



Decoding quantum Tanner codes

Anthony Leverrier, Gilles Zémor

► To cite this version:

Anthony Leverrier, Gilles Zémor. Decoding quantum Tanner codes. IEEE Transactions on Information Theory, 2022, 69 (8), pp.5100-5115. 10.1109/TIT.2023.3267945 . hal-04277199

HAL Id: hal-04277199

<https://inria.hal.science/hal-04277199>

Submitted on 9 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Decoding quantum Tanner codes

Anthony Leverrier*, Gilles Zémor†

December 9, 2022

Abstract

We introduce sequential and parallel decoders for quantum Tanner codes. When the Tanner code construction is applied to a sufficiently expanding square complex with robust local codes, we obtain a family of asymptotically good quantum low-density parity-check codes. In this case, our decoders provably correct arbitrary errors of weight linear in the code length, respectively in linear or logarithmic time. The same decoders are easily adapted to the expander lifted product codes of Panteleev and Kalachev. Along the way, we exploit recently established bounds on the robustness of random tensor codes to give a tighter bound on the minimum distance of quantum Tanner codes.

1 Introduction and Overview

Quantum low-density parity-check (LDPC) codes hold the promise of drastically reducing the required overhead for fault-tolerant quantum computing compared to approaches based on the surface code [1–5]. Rather surprisingly, this advantage was already established for the class of hypergraph product codes [6] that predates the recent series of breakthroughs [7–9] culminating with the discovery of asymptotically good quantum LDPC codes [10, 11], that is, quantum codes of length n with a constant encoding rate and a minimum distance d linear in n . These new codes should significantly improve the performance of quantum computers that can implement long-range gates between arbitrary qubits, for instance ion-based [12], photonic [13], or Rydberg-based [14] architectures. To take full advantage of these capabilities, it is crucial to devise efficient decoding algorithms so as to keep the errors under control during an execution of a quantum algorithm. This in particular requires highly parallelisable decoders that run in logarithmic time, since new errors keep accumulating while the classical decoder tries to identify errors that have occurred earlier. We present such an algorithm for the family of quantum Tanner codes [11]. The same decoder also serves as the main subroutine for the decoding of the expander lifted product codes of Panteleev and Kalachev [10].

*Anthony Leverrier is with Inria Paris, France (e-mail: anthony.leverrier@inria.fr)

†Gilles Zémor is with the Institut de Mathématiques de Bordeaux UMR 5251, Université de Bordeaux, 351 cours de la Libération - F33405, Talence, France (e-mail: zemor@math.u-bordeaux.fr).

There are two interesting settings for decoding algorithms, depending on whether the errors are arbitrary (or adversarial) or whether they follow a simple stochastic model (such as independent and identically distributed errors or local stochastic errors [1]). Before the recent breakthroughs yielding codes with a linear minimum distance, the distinction was crucial because experimentally relevant errors have a weight linear in the code length (since each qubit suffers an error with constant probability) and could only be dealt with by making some assumptions about their distribution. This is problematic in the context of fault tolerance because correlations between errors are essentially impossible to track down. While several decoders perform reasonably well against random noise, even with a noisy syndrome extraction [15–21], they cannot handle much more than $\sqrt{n}$ errors in an adversarial setting [22, 23].

We will first review the construction of quantum Tanner codes and then present our logarithmic-time decoding algorithm. It is strongly inspired by a mostly sequential, linear-time decoder analysed in [24], and is an adaptation of the small-set-flip decoder initially designed for hypergraph product codes [22], which was also recently applied to other good quantum LDPC codes [25–27]. Along the way we will give a sharper estimate for the minimum distance of quantum Tanner codes and present a sequential decoder that is significantly simpler than that of [24], both conceptually and technically.

Quantum Tanner codes.— These codes are a generalisation of classical Tanner codes [28, 29]. They are obtained by enforcing local linear constraints (corresponding to the dual of a small tensor code of constant size) of constant weight on qubits associated with the 2-faces (squares) of a square complex [8, 30]. The square complex that we use appears in the work of Panteleev and Kalachev [10] and can also be thought of as a quadripartite version of the left-right Cayley complex of Dinur *et al.* [30]. It is an incidence structure between a set V of vertices, two sets of edges E_A and E_B , that we will refer to as A -edges and B -edges, and a set Q of squares. The vertex-set V is partitioned into four subsets $V = V_{00} \cup V_{01} \cup V_{10} \cup V_{11}$, corresponding to four copies of a fixed group G , that is, $V_{ij} = G \times \{i, j\}$. We also have two self-inverse subsets $A = A^{-1}$ and $B = B^{-1}$ of the group G and assume for simplicity that A and B are of the same cardinality Δ . For $i \in \{0, 1\}$, two vertices $v = (g, i0) \in V_{i0}$ and $v' = (g', i1) \in V_{i1}$ are related by an A -edge if $g' = ag$ for some $a \in A$. Similarly, for $j \in \{0, 1\}$, vertices $v = (g, 0j)$ and $v' = (g', 1j)$ are related by a B -edge if $g' = gb$ for some $b \in B$. The sets E_A and E_B make up the set of A -edges and B -edges respectively and define two graphs $\mathcal{G}_A = (V, E_A)$, $\mathcal{G}_B = (V, E_B)$, each of which consists of two disjoint copies of the double cover of a Cayley graph over the group G (with generator set A for $\mathcal{G}_A$, and B for $\mathcal{G}_B$). Next, the set Q of squares is defined as the set of 4-subsets of vertices of the form $\{(g, 00), (ag, 01), (agb, 11), (gb, 10)\}$, with the four vertices belonging to distinct copies of G .

If we restrict the vertex set to $V_0 := V_{00} \cup V_{11}$, every square is now incident to only two vertices: one in V_{00} and one in V_{11} . The set of squares can then be seen as a set of edges on V_0 , and it therefore defines a bipartite graph that we denote by $\mathcal{G}_0^\square = (V_0, Q)$. Similarly, the restriction to the vertices of $V_1 := V_{01} \cup V_{10}$ defines the graph $\mathcal{G}_1^\square$, which is an exact replica of $\mathcal{G}_0^\square$: both graphs are defined over two copies of the group G , with

$g, g' \in G$ being related by an edge whenever $g' = agb$ for some $a \in A, b \in B$. For any vertex v , we denote by $Q(v)$ the Q -neighbourhood of v defined as the set of squares incident to v . The Q -neighbourhood $Q(v)$ has cardinality Δ^2 and is isomorphic to the product set $A \times B$: the situation is illustrated on Fig. 1.

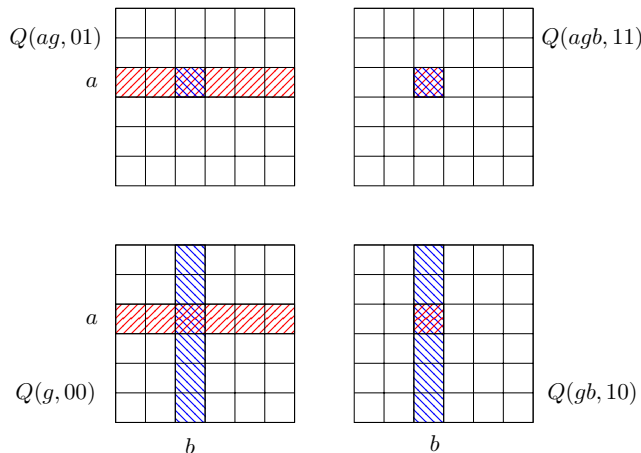


Figure 1: The four Q -neighbourhoods $Q(v)$ that contain the square $\{(g, 00), (ag, 01), (agb, 11), (gb, 10)\}$ depicted in red and blue. The Q -neighbourhoods of two vertices connected by an A -edge (resp. a B -edge) share a row (resp. a column) depicted in red (resp. a column in blue). The labeling is chosen to ensure that a given square, such as the one in red and blue, is indexed similarly, by (a, b) here, in the four Q -neighbourhoods. The σ_x -type generators are codewords of $C_A \otimes C_B$ in the Q -neighbourhoods of $V_{00} \cup V_{11}$; the σ_z -type generators are codewords of $C_A^\perp \otimes C_B^\perp$ in the Q -neighbourhoods of $V_{01} \cup V_{10}$. They automatically commute since their support can only intersect on a shared row or column (as depicted), and the orthogonality of the local codes ensure that they commute on this row or column.

A classical *Tanner code* on a Δ -regular graph $\mathcal{G} = (V, E)$ is the set of words of $\mathbb{F}_2^E$ such that on the edge neighbourhood of every vertex $v \in V$, we see a codeword of a small code C of length Δ [28]. We denote the resulting code by $\text{Tan}(\mathcal{G}, C) \subset \mathbb{F}_2^E$. Expander codes, obtained by combining good codes C with an expander graph G , gave the first explicit family of good classical LDPC codes with a constant encoding rate, a linear minimum distance and an efficient decoder [29].

