire under this action. Since changing a common lift then multiplies both basis matrices by the same matrix in each factor, it follows from the second identity in (9) that we may compute the change of basis for faces of X on any suitable lift in $\mathcal{T}$.

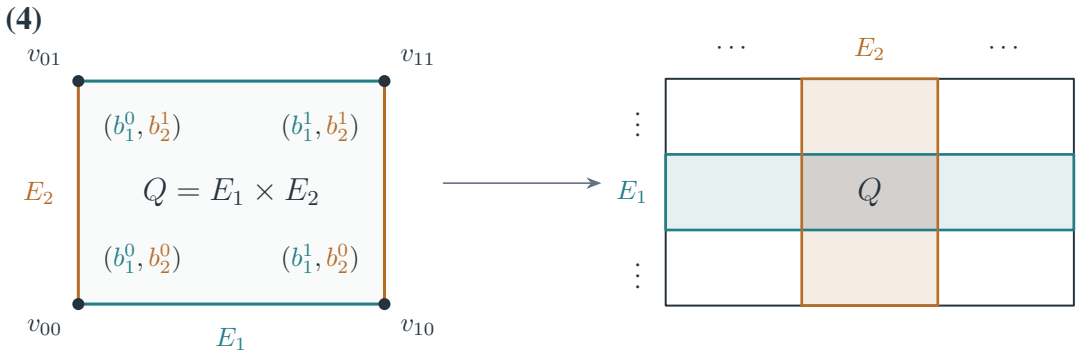
6.1 Proof of Lemma 4.1

Before proving Lemma 4.1, we give an informal sketch of its four properties. Properties (1) and (2) are essentially standard features of the construction. For (3) and (4), we lift to the product of trees, where the relevant matrices describe changes of lattice basis.



In the lift on the RHS, each colored square Q_r contains the fixed edge $e = [u, v]$. Its two transverse j-edges project to the same tree edge $E_j(Q_r)$; a_r and a'_r are their labels in the bases at u and v. Since these bases are fixed, one matrix $B_e^{(j)}$ relates the labels across all the squares.

For (3), we fix an edge e and consider the squares containing it in another direction. The opposite edges of each square project to the same tree edge, whose labels we read using the bases at the two endpoints of e. Consequently, the same change-of-basis matrix works for the entire book of squares around e because those vertex bases are fixed independently of the square. This is essentially the content of the $B_e^{(j)}$ matrices.



The corner pairs (b_1^a, b_2^b) are induced by "change of basis" matrices on E_1 and E_2. The square $Q = E_1 \times E_2$ corresponds to the common entry of the teal row (E_1 fixed) and orange column (E_2 fixed). Both checks use the same basis at this entry.

For (4), we lift a square into $\mathcal{T}$ as $\widetilde{Q} = E_1 \times E_2$. For each E_i, we choose corresponding matrices that allows us to read off lattices corresponding to its endpoints. In the coordinates given by V_i, these lattices have the standard forms $\mathcal{O}_i^2$ and $\mathrm{diag}(1, \pi_i)\mathcal{O}_i^2$. At each vertex v of the Q, the matrix $F_i(v, Q)$ essentially records the comparison between the (relative) choice of basis for v as read off from the matrices assigned for E_i and the choice of basis for the lattice representing v_i. The composition identities give the compatibility relations for both incident edges to v using these same matrices (which is why we can construct the same weight $\nu_v(Q)$ for both columns and rows in Section 4.3).

Proof of Lemma 4.1. (1) Quotient structure of X. This follows essentially from the machinery already developed in [RSV19]. The construction in [RSV19, § 2.1-2.4] supplies a group Λ acting transitively on the vertices of the product of trees $\mathcal{T}$. The stabilizer H_0 of a base vertex (which can be thought of as the origin) is isomorphic to S_3, by [RSV19, Proposition 2.1 and the proof of Proposition 2.21(1)].