Quantum Tanner codes are quantum CSS codes formed by two classical Tanner codes $\mathcal{C}_0$ and $\mathcal{C}_1$ with support on the set Q of squares of a square complex as above. The CSS construction [31, 32] requires both codes to satisfy the orthogonality condition $\mathcal{C}_0^\perp \subset \mathcal{C}_1$. To this end, we define local codes on the space $\mathbb{F}_2^{A \times B}$ that we may think of as the space of matrices whose rows (columns) are indexed by A (by B). If $C_A \subset \mathbb{F}_2^A$ and $C_B \subset \mathbb{F}_2^B$ are two linear codes, we define the *tensor* (or product) code $C_A \otimes C_B$ as the space of matrices x such that for every $b \in B$ the column vector $(x_{ab})_{a \in A}$ belongs to C_A and for

every $a \in A$ the row vector $(x_{ab})_{b \in B}$ belongs to C_B . Recalling that the dual $C^\perp$ of a code $C \subset \mathbb{F}_2^\Delta$ is the set of words orthogonal to all words in C , we define $\mathcal{C}_0$ and $\mathcal{C}_1$ to be the classical Tanner codes $\mathcal{C}_0 := \text{Tan}(\mathcal{G}_0^\square, (C_A \otimes C_B)^\perp)$ and $\mathcal{C}_1 := \text{Tan}(\mathcal{G}_1^\square, (C_A^\perp \otimes C_B^\perp)^\perp)$, with bits associated to each square of Q and local constraints enforced at the vertices of V_0 and V_1 , respectively. To check the orthogonality condition between the two codes, it is convenient to look at their generators (or parity-checks). We define a σ_x -generator for $\mathcal{C}_0$ (resp. a σ_z -generator for $\mathcal{C}_1$) as a vector of $\mathbb{F}_2^Q$ whose support lies entirely in the Q -neighbourhood $Q(v)$ of V_0 (resp. V_1), and which is equal to a codeword of $C_A \otimes C_B$ (resp. $C_A^\perp \otimes C_B^\perp$) on $Q(v)$. The Tanner code $\mathcal{C}_0$ (resp. $\mathcal{C}_1$) is defined as the set of vectors orthogonal to all σ_x -generators (resp. σ_z -generators). The commutation between both types of generators follows from the fact that if a σ_x -generator on $v_0 \in V_0$ and a σ_z -generator on $v_1 \in V_1$ have intersecting supports, then v_0 and v_1 must be neighbours in the left-right Cayley complex and their Q -neighbourhoods must intersect on either a column or a row, on which the two generators equal codewords of C_A and $C_A^\perp$, or of C_B and $C_B^\perp$ (see Fig. 1). Since Δ is chosen constant with respect to n , we see that the generators of the code have constant weight and that each qubit only appears in a constant number of generators: the resulting quantum Tanner code is therefore a quantum LDPC code by definition. Choosing C_A and C_B of rates ρ and $1 - \rho$, so that both $\mathcal{C}_0$ and $\mathcal{C}_1$ have rate $\rho(1 - \rho)$, yields a quantum code with encoding rate $k/n \geq (1 - 2\rho)^2$, as can be seen by counting the number of generators of the code. Here, k is the number of logical qubits.

Crucial for the analysis of quantum Tanner codes is the *robustness* of the small tensor codes. Very recently, [26, 33] showed that if C_A and C_B are chosen randomly with some fixed rate, then their tensor product is κ -robust for some constant κ independent of the code length Δ . This means that for any *dual tensor codeword* $x \in (C_A^\perp \otimes C_B^\perp)^\perp$, there exist $c \in C_A \otimes \mathbb{F}_2^B, r \in \mathbb{F}_2^A \otimes C_B$ such that $x = c + r$ and

$$|x| \geq \kappa \Delta (\|c\| + \|r\|), \quad (1)$$

where $|\cdot|$ denotes the Hamming weight and $\|\cdot\|$ counts the number of nonzero columns (resp. rows) of c (resp. r). Note that the distance of the random codes C_A, C_B and their dual will be at least $\delta \Delta$ for some $\delta > 0$ with overwhelming probability. We show in Section 3 that these two facts imply a linear lower bound for the distance of the quantum Tanner code.

Theorem 1. *For a constant Δ large enough, the quantum Tanner codes described in the construction above with a Ramanujan left-right Cayley complex have a linear minimum distance:*

$$d_{\min} \geq \frac{\delta^2 \kappa^2}{256 \Delta} n.$$

The dependency in Δ of this bound is tight since there exist logical errors of weight $\leq n/\Delta$ [11]. The bound is sharper than that of [11], and its proof is somewhat simpler.

We note that quantum Tanner codes have found recent applications outside of coding theory: they can fool optimisation algorithms exploiting the sum-of-squares hierarchy [34] and are instrumental in the recent proof of the NLTS theorem in quantum complexity theory [35, 36].

Decoding quantum Tanner codes.— We focus here on decoding σ_x -type errors, which are detected by σ_z -generators. This is without loss of generality for a CSS code. The general strategy outlined in [24] consists in defining a *mismatch* vector associated with the error $\mathbf{e} \in \mathbb{F}_2^Q$, that summarises how the local decoders associated to the local code around each vertex may disagree about the error, and then try to locally modify this mismatch in order to reduce its weight. It is natural to see the error $\mathbf{e}$ as a collection of local views on the Q -neighbourhoods of vertices of V_1 : abusing notation slightly, we can write $\mathbf{e} = \{e_v\}_{v \in V_1}$, with local views e_v restricted to $Q(v)$. For each vertex $v \in V_1$, one can compute (in parallel if needed) a local error ε_v with support on $Q(v)$ of minimal Hamming weight yielding the same local syndrome as e_v . This gives a decomposition of the local views of the error $e_v = \varepsilon_v + c_v + r_v$, with $c_v \in C_A \otimes \mathbb{F}_2^B$, $r_v = \mathbb{F}_2^A \otimes C_B$. Note that the κ -robustness property will apply to any such $c_v + r_v$. The issue is that the local views $\{\varepsilon_v\}_{v \in V_1}$ of the decoder are in general not consistent and do not define a global error candidate. We measure this inconsistency by defining the mismatch vector:

$$Z := \sum_{v \in V_1} \varepsilon_v \in \mathbb{F}_2^Q. \quad (2)$$

If it is equal to zero, it means that each square/qubit is affected the same value for the two views it belong to, and the decoder is able to define a global error. Otherwise, the support of Z corresponds to the set of squares for which the local views of the decoder disagree. On the other hand, the local views e_v of the error *are* consistent, and satisfy $\sum_{v \in V_1} e_v = 0$ since each square appears twice in this sum. We can rewrite the mismatch (2) as

$$Z = \sum_{v \in V_1} r_v + c_v = C_0 + R_0 + C_1 + R_1,$$

with $C_i := \sum_{v \in V_{\bar{i}}} c_v$ and $R_j := \sum_{v \in V_{j\bar{j}}} r_v$, where we denote $\bar{i} := 1 - i$, $\bar{j} := 1 - j$. This convenient representation of Z highlights the symmetry of the local representations of $C_j + R_i$ on vertices of V_{ij} , and one may look for local modifications of Z by adding to it a dual tensor codeword on the Q -neighbourhood of any of the four types of vertices.

The main subroutine of the decoding algorithm consists in finding a valid decomposition $Z = \hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1$. The simplest approach to this is a sequential decoder that proceeds in a greedy fashion: simply look for a vertex v , and local codewords c_v, r_v such that flipping $c_v + r_v$ (replacing Z by $Z + c_v + r_v$) decreases the Hamming weight of Z . Our main technical result exploits the expansion properties of the square complex together with the robustness of the local codes to establish the existence of some vertex $v \in V$ and local dual tensor codeword $c_v + r_v$ that decreases the weight of Z almost optimally. Instrumental in this analysis is the notion of the *norm* of a representation of a mismatch vector Z which can be expressed in many different ways as $Z = C_0 + R_0 + C_1 + R_1$. The “column” vector C_i is expressed as a sum of local vectors with disjoint supports, each of which is a codeword of C_A supported by a column common to two Q -neighbourhoods of two different types of vertices (V_{ii} and $V_{\bar{i}\bar{i}}$). The row vectors R_j admit similar decompositions into C_B components. We will write $\|C_i\|$ ($\|R_j\|$) to denote the number of non-zero C_A codewords (C_B codewords) in the decomposition of C_i (R_j). We will say

that a decomposition $Z = C_0 + R_0 + C_1 + R_1$ is *minimal* if it minimises its norm, namely the value $\|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|$. Finally, we shall say that a vertex $v \in V_{ij}$ is *active* for a decomposition (C_0, R_0, C_1, R_1) if $C_j + R_i$ is non-zero on $Q(v)$.

Theorem 2. *Fix $\varepsilon \in (0, 1)$. If, for every $i, j \in \{0, 1\}$, the sets of active vertices $S_{ij} \subset V_{ij}$ for a minimum decomposition of a mismatch vector Z satisfy*

$$|S_{ij}| \leq \frac{1}{2^{12}} \delta^2 \varepsilon^3 \kappa |V_{00}|$$

then there exists some vertex v and some codeword x_v of the dual tensor code such that

$$|Z| - |Z + x_v| \geq (1 - \varepsilon)|x_v|.$$

Provided the initial error is not too large, we show that the condition on the number of active vertices in Theorem 2 keeps being satisfied, so that one can keep on iterating the procedure, finding a local codeword to flip at each step, and that it will eventually give a decomposition $Z = \hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1$ of the mismatch. This decomposition gives a correct guess $\hat{e} = \sum_{v \in V_{00}} \varepsilon_v + \hat{C}_0 + \hat{R}_0$ for the error, that differs from e by a sum of generators. This naturally yields a sequential decoder, with parameter ε , described formally later.

It is easy to see that the pre-processing (defining the mismatch) and the post-processing (computing $\hat{e}$ from the decomposition of Z) can be performed in parallel. To get a fully parallel decoder, one simply needs a parallel procedure for decomposing Z . This is obtained by looking for vertices that would be candidates for the sequential decoder of parameter $1/2$, *i.e.* finding some x_v such that $|Z| - |Z + x_v| \geq |x_v|/2$. In order to apply several sequential iterations in parallel, the decoder needs the different $c_v + r_v$'s that it will identify to have disjoint supports. This is achieved if one restricts the set of candidate vertices v to a single set V_{ij} : so the decoder applies four parallel substeps, for each of the four sets of vertices $V_{00}, V_{01}, V_{10}, V_{11}$, each of which consists of applying simultaneously all sequential iterations for all the candidate vertices it has identified in the current V_{ij} . We will show that after those four substeps, the value of the current mismatch $|\hat{Z}|$ has been decreased significantly, provided the original error vector has sufficiently small weight. The pseudo-code of all the algorithms is presented in Section 4.

Theorem 3. *Fix $\varepsilon \in (0, 1/2)$, and $\mu \in (\varepsilon, 1/6)$. If the Hamming weight $|e|$ is less than*

$$\frac{1}{2^{12}} \min\left(\frac{\varepsilon^3}{16}, \kappa\right) \delta^2 \kappa^2 \frac{n}{\Delta},$$

then the parallel decoder returns a valid correction in logarithmic time.

The value of ε can be tuned to what one wants to achieve. A larger ε will increase the weight of correctable errors but increases the number of required parallel steps, while a smaller ε decreases the number of parallel steps but only decodes smaller weight errors.

The main technical result needed to prove that the parallel decoder converges after a logarithmic number of iterations is that there is a linear number of vertices v that can be updated at each step to decrease the weight of Z . The idea to prove their existence consists in running the sequential algorithm virtually and then establishing that many of the updates corresponding to the sequential decoder can in fact be performed at the same time. It is detailed in the proof of Theorem 20 in Section 5.

Remark: we have not tried to optimise the power of 2 numerical constants in Theorems 1,2,3 for the sake of readability.

Decoding expander lifted product codes.—We remark that the same decoding algorithm can be applied to the expander lifted product codes of [10]. Indeed, their decoding can be reduced (with a parallel procedure) to that of quantum Tanner codes as explained in [24]. Furthermore, the decoder also works for hypergraph products of two classical Tanner codes, albeit with smaller correction capabilities since these codes have a distance scaling like $\sqrt{n}$.

Discussion and open questions.—Rapid improvements of decoding algorithms over the last decade have led to a surge of interest for LDPC as a path towards hardware-efficient fault-tolerant quantum computing. The unexpected discovery of good quantum LDPC codes with parameters close to optimal will likely impact this research direction in major ways. The parallel decoding algorithm presented here is a first step towards this goal. Of course, a great number of open questions remains. While our proof requires codes of large size, the decoder is well defined for any quantum Tanner code or lifted product code, and it will be interesting to investigate its performance against random noise. A practical decoder should be robust to errors in the syndrome extraction process. We believe this is the case here, and that an analysis along the lines of [3] could be extended to the present decoder. It also does not suffice to correct errors in order to perform a computation, and one must be able to apply logical gates in a fault-tolerant manner [37]. At the moment, a complete understanding of the logical operators for good LDPC codes is still lacking. Finally, and crucially, we will need examples of good codes of reasonably small size if such codes are to be implemented in real devices.

2 Technical preliminaries

2.1 Graph expansion

In this section, we recall some useful facts about graph expansion.

Let $\mathcal{G} = (V, E)$ be a graph. Graphs will be undirected but may have multiple edges. For $S, T \subset V$, let $E(S, T)$ denote the multiset of edges with one endpoint in S and one endpoint in T . Let $\mathcal{G}$ be a connected Δ -regular graph on n vertices, and let $\Delta = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ be the eigenvalues of the adjacency matrix of $\mathcal{G}$. For $n \geq 3$, we define $\lambda(\mathcal{G}) := \max\{|\lambda_i|, \lambda_i \neq \pm\Delta\}$. The graph $\mathcal{G}$ is said to be *Ramanujan* if $\lambda(\mathcal{G}) \leq 2\sqrt{\Delta - 1}$.

We recall the following version of the expander mixing lemma (see *e.g.* [38]) for bipartite graphs.

Lemma 4 (Expander mixing lemma). *Let $\mathcal{G}$ be a connected Δ -regular bipartite graph on the vertex set $V_0 \cup V_1$. For any pair of sets $S \subset V_0, T \subset V_1$, it holds that*

$$|E(S, T)| \leq \frac{\Delta}{|V_0|} |S||T| + \lambda(\mathcal{G}) \sqrt{|S||T|}.$$

2.2 Tanner codes

A binary linear code of length n is an $\mathbb{F}_2$ -linear subspace of $\mathbb{F}_2^n$. For sets E of cardinality $|E| = n$, it will be convenient for us to identify $\mathbb{F}_2^n$ with $\mathbb{F}_2^E$, which we can think of as the space of functions from E to $\mathbb{F}_2$. Identification with $\mathbb{F}_2^n$ amounts to defining a one-to-one map between E and $[n] = \{1, 2, \dots, n\}$, *i.e.* a numbering of the elements of E .

Let $\mathcal{G} = (V, E)$ be a regular graph of degree Δ , and for any vertex v denote by $E(v)$ the set of edges incident to v . Assume an identification of $\mathbb{F}_2^{E(v)}$ with $\mathbb{F}_2^\Delta$ for every $v \in V$. Let $x \in \mathbb{F}_2^E$ be a vector indexed by (or a function defined on) the set E . Let us define the *local view* of x at vertex v as the subvector $x_v := (x_e)_{e \in E(v)}$, *i.e.* x restricted to the edge-neighbourhood $E(v)$ of v .

Let C_0 be a linear code of length Δ , dimension $k_0 = \rho_0 \Delta$, and minimum distance $d_0 = \delta_0 \Delta$. We define the Tanner code [28] associated to $\mathcal{G}$ and C_0 as

$$\text{Tan}(\mathcal{G}, C_0) := \{x \in \mathbb{F}_2^E : x_v \in C_0 \text{ for all } v \in V\}.$$

In words, the Tanner code is the set of vectors over E all of whose local views lie in C_0 . By counting the number of linear equations satisfied by the Tanner code, we obtain

$$\dim \text{Tan}(\mathcal{G}, C_0) \geq (2\rho_0 - 1)n. \quad (3)$$

We also have the bound [29, 39] on the minimum distance d of the Tanner code:

$$d \geq \delta_0(\delta_0 - \lambda(\mathcal{G})/\Delta)n.$$

Therefore, if $(\mathcal{G}_i)$ is a family of Δ -regular expander graphs with $\lambda(\mathcal{G}_i) \leq \lambda < d_0$, and if $\rho_0 > 1/2$, then the associated family of Tanner codes has rate and minimum distance which are both $\Omega(n)$, meaning we have an asymptotically good family of codes, as was first shown in [29].

2.3 Quantum CSS codes

A quantum CSS code is specific instance of a stabilizer code [40] that can be defined by two classical codes $\mathcal{C}_0$ and $\mathcal{C}_1$ in the ambient space $\mathbb{F}_2^n$, with the property that $\mathcal{C}_0^\perp \subset \mathcal{C}_1$ [32, 41]. It is a *low-density parity-check* (LDPC) code whenever both $\mathcal{C}_0$ and $\mathcal{C}_1$ are the kernels of sparse parity-check matrices. The resulting quantum code $\mathcal{Q} = (\mathcal{C}_0, \mathcal{C}_1)$ is a subspace of $(\mathbb{C}_2)^{\otimes n}$, the space of n qubits:

$$\mathcal{Q} := \text{Span} \left\{ \sum_{z \in \mathcal{C}_1^\perp} |x + z\rangle : x \in \mathcal{C}_0 \right\},$$

where $\{|x\rangle : x \in \mathbb{F}_2^n\}$ is the canonical basis of $(\mathbb{C}_2)^{\otimes n}$. The dimension k of the code counts the number of logical qubits and is given by

$$k = \dim(\mathcal{C}_0/\mathcal{C}_1^\perp) = \dim \mathcal{C}_0 + \dim \mathcal{C}_1 - n.$$

Its minimum distance is $d = \min(d_X, d_Z)$ with

$$d_X = \min_{w \in \mathcal{C}_0 \setminus \mathcal{C}_1^\perp} |w|, \quad d_Z = \min_{w \in \mathcal{C}_1 \setminus \mathcal{C}_0^\perp} |w|.$$

We denote the resulting code parameters by $\llbracket n, k, d \rrbracket$ and say that a code family $(\mathcal{Q}_n)_n$ is *asymptotically good* if its parameters are of the form

$$\llbracket n, k = \Theta(n), d = \Theta(n) \rrbracket.$$

An n -qubit Pauli error $E_1 \otimes \dots \otimes E_n$ with $E_i \in \{\mathbb{1}, \sigma_X, \sigma_Y, \sigma_Z\}$ ¹ is conveniently described by two n -bit strings $(e_0, e_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ via the mapping

$$\mathbb{1} \mapsto (0, 0), \quad \sigma_X \mapsto (1, 0), \quad \sigma_Y \mapsto (1, 1), \quad \sigma_Z \mapsto (0, 1),$$

which forgets global phases. The parity-check matrices of $\mathcal{C}_0$ and $\mathcal{C}_1$ give rise to syndrome maps $\sigma_0, \sigma_1 : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ that associate a pair of syndromes $(\sigma_0(e_0), \sigma_1(e_1)) \in \mathbb{F}_2^m \times \mathbb{F}_2^m$ to any n -qubit Pauli error $(e_0, e_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. The decoding problem for a stabilizer code is as follows: given a syndrome $(\sigma_0(e_0), \sigma_1(e_1))$, recover the error up to an element of the stabilizer group, that is return $(\hat{e}_0, \hat{e}_1)$ such that $e_0 + \hat{e}_0 \in \mathcal{C}_1^\perp$ and $e_1 + \hat{e}_1 \in \mathcal{C}_0^\perp$.

While an optimal decoding of *random* errors would typically exploit possible correlations between e_0 and e_1 , it is always possible to correct both errors independently. Here, we will be concerned with the adversarial setting where e_0 and e_1 are of sufficiently low weight, but otherwise arbitrary. In that case, both errors should be decoded independently, and we will focus on the case where $(e_0 = 0, e_1 = e)$ in the sequel.