It is possible to embed H_0 as a subgroup conjugate to $H = \mathrm{PGL}_2(\mathbb{F}_2)$, and a standard argument allow us to identify X with $(G/H) \times \{0,1\}^4$. In slightly more detail, let Λ act on $\mathcal{T}$ as in [RSV19], where Λ is defined using an appropriate quarternion algebra, and let $\theta : \Lambda \to G$ be the group homomorphism corresponding to reduction at a irreducible polynomial of $\mathbb{F}_2[t]$ of sufficiently large degree [RSV19, Remark 6.10]. We also define $\tau : \Lambda \to \{0,1\}^4$ which records how the action by Λ changes the vertex types of the tree, and let $\Gamma = \ker\theta \cap \ker\tau$. The ideas in [RSV19, Remark 6.10] show that $\theta(\ker\tau) = G$. Since Λ is vertex transitive, this shows in turn that (θ, τ) is onto. That is, each coset G/H of every type vector occurs. Moreover, the aforementioned reduction map can be shown to embed H_0 as a subgroup conjugate to $H = \mathrm{PGL}_2(\mathbb{F}_2)$, and $\tau(H_0) = 0$. Consequently,

$$V(X) = \Gamma\backslash\Lambda/H_0 \cong (G \times \{0,1\}^4)/(H \times \{0\}) \cong (G/H) \times \{0,1\}^4.$$

We may choose the irreducible polynomial of sufficiently large degree to not be amongst the finitely many nonidentity elements so that $\theta(\lambda) \neq 1$ for every nonidentity $\lambda \in \Lambda$ sending the base vertex o to λo of distance at most 8 from it. By using the definition of Γ as the intersection of kernels and the vertex transitivity of Λ,

we can show that no two vertices at distance at most 8 are merged in the quotient, which ensures that the cubical structure around every vertex in the quotient X is the same as its corresponding vertex in the infinite graph $\mathcal{T}$.

The relations $D_iD_j = D_jD_i$ for $i \neq j$ come from the product structure of $\mathcal{T}$: two successive steps in distinct directions can be taken in either order to reach the same vertex.

(2) Projective line labels for edges. In each direction the local construction is the same Bruhat-Tits description that underlies the LPS graphs [LPS88]: links are the lines in its two-dimensional residue space, showing the identification of the incident edges with $\mathbb{P}^1(\mathbb{F}_{s_i})$. (Alternatively, as we have already observed this also follows from [Ser80, Chapter II].)

Interlude: Lifting from the complex into the tree. For each vertex v of X, we choose one lift $\widetilde{v}$ and, in each factor i, a basis matrix $U_{\widetilde{v},i}$ for a lattice representing its ith coordinate. We extend these choices to the other lifts by the action of Γ. More precisely, if $A_i \in \mathrm{GL}_2(K_i)$ represents the ith component of $\gamma \in \Gamma$, we set $U_{\gamma\widetilde{v},i} = A_iU_{\widetilde{v},i}$. As observed in the preamble, we may carry out the computations on lifted vertices, edges, and cubes in $\mathcal{T}$.

(3) The matrix attached to a directed edge. Fix a direction-i edge $e : v \to w$ and lift it to $\widetilde{e} : \widetilde{v} \to \widetilde{w}$. For $j \neq i$ the endpoints have the same jth tree coordinate. Hence their factor-j basis matrices span homothetic lattices, and we define

$$B_e^{(j)} = \mathsf{C}_j(U_{\widetilde{w},j}, U_{\widetilde{v},j}). \tag{10}$$

This matrix converts the direction-j labels at v into the coordinates used at w. It depends only on the directed edge e and the fixed vertex coordinates.

In a lifted square containing e, the two opposite direction-j edges project to the same edge of $\mathcal{T}_j$. Their labels a at v and $\beta_e^{(j)}(a)$ at w describe this edge in the respective vertex coordinates. Since $B_e^{(j)}$ converts the coordinates at v to those at w, its definition in (10) gives $\beta_e^{(j)}(a) = \langle B_e^{(j)} r_a^{(j)} \rangle$. For the reverse edge $\bar{e}$, we exchange the two bases, giving $B_{\bar{e}}^{(j)} = (B_e^{(j)})^{-1}$.