2.4 Left-right Cayley complexes (quadripartite version)

The square complex we shall rely on for the construction first appeared in [10] as a balanced product of double covers of non-bipartite Cayley graphs. For the sake of simplicity, we will rather use the language of left-right Cayley complexes in their quadripartite version. A *left-right Cayley complex* X is introduced in [30] from a group G and two sets of generators $A = A^{-1}$ and $B = B^{-1}$. As in [30] we will restrict ourselves, for the sake of simplicity, to the case $|A| = |B| = \Delta$. The complex is made up of vertices, A -edges, B -edges, and squares. The vertex set consists of four copies of the group G in the quadripartite version, $V = V_{00} \cup V_{10} \cup V_{01} \cup V_{11}$ with $V_{ij} = G \times \{ij\}$. The advantage of this quadripartite version, also considered in [10] and [42], is that it does not require any additional assumption on the choice of group and generators, for instance that $ag \neq gb$ for all $g \in G, a \in A, b \in B$, as in [30]. We will also use the notation $V_0 := V_{00} \cup V_{11}$ and $V_1 := V_{01} \cup V_{10}$. The A -edges are pairs of vertices of the form

¹The 1-qubit Pauli matrices are defined by $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\sigma_Y = i\sigma_X\sigma_Z$.

$\{(g, i0), (ag, i1)\}$ and B -edges are of the form $\{(g, 0j), (gb, 1j)\}$ for $g \in G, a \in A, b \in B, i, j = 0, 1$. We denote by E_A and E_B these two edge sets. The associated graphs are denoted by $\mathcal{G}_A = (V, E_A)$ and $\mathcal{G}_B = (V, E_B)$. A *square* is a set of four vertices of the form $\{(g, 00), (ag, 01), (gb, 10), (agb, 11)\}$. The set of squares (or quadrangles) of the complex is denoted by Q . Every vertex is incident to exactly Δ^2 squares. For a vertex v , the set of incident squares is called the Q -neighbourhood, and denoted by $Q(v)$.

The sets of generators A and B will be chosen so that the Cayley graphs $\text{Cay}(G, A)$ and $\text{Cay}(G, B)$ are non-bipartite Ramanujan graphs. It should be understood that when writing $\text{Cay}(G, A)$ we implicitly mean the Cayley graph defined by left multiplication by elements of A , while $\text{Cay}(G, B)$ stands for the Cayley graph defined by right multiplication by elements of B . The sets A and B could in principle be chosen to be identical, but we keep a distinct notation for both sets, in particular in order to allow the above abuse of notation to be non-confusing.

We see that the subset of edges of E_A that connect vertices of V_{00} to vertices of V_{01} make up a double cover of the Cayley graph $\text{Cay}(G, A)$, the edges E_A that connect V_{10} to V_{11} make up a second copy of the same double cover. Therefore, the graph $\mathcal{G}_A$ is a disjoint union of two copies of the double cover of $\text{Cay}(G, A)$. Similarly, $\mathcal{G}_B$ is a disjoint union of two copies of the double cover of $\text{Cay}(G, B)$.

Let us introduce one additional graph that exists on the complex X , and that we denote by $\mathcal{G}^\square$. This graph puts an edge between all pairs of vertices of the form $\{(g, i), (agb, i)\}$, $g \in G, a \in A, b \in B, i = 0, 1$. The graph $\mathcal{G}^\square$ is therefore made up of two connected components, on V_0 and V_1 , that we denote by $\mathcal{G}_0^\square$ and $\mathcal{G}_1^\square$. We note that $\mathcal{G}^\square$ is regular of degree Δ^2 , and may have multiple edges.

If $\text{Cay}(G, A)$ and $\text{Cay}(G, B)$ are Ramanujan, then $\mathcal{G}^\square$ inherits some of their expansion properties. Specifically:

Lemma 5. *Assume that $\text{Cay}(G, A)$ and $\text{Cay}(G, B)$ are Ramanujan graphs, then*

$$\lambda(\mathcal{G}_0^\square) \leq 4\Delta, \quad \lambda(\mathcal{G}_1^\square) \leq 4\Delta.$$

The proof follows from the fact that the adjacency matrix of $\mathcal{G}^\square$ is the product of the adjacency matrices of $\mathcal{G}_A$ and $\mathcal{G}_B$, and that these two adjacency matrices commute, by definition of the square complex. See [11] for a little more detail.

2.5 Labelling Q -neighbourhoods

We will define Tanner codes on $\mathcal{G}_0^\square$ and $\mathcal{G}_1^\square$, which implies a labelling of the coordinates in every Q -neighbourhood $Q(v)$. There is a natural labeling of $Q(v)$ by the set $A \times B$, namely a one-to-one map $\phi_v : A \times B \rightarrow Q(v)$, which we now state explicitly.

We set

$$\begin{aligned} \text{for } v = (g, 00) \in V_{00}, \quad & \phi_v(a, b) = \{(g, 00), (ag, 01), (gb, 10), (agb, 11)\}, \\ \text{for } v = (g, 01) \in V_{01}, \quad & \phi_v(a, b) = \{(g, 01), (a^{-1}g, 00), (gb, 11), (a^{-1}gb, 10)\}, \\ \text{for } v = (g, 10) \in V_{10}, \quad & \phi_v(a, b) = \{(g, 10), (ag, 11), (gb^{-1}, 00), (agb^{-1}, 01)\}, \\ \text{for } v = (g, 11) \in V_{11}, \quad & \phi_v(a, b) = \{(g, 11), (a^{-1}g, 10), (gb^{-1}, 01), (a^{-1}gb^{-1}, 00)\}. \end{aligned}$$

The map ϕ_v thus defined is obviously one-to-one, and one easily checks that:

Any two vertices $v = (g, 0i)$ and $w = (gb, 1i)$, $i = 0, 1$, that are connected through a B -edge (labelled b), have a common “column”, i.e. their Q -neighbourhoods share exactly Δ squares that are labelled (a, b) , $a \in A$, in both $Q(v)$ and $Q(w)$.

Similarly,

Any two vertices $v = (g, i0)$ and $w = (ag, i1)$, $i = 0, 1$, that are connected through an A -edge (labelled a), have a common row, i.e. their Q -neighbourhoods share exactly Δ squares that are labelled (a, b) , $b \in B$, in both $Q(v)$ and $Q(w)$.

The situation is illustrated on Figure 1. Summarising, any two vertices connected by B -edge (an A -edge) have a common column (row) in their Q -neighbourhoods, that is labelled by the same $b \in B$ ($a \in A$).

2.6 Local codes

The constraints of a classical Tanner code consist of local constraints from small codes enforced on the edge-neighbourhood of each vertex. For quantum Tanner codes, now that all local Q -neighbourhoods are isomorphic to $A \times B$, we may put local constraints that are codewords of the tensor codes $C_A \otimes C_B$ and $C_A^\perp \otimes C_B^\perp$.

Recall that the generators of the quantum Tanner code correspond to a basis of $C_A \otimes C_B$ on each Q -neighbourhood of $V_{00} \cup V_{11}$ (for the σ_X -type generators) and to a basis of $C_A^\perp \otimes C_B^\perp$ on each Q -neighbourhood of $V_{01} \cup V_{10}$ for the σ_Z -type generators). The classical code $\mathcal{C}_0 \subset \mathbb{F}_2^Q$ correcting σ_Z -type errors is the Tanner code on the graph $\mathcal{G}_0^\square$ with local constraints corresponding to the dual tensor code $(C_A \otimes C_B)^\perp = C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$. With the notation of Section 2.2, $\mathcal{C}_0 = \text{Tan}(\mathcal{G}_0^\square, C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp)$. Similarly, the classical code $\mathcal{C}_1 \subset \mathbb{F}_2^Q$ correcting σ_X -type errors is the Tanner code on the graph $\mathcal{G}_1^\square$ with local constraints corresponding to the dual tensor code $(C_A^\perp \otimes C_B^\perp)^\perp = C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$, i.e. $\mathcal{C}_1 = \text{Tan}(\mathcal{G}_1^\square, C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B)$.

Summary of parameters. A large enough Δ is chosen, together with an infinite family of groups G with generating sets A, B , $|A| = |B| = \Delta$, such that the left Cayley graph $\text{Cay}(G, A)$ and the right Cayley graph $\text{Cay}(G, B)$ are Ramanujan. The quadripartite left-right square complex X is defined by G, A, B .

We take inner codes C_A and C_B such that $\dim C_B = \Delta - \dim C_A$. Defining the inner rate ρ such that $\dim C_A = \rho\Delta$, and counting the number of constraints, we obtain that the dimension of the quantum Tanner code $\mathcal{Q} = (\mathcal{C}_0, \mathcal{C}_1)$ of length $|Q|$ is at least $(1 - 2\rho)^2|Q|$.

2.7 Robustness of (dual) tensor codes

Besides the expansion properties of the left-right Cayley complex, the other required property to obtain good quantum LDPC codes is the robustness of the local dual tensor codes [11, 33]. In words, it states that any low-weight codeword of the dual tensor code $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ can be obtained as a sum $c_v + r_v$ where the Hamming weights

of c_v and r_v are not much larger than that of $c_v + r_v$. Very recently, better bounds on the robustness of dual tensor codes obtained from two random codes were obtained in [25, 26, 33]. In particular, [26, 33] prove essentially tight bounds. We recall the main result from [33], where robustness is called product expansion and is defined as follows: for two linear codes $C_A, C_B \subset \mathbb{F}_2^\Delta$, the dual tensor code $C_A \otimes \mathbb{F}_2^\Delta + \mathbb{F}_2^\Delta \otimes C_B$ is said to be κ -product expanding, if any codeword $x \in C_A \otimes \mathbb{F}_2^\Delta + \mathbb{F}_2^\Delta \otimes C_B$ can be written as $x = c + r$ with $c \in C_A \otimes \mathbb{F}_2^\Delta, r \in \mathbb{F}_2^\Delta \otimes C_B$ and

$$|c + r| \geq \kappa \Delta (\|c\| + \|r\|),$$

where $\|c\|$ ($\|r\|$) denotes the number of columns (rows) involved in the support of c (r).