(4) Change of coordinates for cubes.

We write $T_i = \mathrm{diag}(1, \pi_i)$. For a tree edge oriented from type zero to type one, we choose representatives $\pi_iL^0 \subsetneq L^1 \subsetneq L^0$. A basis $V = (p\ q)$ of L^0 is *adapted* to the edge if $L^1 = \mathcal{O}_ip \oplus \pi_i\mathcal{O}_iq$; its endpoint bases are V and VT_i. Such a basis exists by lifting a basis of L^0/π_iL^0 whose first vector spans L^1/π_iL^0.

For each cube Q, we choose one lift $\widetilde{Q}$ and adapted bases $V_{\widetilde{Q},i}$ for its projected tree edges. For $v \in Q$, we define, in a common lift,

$$F_i(v, Q) = \mathsf{C}_i\big(U_{\widetilde{v},i}, V_{\widetilde{Q},i}T_i^{\mathrm{type}_i(v)}\big). \tag{11}$$

The invariance in (9) makes this independent of the choice of lift. In the adapted endpoint bases, the projective lines corresponding to the edges are given by $\langle(1, 0)^{\mathsf{T}}\rangle$ and $\langle(0, 1)^{\mathsf{T}}\rangle$ respectively, since the type 0 vertex as viewed from type 1 is represented by π_iL^0. It follows that column $\mathrm{type}_i(v) + 1$ of $F_i(v, Q)$ spans the edge label.

For a direction-i edge $e : v \to w$ in Q and $j \neq i$, the cube uses the same factor-j endpoint basis at v and w. The composition identity gives

$$F_j(w, Q) = \mathsf{C}_j(U_{\widetilde{w},j}, U_{\widetilde{v},j})F_j(v, Q) = B_e^{(j)}F_j(v, Q). \tag{12}$$

We now orient $e : v \to w$ from type zero to type one in direction i, and choose a corresponding matrix $V_{\widetilde{e},i} = (p, q)$ (which we think of as assigning its two endpoints the choice of basis $L^0 = \mathcal{O}_ip \oplus \mathcal{O}_iq$ and

$L^1 = \mathcal{O}_i p \oplus \mathcal{O}_i(\pi_i q)$) independently of Q, and define

$$P_{v,e} = \mathsf{C}_i(U_{\widetilde{v},i}, V_{\widetilde{e},i}) \quad \text{and} \quad P_{w,e} = \mathsf{C}_i(U_{\widetilde{w},i}, V_{\widetilde{e},i} T_i).$$

For $Q \supset e$, we lift both into the same edge in $\mathcal{T}$ and rescale $V_{\widetilde{Q},i}$ (since the lattices are defined up to homothety) so that both type-zero lattices equal $L^0 = V_{\widetilde{e},i}\mathcal{O}_i^2$. Their type-one lattices also agree, by uniqueness of the neighbor representative between $\pi_i L^0$ and L^0. The matrix $M = V_{\widetilde{e},i}^{-1} V_{\widetilde{Q},i}$ lies in $\mathrm{GL}_2(\mathcal{O}_i)$ and preserves $T_i\mathcal{O}_i^2$. As $(1 \quad 0)^\top \in T_i\mathcal{O}_i^2$, it follows that $M(1 \quad 0)^T \in T_i\mathcal{O}_i^2$ so its lower-left entry is divisible by π_i. We obtain

$$M = \begin{pmatrix} a & b \\ \pi_i c & d \end{pmatrix}, \qquad T_i^{-1} M T_i = \begin{pmatrix} a & \pi_i b \\ c & d \end{pmatrix}, \qquad a, b, c, d \in \mathcal{O}_i.$$