Theorem 6 ([33]). *For every $\rho_A, \rho_B \in (0, 1)$, the dual tensor code $C_A \otimes \mathbb{F}_2^\Delta + \mathbb{F}_2^\Delta \otimes C_B$ obtained from a uniformly random pair of linear codes (C_A, C_B) , of lengths Δ and of codimensions $\lceil \rho_A \Delta \rceil$ and $\lceil \rho_B \Delta \rceil$ respectively, is κ -product-expanding with high probability as $\Delta \rightarrow \infty$ with*

$$\kappa = \frac{1}{2} \min \left(\frac{1}{4} H_2^{-1} \left(\frac{\rho_A}{8} \right) H_2^{-1} \left(\frac{\rho_B}{8} \right), H_2^{-1} \left(\frac{\rho_A \rho_B}{8} \right) \right).$$

Here H_2^{-1} is the inverse of the binary entropy function given by

$$H_2(x) := -x \log_2 x - (1 - x) \log_2 (1 - x).$$

Remark: for our application we have $\rho_B = 1 - \rho_A$. Therefore, Theorem 6 implies that both dual tensor codes $C_A \otimes \mathbb{F}_2^\Delta + \mathbb{F}_2^\Delta \otimes C_B$ and $C_A^\perp \otimes \mathbb{F}_2^\Delta + \mathbb{F}_2^\Delta \otimes C_B^\perp$ are with high probability κ -product-expanding for the same value κ .

3 Minimum distance

Let $\mathbf{x}$ be a codeword of $\mathcal{C}_1$ not in $\mathcal{C}_0^\perp$. We will derive a lower bound on the weight of $\mathbf{x}$, that will also be valid for the weight of a codeword of $\mathcal{C}_0$ not in $\mathcal{C}_1^\perp$, and therefore bound from below the minimum distance of the code.

The codeword $\mathbf{x}$ is a Tanner codeword on the graph $\mathcal{G}_1^\square$, with vertex set $V_1 = V_{01} \cup V_{10}$ and the dual tensor code $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ as inner code. The set Q of coordinates is partitioned into Q -neighbourhoods $Q(v)$ of vertices v of V_{01} , on which $\mathbf{x}$ reduces to a dual-tensor codeword $x_v = c_v + r_v$, with c_v and r_v being $C_A \otimes \mathbb{F}_2^B$ codewords and $\mathbb{F}_2^A \otimes C_B$ codewords respectively. We regroup the ‘‘column’’ vectors and ‘‘row’’ vectors and write $C_1 = \sum_{v \in V_{01}} c_v$ and $R_0 = \sum_{v \in V_{01}} r_v$. Similarly, using the partition of Q into $Q(v)$ s for $v \in V_{10}$, we define $C_0 = \sum_{v \in V_{10}} c_v$ and $R_1 = \sum_{v \in V_{10}} r_v$.

Let us denote by $\|C_i\| = \sum_{v \in V_{\bar{i}i}} \|c_v\|$ the total number of non-zero column vectors in C_A intervening in the local views of $\mathbf{x}$, for the partition over $Q(v), v \in V_{\bar{i}i}$. We write $\bar{i} = 1 - i$ and $\bar{j} = 1 - j$ to lighten notation. Similarly, we denote by $\|R_i\|$ the quantity $\|R_i\| = \sum_{v \in V_{\bar{i}i}} \|r_v\|$. Note that there are several possible decompositions of a local view x_v of $\mathbf{x}$ as $x_v = c_v + r_v$. If for every $v \in V_1$ we choose one that minimises

$\|c_v\| + \|r_v\|$, that is the total number of C_A and C_B codewords in the decomposition of x_v , we will obtain a *minimal* representation (C_0, R_0, C_1, R_1) of $\mathbf{x}$ that minimises the quantity $\|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|$. We call the latter quantity the *norm* of $\mathbf{x}$, and denote it by

$$\|\mathbf{x}\| := \|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|.$$

Since $\mathbf{x} = \sum_{v \in V_{01}} x_v = C_1 + R_0 = \sum_{v \in V_{10}} x_v = C_0 + R_1$, we have $C_0 + R_0 + C_1 + R_1 = 0$, and therefore $C_0 + R_0 = C_1 + R_1$. We make the remark that the local views of the vector $\mathbf{x}^0 := C_0 + R_0 = C_1 + R_1$ at vertices of V_0 are also dual tensor codewords in $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ (which should not be confused with the local views of $\mathbf{x}$ on vertices of V_0). In other words the vector $\mathbf{x}^0$ is a codeword of the Tanner code on the graph $\mathcal{G}_0^\square$ and the same inner code $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$. Now, if (C_0, R_0, C_1, R_1) is a minimal representation for $\mathbf{x}$, it may not necessarily be a minimal representation for $\mathbf{x}^0$. However, if we consider a local view x_v^0 of $\mathbf{x}^0$ at $v \in V_{00}$, and if its decomposition $x_v^0 = c_v + r_v$ is not minimal, where r_v and c_v are the local views at v of R_0 and C_0 , then we may replace the local decomposition by a minimal one, which will equal $x_v^0 = (c_v + t_v) + (r_v + t_v)$, where t_v is some tensor codeword in $(C_A \otimes \mathbb{F}_2^B) \cap (\mathbb{F}_2^A \otimes C_B)$. This has the effect of changing $C_1 + R_0$ and $C_0 + R_1$ to $C_1 + R_0 + t_v$ and $C_0 + R_1 + t_v$, and of reducing the sum $\|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|$. We observe that in this case $\mathbf{x}^0$ is unchanged and $\mathbf{x}$ is changed to another codeword of $\mathcal{C}_1$, which stays in the same class $\mathbf{x} + \mathcal{C}_0^\perp$, since the tensor codeword t_v is a generator. The conclusion is that we can keep proceeding in this way until no local modification is possible, at which point we will have replaced $\mathbf{x}$ by a codeword of the same class modulo $\mathcal{C}_0^\perp$, and (C_0, R_0, C_1, R_1) will be a minimal representation of both $\mathbf{x}$ and of $\mathbf{x}^0$. We have shown in particular:

Lemma 7. *If $\mathbf{x} \in \mathcal{C}_1 \setminus \mathcal{C}_0^\perp$, and $\|\mathbf{x}\|$ is the minimum norm in the coset $\mathbf{x} + \mathcal{C}_0^\perp$, then a minimal representation for $\mathbf{x}$ is also a minimum representation for $\mathbf{x}^0$.*

We will prove the following lower bound on the norm of a codeword of $\mathcal{C}_1 \setminus \mathcal{C}_0^\perp$.

Lemma 8. *For any $\mathbf{x} \in \mathcal{C}_1 \setminus \mathcal{C}_0^\perp$, we have*

$$\|\mathbf{x}\| \geq \frac{\delta^2 \kappa n}{512 \Delta^2}.$$

From Lemma 8 we easily deduce a lower bound on the quantum minimum distance.

Theorem 9. *The minimum distance of the quantum Tanner code satisfies*

$$d_{\min} \geq \frac{\delta^2 \kappa^2 n}{256 \Delta}.$$

Proof. Let $\mathbf{x}$ be a codeword of $\mathbf{x} \in \mathcal{C}_1 \setminus \mathcal{C}_0^\perp$, and let (C_0, R_0, C_1, R_1) be a minimal representation for $\mathbf{x}$. We have

$$\mathbf{x} = C_1 + R_0 = \sum_{v \in V_{01}} (c_v + r_v).$$

Since the local vectors in this sum have disjoint supports, we have

$$|\mathbf{x}| = \sum_{v \in V_{01}} |c_v + r_v| \geq \sum_{v \in V_{01}} \kappa \Delta (\|c_v\| + \|r_v\|)$$

applying robustness of the local codes and minimality of the representation. Hence $|\mathbf{x}| \geq \kappa \Delta (\|C_1\| + \|R_0\|)$, and similarly $|\mathbf{x}| \geq \kappa \Delta (\|C_0\| + \|R_1\|)$ by summing over V_{10} . Therefore,

$$|\mathbf{x}| \geq \kappa \Delta \frac{1}{2} (\|C_1\| + \|R_0\| + \|C_0\| + \|R_1\|) = \kappa \Delta \frac{1}{2} \|\mathbf{x}\|$$

which proves the lower bound for non-trivial codewords of $\mathcal{C}_1$. The same lower bound holds for non-trivial codewords of $\mathcal{C}_0$ by symmetry. $\square$

It remains to prove Lemma 8. We may suppose that $\mathbf{x}$ achieves the minimum of $\|\mathbf{x}\|$ in $\mathbf{x} + \mathcal{C}_0^\perp$, so that Lemma 7 holds. In the following, (C_0, R_0, C_1, R_1) is a minimal representation of $\mathbf{x}$.

Let us denote S_{ij} the set of vertices v of V_{ij} for which $C_j + R_i$ is non-zero on $Q(v)$. Let us also set $S_0 = S_{00} \cup S_{11}$, and $S_1 = S_{01} \cup S_{10}$.

Let us call a vertex v of V_{ij} *exceptional*, if $\|c_v\| + \|r_v\|$ is at least $\alpha \Delta$ with $\alpha = \delta^2/256$, where c_v and r_v are the restrictions to $Q(v)$ of C_j and R_i respectively. Then we write $S_{ij}^e \subset S_{ij}$ the set of exceptional vertices in S_{ij} .