Both reductions $\pmod{\pi_i}$ have determinant $\bar{a}\bar{d} \neq 0$. Normalizing the determinant to one, we get

$$\begin{aligned} P_{v,e}^{-1} F_i(v, Q) &\underset{(8)}{=} \nu_i(\overline{M}) = \begin{pmatrix} \eta & * \\ 0 & \eta^{-1} \end{pmatrix}, \\ P_{w,e}^{-1} F_i(w, Q) &\underset{(8)}{=} \nu_i\left(\overline{T_i^{-1} M T_i}\right) = \begin{pmatrix} \eta & 0 \\ * & \eta^{-1} \end{pmatrix}, \end{aligned} \tag{13}$$

where $\eta = \bar{a}/\sqrt{\bar{a}\bar{d}}$. The matrices $P_{v,e}$, $P_{w,e}$ and $B_e^{(j)}$ depend only on e and the fixed choices, while η and the starred entries may vary with Q. Equations (12) and (13) prove (3) on every edge, using the same matrices (11). This proves (4). □

6.2 Expansion of the cubical complex

Finally, we prove the expansion properties of X described in Lemma 5.3. RSV establish spectral expansion for their complexes in odd characteristic [RSV19, Theorem 6.12]. For completeness we provide the characteristic-two argument needed for our construction using Morgenstern's spectral bounds on quotients of trees [Mor94].

We write B_i for the $(s_i + 1)$-regular graph on G/H with edges $gH \sim gdH$ for $dH \in D_i/H$. We prove the bound by identifying B_i with a quotient of $\mathcal{T}_i$ and applying Morgenstern's theorem [Mor94] to a graph cover $p : \widetilde{B}_i \to B_i$. The covering map allows us to transfer the spectral bound to B_i.

Proof of Lemma 5.3. We first show that B_i is connected. We fix i and let $n_i = s_i + 1$. We recall the parts of the construction in [RSV19] needed below. The complex $X = \Gamma\backslash\mathcal{T}$ is a quotient of the Cartesian product $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_4$ of four infinite trees. We fix a base vertex $o = (o_1, o_2, o_3, o_4) \in \mathcal{T}$, with $o_j \in \mathcal{T}_j$ to be the origin. For each i, we write $\Lambda_i \leqslant \Lambda$ for the subgroup of elements whose action on $\mathcal{T}_j$ fixes o_j for every $j \neq i$. We write $\theta_i : \Lambda_i \to G$ for the finite quotient map from the construction, restricted to Λ_i. (This map reduces coefficients modulo an irreducible polynomial of degree r, which maps into the residue field $\mathbb{F}_{2^r}$.) The image of θ_i is a projective 2×2 matrix over $\mathbb{F}_{2^r}$. The group Λ_i acts transitively on $\mathcal{T}_i$, with stabilizer H_0 satisfying $\theta_i(H_0) = H$. The strong approximation argument in [RSV19, Remark 6.10] shows that θ_i remains surjective when restricted to elements preserving the bipartition of $\mathcal{T}_i$. With $\Delta_i = \ker \theta_i$, we note that $B_i = \Delta_i\backslash\mathcal{T}_i$, and so B_i is connected.

Morgenstern's result [Mor94, Theorem 3.2] shows that the degree-(s_i+1) tree quotients in his construction have normalized adjacency eigenvalues $|\mu| \leqslant 2\sqrt{s_i}/(s_i+1)$ whenever $\mu \neq \pm 1$. We will construct a cover of B_i in this family and transfer its spectral bound to B_i.

To be able to apply Morgenstern's result, we must impose one additional restriction on Δ_i. We recall that our quaternion coefficients belong to $\mathbb{F}_2(t)$, the field of rational functions in t. The restriction requires each group element to admit a representative x such that the coefficients of both x and x^{-1}, written in lowest terms, have denominators not divisible by t [Mor94, §3].