Ordinary vertices of S_{ij} are defined as non-exceptional. We remark that any non-zero column C_A -vector of C_j (or any row vector of R_i) in the Q -neighbourhood of any ordinary vertex of S_{ij} has, by definition of ordinary, at most $\frac{\delta}{256} \delta \Delta$ non-zero coordinates in common with R_i , which we bound from above by $\frac{1}{2} \delta \Delta$ for the sake of readability.

The following lemma states that whenever S_{ij} is small enough, the number of exceptional vertices is a small fraction of $|S_{ij}|$, of order $O(1/\Delta^2)$, hence the terminology.

Lemma 10. *Let $i = 0, 1$. Under the hypothesis $|S_{ij}| \leq \frac{\alpha \kappa}{2} |V_{00}|$, we have*

$$|S_{ii}^e| \leq \frac{64}{\alpha^2 \kappa^2 \Delta^2} |S_0| \quad \text{and} \quad |S_{i\bar{i}}^e| \leq \frac{64}{\alpha^2 \kappa^2 \Delta^2} |S_1|.$$

Proof. We prove the upper bound for S_{00}^e , the other cases being similar. Viewing $C_0 + R_0$ as a subgraph of $\mathcal{G}_0^\square$, we have that vertices of S_{00}^e have degree at least $\kappa \Delta \alpha \Delta$ (applying robustness, which we may do by minimality of $\|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|$), and the Expander mixing Lemma in $\mathcal{G}_0^\square$ gives

$$|S_{00}^e| \alpha \kappa \Delta^2 \leq |E(S_{00}^e, S_{11})| \leq \Delta^2 \frac{|S_{00}^e| |S_{11}|}{|V_{00}|} + 4 \Delta \sqrt{|S_{00}^e| |S_{11}|}.$$

Upper bounding $|S_{11}|/|V_{00}|$ by $\kappa \alpha/2$, we obtain

$$\frac{1}{2} |S_{00}^e| \Delta^2 \alpha \kappa \leq 4 \Delta \sqrt{|S_{00}^e| |S_{11}|}$$

and finally

$$|S_{00}^e| \leq \frac{64}{\alpha^2 \kappa^2 \Delta^2} |S_{11}| \leq \frac{64}{\alpha^2 \kappa^2 \Delta^2} |S_0|. \quad \square$$

Let v be a vertex of V_{ij} and consider a column on its Q -neighbourhood. Suppose this column supports a non-zero C_A -codeword that is part of C_j . Recall that this column is shared by the Q -neighbourhood of a neighbouring vertex w in V_{ij} . Below we consider the set T of vertices of V_{ij} whose local views see at least one non-zero C_A -codeword of C_j that is shared by the local view of an *ordinary* vertex $w \in V_{ij}$.

Lemma 11. *If we have $|S_{i\bar{j}}| \leq \delta|V_{00}|/4$, then*

$$|T| \leq \frac{64}{\delta^2 \Delta} |S_{i\bar{j}}|.$$

Furthermore, exactly the same result holds if T is defined as the subset of vertices of $S_{i\bar{j}}$ on whose Q -neighbourhood $R_{\bar{i}}$ displays a non-zero row codeword of C_B that is shared by the local view of an ordinary vertex $w \in S_{i\bar{j}}$.

Proof. We deal with the case when $i = j = 0$ and T is defined as the set of vertices of V_{00} whose Q -neighbourhoods share a non-zero column vector with an ordinary vertex of V_{10} . The other cases will hold by symmetry. If we keep only the two sets of vertices V_{00} and V_{01} , then every square of the square complex becomes incident to two vertices (instead of four), and we obtain a multigraph. In this multigraph, every row of a local view appears in two Q -neighbourhoods, one of a vertex $v \in V_{00}$, and the other in a neighbouring vertex of V_{01} . If we collapse this row to a single “square” (which has now become an edge) by identifying them, the bipartite graph over $V_{00} \cup V_{01}$ that we obtain in this way is exactly the double cover of the Cayley graph $\text{Cay}(G, A)$. Now the codeword $\mathbf{x}$ induces a subgraph of this graph, which we obtain by putting an edge in the subgraph whenever the row view it originates from is non-zero in $\mathbf{x}$.

By construction, every vertex of T in this subgraph has degree at least $\delta\Delta/2$ (actually degree at least $(1 - \frac{\delta}{256})\delta\Delta$, as discussed above), and its edges fall into S_{01} . Hence,

$$|T| \frac{\delta\Delta}{2} \leq |E(S_{01}, T)|.$$

Applying the expander mixing Lemma in the double cover of $\text{Cay}(G, A)$ we obtain,

$$|E(S_{01}, T)| \leq \Delta \frac{|S_{01}||T|}{|V_{01}|} + 2\sqrt{\Delta} \sqrt{|S_{01}||T|}.$$

Finally, applying the hypothesis $|S_{01}| \leq \delta|V_{01}|/4$ we get

$$|T| \frac{\delta\Delta}{4} \leq 2\sqrt{\Delta} \sqrt{|S_{01}||T|}$$

and the result follows. We remark that when T is defined in $V_{i\bar{i}}$ as opposed to V_{ii} , it is the codeword $\mathbf{x}^0 = R_0 + C_0$ that is considered rather than $\mathbf{x}$. $\square$

Now, Lemma 10 will tell us that if $\|\mathbf{x}\|$ is too small, there can only be very few exceptional vertices. But what Lemma 11 then tells us, is that the non-zero column codewords in the neighbourhoods of ordinary vertices must cluster in a limited number of Q -neighbourhoods for vertices of a neighbouring type, yielding too many exceptional vertices and contradicting Lemma 10. We now make this argument formal.

Proof of Lemma 8. Suppose $\mathbf{x} \in \mathcal{C}_1 \setminus \mathcal{C}_0^\perp$ satisfies $\|\mathbf{x}\| < \frac{\kappa\delta^2 n}{512\Delta^2}$. We may suppose that $\|\mathbf{x}\|$ achieves its minimum value inside the coset $\mathbf{x} + \mathcal{C}_0^\perp$, and that (C_0, R_0, C_1, R_1) is a minimal representation for $\mathbf{x}$. Since we have $|S_{ij}| \leq \|C_j\| + \|R_i\| \leq \|C_0\| + \|R_0\| + \|C_1\| + \|R_1\| = \|\mathbf{x}\|$, we have that $|S_{ij}| \leq \frac{\kappa\delta^2}{512}|V_{00}| = \frac{1}{2}\kappa\alpha|V_{00}|$. Therefore Lemma 10 holds, and so does Lemma 11 since clearly $\frac{1}{2}\kappa\alpha \leq \delta/4$.

Without loss of generality let us suppose $|S_1| \geq |S_0|$ (otherwise invert their roles in the argument below) and $|S_{10}| \geq |S_{01}|$ (otherwise invert their roles).

Because there are so few exceptional vertices in S_{10} (by Lemma 10), the number of ordinary vertices of S_{10} is almost equal to $|S_{10}|$ namely it is at least $a|S_{10}|$ for any constant $a < 1$ and Δ large enough.

We obtain that either the number of ordinary rows or the number of ordinary columns in (the Q -neighbourhoods of vertices of) S_{10} is at least $a|S_{10}|/2$. Suppose without loss of generality² that the number of ordinary columns is at least $a|S_{10}|/2$. Applying Lemma 11, they must cluster among the Q -neighbourhoods of a set $T \subset S_{00}$ of size at most $\frac{64}{\delta^2\Delta}|S_{01}| \leq \frac{64}{\delta^2\Delta}|S_{10}|$ (since we have supposed $|S_{01}| \leq |S_{10}|$). Therefore, the average number of non-zero columns of C_0 on the Q -neighbourhoods of the vertices of this set T is at least $a|S_{10}|/2 \left(\frac{64}{\delta^2\Delta}|S_{10}|\right)^{-1} = a\frac{\delta^2\Delta}{128}$, which is close to twice the minimum norm of a local view of $R_i + C_j$ for an exceptional vertex. This shows that a constant proportion of vertices of T must be exceptional vertices of S_{00} . Now since the Q -neighbourhood of a vertex can host at most Δ columns, we also have $|T| \geq \frac{1}{\Delta} \frac{a}{2} |S_{10}| \geq \frac{a}{4\Delta} |S_1|$ (since $|S_{10}| \geq |S_{01}|$), and since $|S_1| \geq |S_0|$, we have $|T| \geq \frac{a}{4\Delta} |S_0|$. Therefore, for large enough Δ , we have a contradiction with Lemma 10 which limits the number of exceptional vertices of S_{00} to not more than $O(1/\Delta^2)|S_0|$. $\square$

4 Description of the decoders

In this section, we present in detail the various decoders: the general decoding procedure including the computation of the mismatch and the post-processing is described in Algorithm 1. We give two procedures for finding a mismatch decomposition of the form $Z = \hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1$, a sequential procedure described in Algorithm 2 and a parallel one described in Algorithm 3.

Both the sequential decoder and the parallel decoder follow the blueprint of Algorithm 1. They use a *preprocessing phase* which consists of computing a mismatch vector Z . Specifically, on the Q -neighbourhood of every vertex of V_0 (if one is trying to correct a σ_z -error), or of V_1 (if one is trying to correct a σ_x -error), the decoder computes a local estimation ε_v of the original error vector $\mathbf{e}$, and then sums all these ε_v 's to make up the mismatch vector Z . Note that this common preprocessing phase can be parallelised straightforwardly when needed.