We use the auxiliary permutation action $\rho : \Lambda \to S_3$ from [RSV19, §2.6] to describe this restriction. We define $\Delta_i^0 = \{\delta \in \Delta_i : \rho(\delta) \text{ is even}\}$. Since permutation parity is a homomorphism, Δ_i^0 is a subgroup of index at most two. We define $\widetilde{B}_i = \Delta_i^0 \backslash \mathcal{T}_i$. The construction ensures that the natural projection $p : \widetilde{B}_i \to B_i$ is a graph cover of degree $[\Delta_i : \Delta_i^0] \leqslant 2$. We now verify that Δ_i^0 consists exactly of the elements admitting the representative required by Morgenstern's construction.

To that end, the local description in [RSV19, §2.6] gives a quaternion element u with $u^2 = t$. Every nonzero representative can be written as $x = u^m y$, where $m \in \mathbb{Z}$ and the coefficients of both y and y^{-1} have denominators not divisible by t. Under ρ, the element u acts by a transposition, while y acts by an even permutation. It follows that $\rho([x])$ is even exactly when m is even. If $m = 2a$, then $x = t^a y$, so the scalar rescaling $t^{-a}x = y$ gives the required representative. Conversely, every representative satisfying the requirement acts by an even permutation.

Combining the above and applying [Mor94, Theorem 3.2], we get

$$\lambda_i(X) = \left\| R_i|_{\ell_0^2(W)} \right\| \leqslant \frac{2\sqrt{s_i}}{s_i+1}. \qquad \square$$

References

[AAV13] Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum pcp conjecture. *Acm sigact news*, 44(2):47–79, 2013.

[ABN23] Anurag Anshu, Nikolas P. Breuckmann, and Chinmay Nirkhe. NLTS hamiltonians from good quantum codes. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1090–1096. ACM, 2023.

[AE15] Dorit Aharonov and Lior Eldar. Quantum locally testable codes. *SIAM Journal on Computing*, 44(5):1230–1262, 2015.

[ALG20] Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. In *61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1319–1330. IEEE, 2020.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AN02] Dorit Aharonov and Tomer Naveh. Quantum NP—a survey, 2002.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[BE21] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced product quantum codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021.

[BFLS91] László Babai, Lance Fortnow, Leonid A Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC 1991*, pages 21–32, 1991.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[BLM24] Mitali Bafna, Noam Lifshitz, and Dor Minzer. Constant degree direct product testers with small soundness. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 862–869. IEEE, 2024.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[BMVY25] Mitali Bafna, Dor Minzer, Nikhil Vyas, and Zhiwei Yun. Quasi-linear size pcps with small soundness from HDX. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 45–53. ACM, 2025.

[BNZ26] Mitali Bafna, Quynh T. Nguyen, and Tina Zhang. Gap amplification for local hamiltonians with combinatorial soundness. In *Proceedings of the 67th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2026)*, 2026. To appear.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short pcps with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[BSVW03] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 612–621. ACM, 2003.

[BV26] Mitali Bafna and Nikhil Vyas. Private PCPs from Product Expansion. *Electronic Colloquium on Computational Complexity (ECCC)*, 33(150), 2026. TR26-150.

[CHN+24] Andrew Cross, Zhiyang He, Anand Natarajan, Mario Szegedy, and Guanyu Zhu. Quantum locally testable code with constant soundness. *Quantum*, 8:1501, 2024.

[DDL24] Yotam Dikstein, Irit Dinur, and Alexander Lubotzky. Low acceptance agreement tests via bounded-degree symplectic hdxs. *CoRR*, abs/2402.01078, 2024. To appear in FOCS 2024.

[DEL+22] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Stoc 2022. pages 357–374. ACM, 2022.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[DLV24] Irit Dinur, Ting-Chun Lin, and Thomas Vidick. Expansion of high-dimensional cubical complexes: with application to quantum locally testable codes. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 379–385. IEEE, 2024.

[DLZ25] Irit Dinur, Siqi Liu, and Rachel Yun Zhang. New codes on high dimensional expanders. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, Toronto, Canada, August 5-8, 2025*, volume 339 of *LIPIcs*, pages 27:1–27:42. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[EH17] Lior Eldar and Aram W. Harrow. Local hamiltonians whose ground states are hard to approximate. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–438. IEEE, 2017.