The core of the decoding algorithm is then to uncover a decomposition of the mismatch vector of the form $Z = \hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1$, where $\hat{C}_i$ is a sum of local (column) codewords of C_A whose coordinates are indexed by a column of a Q -neighbourhood $Q(v)$

²If it is the number of ordinary rows that exceeds $a|S_{10}|/2$, then T is defined inside S_{11} instead of S_{00} and the rest of the argument is unchanged.

for $v \in V_{ii}$, and similarly $\hat{R}_i$ is a sum of local row codewords of C_B whose coordinates are indexed by a row in some $Q(v)$, $v \in V_{ii}$. Note that the individual local column codewords of C_i also appear in the Q -neighbourhoods of vertices of V_{ii} , and the row codewords of R_i appear likewise in Q -neighbourhoods of $v \in V_{ii}$. The mismatch decomposition procedures differ significantly for the sequential decoder and the parallel decoder.

Finally, both decoders use a common postprocessing phase once they have decomposed the mismatch vector Z . If one is decoding a σ_z -error, the decoders output their estimation of the error which is computed as

$$\hat{e} = \sum_{v \in V_{00}} \varepsilon_v + \hat{C}_0 + \hat{R}_0.$$

We remark that we also have

$$\hat{e} = \sum_{v \in V_{11}} \varepsilon_v + \hat{C}_1 + \hat{R}_1$$

since the sum of those two quantities is zero by construction and by definition of the mismatch. Similarly, if one is decoding a σ_x -error, the decoders compute $\hat{e} = \sum_{v \in V_{10}} \varepsilon_v + \hat{C}_0 + \hat{R}_1 = \sum_{v \in V_{01}} \varepsilon_v + \hat{C}_1 + \hat{R}_0$. Since these summations can be done locally on the components of the partition of Q into Q -neighbourhoods of the relevant V_{ij} , the postprocessing is achieved naturally by means of a parallel computation.

It remains to describe the mismatch decomposition procedures. The sequential mismatch decomposition procedure is given in Algorithm 2. The sequential procedure is rather natural: it consists of initiating a variable $\hat{Z}$ at Z , and iteratively looking for a vertex $v \in V$ and a non-zero dual tensor codeword x_v supported by $Q(v)$, such that adding x_v to $\hat{Z}$ decreases its weight by a sufficient amount. The decoder then decomposes x_v as $x_v = c_v + r_v$ with $\|c_v\| + \|r_v\|$ minimum, and increments $\hat{C}_j$ by c_v and $\hat{R}_i$ by r_v , where i, j are such that $v \in V_{ij}$. Iterations continue until $\hat{Z} = 0$.

The parallel decomposition procedure uses the fact that the local views of vertices of a given V_{ij} have disjoint supports, and one may therefore update these local views in parallel without ever risking conflicting instructions. It follows that one round of parallel decoding will consist of four consecutive parallel substeps, one for every set V_{ij} . During a parallel substep, every vertex v of the relevant V_{ij} will do the following: if it finds that there exists a non-zero dual tensor codeword $x_v = c_v + r_v$ (with, as before, $\|c_v\| + \|r_v\|$ minimum among possible decompositions of x_v), such that $|\hat{Z}| - |\hat{Z} + x_v| \geq |x_v|/2$, then, if there are several choices for x_v it chooses one that maximises $|x_v|$, and it updates (the local views of) $\hat{C}_j$ by c_v and $\hat{R}_i$ by r_v , as in the sequential case, so that $\hat{Z}$ is also updated to $\hat{Z} + x_v$. Every vertex v of V_{ij} does this procedure in parallel, after which the decoder repeats the procedure with another set $V_{i'j'}$. When the four sets of vertices $V_{00}, V_{01}, V_{10}, V_{11}$ have been dealt with as described, the round of parallel decoding is complete.

The whole procedure is summarised in Algorithm 3. Note that, for the sake of readability, the algorithm is described with a failure detecting mechanism that is not

truly part of the parallel procedure. The fully parallel decoder will simply apply a predetermined number of decoding rounds.

Algorithm 1: General decoding procedure

input : a pair of syndromes (s^x, s^z)
output: Either **failure** or a vector $(\hat{e}^x, \hat{e}^z) \in \mathbb{F}_2^{2Q}$ with syndrome (s^x, s^z) .

/ Beginning of preprocessing phase. */*

- 1 Set $Z^x = 0, Z^z = 0, \hat{e}^x = 0, \hat{e}^z = 0$.
- 2 **for** $v \in V_1$ **do**
- 3 Set ε_v to be the error of minimum weight with local syndrome coinciding with s^x .
- 4 $Z^x \leftarrow Z^x + \varepsilon_v$.
- 5 **for** $v \in V_0$ **do**
- 6 Set ε_v to be the error of minimum weight with local syndrome coinciding with s^z .
- 7 $Z^z \leftarrow Z^z + \varepsilon_v$.

/ End of preprocessing phase. */*

- 8 Compute $(\hat{C}_0^x, \hat{R}_0^x, \hat{C}_1^x, \hat{R}_1^x) = \text{mismatch-decomposition}(Z^x)$. */* Calls Algorithm 2 or 3 */*
- 9 Compute $(\hat{C}_0^z, \hat{R}_0^z, \hat{C}_1^z, \hat{R}_1^z) = \text{mismatch-decomposition}(Z^z)$. */* Calls Algorithm 2 or 3 */*

/ Beginning of postprocessing phase. */*

- 10 **for** $v \in V_{10}$ **do**
- 11 Set $e_v^x = \varepsilon_v + c_v + r_v$ where c_v and r_v are the local values of $\hat{C}_0^x$ and $\hat{R}_1^x$.
- 12 **for** $v \in V_{00}$ **do**
- 13 Set $e_v^z = \varepsilon_v + c_v + r_v$ where c_v and r_v are the local values of $\hat{C}_0^z$ and $\hat{R}_0^z$.
- 14 **return** $\hat{e}^x, \hat{e}^z$ corresponding to the decompositions $(e_v^x)_{v \in V_{10}}, (e_v^z)_{v \in V_{00}}$.

/ End of postprocessing phase. */*

5 Sequential decoding

Both the sequential decoder and the parallel decoder look for a decomposition of the mismatch vector Z as a sum, over a subset of vertices v , of local dual tensor codewords $c_v + r_v$ supported by Q -neighbourhoods $Q(v)$. The mismatch decomposition procedure is the core of the decoding algorithm, both in the sequential and in the parallel case.

The sequential mismatch decomposition procedure is parameterised by a constant $\varepsilon \in (0, 1)$ and proceeds in the natural way: it looks for a vertex $v \in V$ together with a local dual tensor codeword $x_v = c_v + r_v$ (with $c_v \in C_A \otimes \mathbb{F}_2^B$, $r_v \in \mathbb{F}_2^A \otimes C_B$) such that flipping x_v decreases the Hamming weight of the mismatch by at least $(1 - \varepsilon)|x_v|$, in other words, such that $|\hat{Z} + x_v| \leq |\hat{Z}| - (1 - \varepsilon)|x_v|$, where $\hat{Z}$ is the current value of the

Algorithm 2: Sequential mismatch decomposition procedure (with parameter ε)

input : a mismatch $Z \in \mathbb{F}_2^Q$
output: Either **failure** or $(\hat{C}_0, \hat{R}_0, \hat{C}_1, \hat{R}_1)$ such that $\hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1 = Z$.

- 1 Set $\hat{C}_0 = 0, \hat{R}_0 = 0, \hat{C}_1 = 0, \hat{R}_1 = 0$ and $\hat{Z} = Z$.
- 2 **while** $\hat{Z} \neq 0$ **do**
- 3 **if** $\exists v \in V_{ij}, c_v + r_v$ such that $|\hat{Z}| - |\hat{Z} + c_v + r_v| \geq (1 - \varepsilon)|c_v + r_v|$, choosing c_v, r_v such that $\|c_v\| + \|r_v\|$ is minimum among c_v, r_v such that $x_v = c_v + r_v$,
 /* **with** $c_v \in C_A \otimes \mathbb{F}_2^B, r_v \in \mathbb{F}_2^A \otimes C_B$ **for** σ_x -**type errors**, **and**
 $c_v \in C_A^\perp \otimes \mathbb{F}_2^B, r_v \in \mathbb{F}_2^A \otimes C_B^\perp$ **for** σ_z -**type errors**. */
 then
 5 $\hat{C}_j \leftarrow \hat{C}_j + c_v$
 6 $\hat{R}_i \leftarrow \hat{R}_i + r_v$
 7 $\hat{Z} \leftarrow \hat{Z} + c_v + r_v$
 8 **else return Failure.**
- 9 **return** $(\hat{C}_0, \hat{R}_0, \hat{C}_1, \hat{R}_1)$.

mismatch, initiated at the original mismatch vector Z . It then proceeds by updating the current mismatch value to $\hat{Z} = \hat{Z} + x_v$, and continues in this way until $\hat{Z} = 0$, at which point it outputs the sum of the updates x_v , which equals Z . Theorem 12 below states that the required local codeword x_v always exists for a given mismatch $\hat{Z}$, provided a minimum decomposition of $\hat{Z}$ has sufficiently few active vertices. We now explain what this means.