[EK16] Shai Evra and Tali Kaufman. Bounded degree cosystolic expanders of every dimension. In *STOC 2016*, pages 36–48, 2016.

[EKZ20] Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders. In *FOCS 2020*, pages 218–227. IEEE, 2020.

[GKO$^+$18] Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the Gilbert–Varshamov bound. *IEEE Transactions on Information Theory*, 64(8):5813–5831, 2018.

[GL25] Louis Golowich and Ting-Chun Lin. Quantum LDPC codes with transversal non-Clifford gates via products of algebraic codes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 689–696. ACM, 2025.

[Got14] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information and Computation*, 14(15–16):1338–1371, 2014.

[GS02] Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. In *FOCS 2002, Vancouver, BC, Canada*. IEEE Computer Society, 2002.

[Has16] Matthew B Hastings. Quantum codes from high-dimensional manifolds. *arXiv preprint arXiv:1608.05089*, 2016.

[HHO21] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: Breaking the $N^{1/2}\,\mathrm{polylog}(N)$ barrier for quantum LDPC codes. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1276–1288. ACM, 2021.

[HLM$^+$25] Jun-Ting Hsieh, Alexander Lubotzky, Sidhanth Mohanty, Assaf Reiner, and Rachel Yun Zhang. Explicit lossless vertex expanders. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 894–911. IEEE, 2025.

[KKL14] Tali Kaufman, David Kazhdan, and Alexander Lubotzky. Ramanujan complexes and bounded degree topological expanders. In *FOCS 2014*, pages 484–493, 2014.

[KMRZS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *Journal of the ACM*, 64(2):11:1–11:42, 2017.

[KP25] Gleb Kalachev and Pavel Panteleev. Maximally extendable product codes are good coboundary expanders. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 1512–1524. IEEE, 2025.

[LLZ22] Anthony Leverrier, Vivien Londe, and Gilles Zémor. Towards local testability for quantum coding. *Quantum*, 6:661, 2022.

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

[LSV05] Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Ramanujan complexes of type $\tilde{A}_d$. *Israel journal of Mathematics*, 149:267–299, 2005.

[Mor94] Moshe Morgenstern. Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *J. Combin. Theory Ser. B*, 62(1):44–62, 1994.

[NP25] Quynh T Nguyen and Christopher A Pattison. Quantum fault tolerance with constant-space and logarithmic-time overheads. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 730–737, 2025.

[PK22a] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *STOC 2022*, pages 375–388. ACM, 2022.

[PK22b] Pavel Panteleev and Gleb Kalachev. Quantum LDPC codes with almost linear minimum distance. *IEEE Transactions on Information Theory*, 68(1):213–229, 2022.

[PK24] Pavel Panteleev and Gleb Kalachev. Maximally extendable sheaf codes. *arXiv preprint arXiv:2403.03651*, 2024.

[PS94] Alexander Polishchuk and Daniel A Spielman. Nearly-linear size holographic proofs. In *STOC 1994*, pages 194–203, 1994.

[RSV19] Nithi Rungtanapirom, Jakob Stix, and Alina Vdovina. Infinite series of quaternionic 1-vertex cube complexes, the doubling construction, and explicit cubical Ramanujan complexes. *Internat. J. Algebra Comput.*, 29(6):951–1007, 2019.

[Ser80] Jean-Pierre Serre. *Trees*. Springer-Verlag, Berlin-New York, 1980. Translated from the French by John Stillwell.

[SS94] Michael Sipser and Daniel A. Spielman. Expander codes. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, November 20-22, 1994*, pages 566–576. IEEE Computer Society, 1994.

[SV26] Baocheng Sun and Thomas Vidick. Probabilistically checking quantum proofs, with interaction. In *41st Computational Complexity Conference (CCC 2026)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:49. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2026.

[TZ14] Jean-Pierre Tillich and Gilles Zémor. Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il