We may reorganise a decomposition $\hat{Z} = \sum_{v \in V} c_v + r_v$ of $\hat{Z}$ into a sum of local dual tensor codewords as

$$\hat{Z} = \hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1, \quad (4)$$

where $\hat{C}_i = \sum_{v \in V_{ii} \cup V_{ji}} c_v$ and $\hat{R}_j = \sum_{v \in V_{jj} \cup V_{ij}} r_v$. Now we may decompose $\hat{C}_0$ as a sum of c_v 's, for v restricted to V_{00} (as opposed to v ranging over $V_{00} \cup V_{10}$), in which case this decomposition is unique, and we shall denote $\|\hat{C}_0\|$ the sum of all corresponding $\|c_v\|$ s, where we recall that $\|c_v\|$ denotes the number of individual column vectors that make up the local codeword in $C_A \otimes \mathbb{F}_2^B$. Note that we obtain the same value $\|\hat{C}_0\|$, if we decompose $\hat{C}_0$ as a sum of c_v 's, for v ranging in V_{10} . The quantities $\|\hat{C}_1\|, \|\hat{R}_0\|, \|\hat{R}_1\|$ are defined similarly. We now define the *norm* $\|\hat{Z}\|$ of $\hat{Z}$ to be the minimum value of

$$\|\hat{C}_0\| + \|\hat{R}_0\| + \|\hat{C}_1\| + \|\hat{R}_1\| \quad (5)$$

over all possible decompositions (4) of $\hat{Z}$. We shall say that a decomposition (4) of $\hat{Z}$ is minimum if (5) equals $\|\hat{Z}\|$. Finally, we shall say that for a decomposition (4), the vertex v in V_{ij} is *active*, if $\hat{C}_j + \hat{R}_i$ is non-zero on $Q(v)$. We may now state Theorem 12, which is the core technical result for the decoder analysis:

Algorithm 3: Parallel mismatch decomposition procedure

input : a mismatch $Z \in \mathbb{F}_2^Q$
output: Either **failure** or $(\hat{C}_0, \hat{R}_0, \hat{C}_1, \hat{R}_1)$ such that $\hat{C}_0 + \hat{R}_0 + \hat{C}_1 + \hat{R}_1 = Z$.
1 Set $\hat{C}_0 = 0, \hat{R}_0 = 0, \hat{C}_1 = 0, \hat{R}_1 = 0$ and $\hat{Z} = Z$.
2 **while** $\hat{Z} \neq 0$ **do**
3 $Temp = \hat{Z}$.
4 **for** $(i, j) \in \{(00), (01), (10), (11)\}$ **do**
5 **for** $v \in V_{ij}$, **do**
6 **if** $\exists x_v = c_v + r_v \neq 0$ such that $|\hat{Z}| - |\hat{Z} + x_v| \geq |x_v|/2$ **then**
7 update $\hat{C}_j \leftarrow \hat{C}_j + c_v, \hat{R}_i \leftarrow \hat{R}_i + r_v, \hat{Z} \leftarrow \hat{Z} + x_v$, choosing $|x_v|$
 maximum among all possible choices, and c_v, r_v being chosen such
 that $\|c_v\| + \|r_v\|$ is minimum among c_v, r_v such that $x_v = c_v + r_v$.
8 **if** $\hat{Z} = Temp$ **return Failure**.
9 **return** $(\hat{C}_0, \hat{R}_0, \hat{C}_1, \hat{R}_1)$.

Theorem 12. Fix $\varepsilon \in (0, 1)$. If, for $i, j \in \{0, 1\}$, the sets of active vertices $S_{ij} \subset V_{ij}$ for a minimum decomposition of a mismatch vector Z satisfy

$$|S_{ij}| \leq \frac{1}{2^{12}} \delta^2 \varepsilon^3 \kappa |V_{00}|$$

then there exists some vertex v and some codeword x_v of the dual tensor code such that

$$|Z| - |Z + x_v| \geq (1 - \varepsilon)|x_v|.$$

From Theorem 12, will follow Theorem 13, which states that if the weight $|e|$ of the original error e is sufficiently small, then the number of active vertices of a minimum decomposition of $\hat{Z}$ will satisfy the hypothesis of Theorem 12 throughout the sequential decoding procedure, meaning that the sequential mismatch decomposition procedure will always converge.

Theorem 13. Fix $\varepsilon \in (0, 1)$. If the Hamming weight $|e|$ is less than

$$\frac{1}{2^{11}} \min\left(\frac{\varepsilon^3}{16}, \kappa\right) (1 - \varepsilon) \delta^2 \kappa^2 \frac{n}{\Delta},$$

then the sequential decoder with parameter ε returns a valid correction $\hat{e}$, namely a correction equivalent to e (differing from e by an element of the stabilizer group).

5.1 Proof of Theorem 12

The proof strategy follows the same footsteps as the proof of the minimum distance. When studying the minimum distance, we analysed the set $C_j + R_i$, which coincided

with $C_{\bar{j}} + R_{\bar{i}}$. For the decoding, we start by identifying the mismatch Z associated with the error and take a minimum decomposition $C_0 + R_0 + C_1 + R_1$. The relevant set is now the support of $(C_j + R_i) \cap (C_{\bar{j}} \cap R_{\bar{i}})$, which is smaller than the set considered when analysing the minimum distance. However, under the assumption that the sequential decoder is stalled, this set cannot be too small, and essentially the same techniques as before will allow us to arrive at a contradiction.

5.1.1 A stalled sequential decoder, Exceptional vertices, ordinary rows and columns

We consider a σ_x -type error e and define its associated mismatch Z . We work with a minimal decomposition of Z :

$$Z = C_0 + R_0 + C_1 + R_1,$$

meaning that the quantity $\|C_0\| + \|R_0\| + \|C_1\| + \|R_1\|$ is minimal. To each vertex $v \in V$, this decomposition associates codewords $c_v \in C_A \otimes \mathbb{F}_2^B$ and $r_v \in \mathbb{F}_2^A \otimes C_B$. We say that a vertex $v \in V_{ij}$ is an *active vertex* if $c_v + r_v \neq 0$, *i.e.* if $C_j + R_i$ is non-zero on $Q(v)$, and we denote by S_{ij} the sets of active vertices in V_{ij} .

The sequential decoder with parameter ε searches for some vertex $v \in V$ and a dual tensor codeword $x_v \in (C_A^\perp \otimes C_B^\perp)^\perp = C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ such that flipping x_v decrease the Hamming weight of Z by at least $(1 - \varepsilon)|x_v|$. To prove Theorem 12 we will assume there is no such vertex and work towards a contradiction.

We will follow the blueprint of Section 3, and define exceptional and ordinary vertices of S_{ij} as before, namely a vertex $v \in S_{ij}$ is said to be exceptional, if the local dual tensor codeword $x_v = c_v + r_v$, equal to the restriction of $C_j + R_i$ to $Q(v)$, satisfies $\|c_v\| + \|r_v\| \geq \alpha\Delta$. Here we will take $\alpha = \frac{1}{2^{10}}\delta^2\varepsilon^2$. The set of exceptional vertices of S_{ij} is denoted by S_{ij}^e and non-exceptional vertices are called *ordinary*. Let us furthermore call an ordinary column (row) of $v \in S_{ij}$ a column (row) of the Q -neighbourhood $Q(v)$ on which C_j (R_i) is non-zero, and for a vertex v that is ordinary. Note that when talking about ordinary columns (or rows) it is important to specify for which vertex v , since this column appears in two different local views for two different vertices, and may be ordinary for one vertex and not for the other.

Lemma 14. *Assume that the sequential decoder of parameter ε is stalled. For all $v \in S_{ij}$, and all dual tensor codewords x_v , components of $C_j + R_i$, we have*

$$|x_v \cap (C_{\bar{j}} + R_{\bar{i}})| \geq \frac{\varepsilon}{2}|x_v|.$$

Furthermore, let y_v be the subvector of C_j supported by some ordinary column for $v \in S_{ij}$. Then

$$|y_v \cap (C_{\bar{j}} + R_{\bar{i}})| \geq \frac{\varepsilon}{4}|y_v|.$$

Proof. Note that for any two binary vectors x, z , identifying them with their supports we have that $|z| - |z + x| \leq (1 - \varepsilon)|x|$ is equivalent to $2|z \cap x| \leq (2 - \varepsilon)|x|$, since

$|z + x| = |z| - 2|z \cap x| + |x|$. Since the decoder is stalled, we have

$$|Z| - |Z + x_v| \leq (1 - \varepsilon)|x_v|$$

which therefore gives

$$2|Z \cap x_v| \leq (2 - \varepsilon)|x_v|.$$

Note that $x_v \subset C_j + R_i$ and therefore $Z \cap x_v = x_v + ((C_{\bar{j}} + R_{\bar{i}}) \cap x_v)$ and $|Z \cap x_v| = |x_v| - |(C_{\bar{j}} + R_{\bar{i}}) \cap x_v|$. Combining this with the previous inequality proves the first claim of the Lemma.

To prove the second claim of the Lemma we argue that, since y_v is an ordinary column vector, that there is some $y' \subset y_v$ such that $|y'| \leq \alpha\Delta$ and $y_v + y' \subset C_j + R_i$. Specifically, y' is supported by the intersection of y_v and R_i . From our choice of α we clearly have $\alpha\Delta \leq \frac{\varepsilon}{2}\delta\Delta$, so that $|y'| \leq \frac{\varepsilon}{4}|y_v|$. We have

$$2|Z \cap y_v| \leq (2 - \varepsilon)|y_v|,$$

otherwise we could decode at vertex v by flipping y_v . Since $y_v + y' \subset y_v$, we can write $2|Z \cap (y_v + y')| \leq (2 - \varepsilon)|y_v|$, and since $y_v + y' \subset C_j + R_i$, we have $|Z \cap (y_v + y')| = |y_v + y'| - |(C_{\bar{j}} + R_{\bar{i}}) \cap (y_v + y')|$, whence

$$2|y_v| - 2|y'| - (2 - \varepsilon)|y_v| \leq 2|(C_{\bar{j}} + R_{\bar{i}}) \cap (y_v + y')| \leq 2|(C_{\bar{j}} + R_{\bar{i}}) \cap y_v|$$

which proves the claim, since $|y'| \leq \frac{\varepsilon}{4}|y_v|$. $\square$

Recall that minimality of the representation (C_0, R_0, C_1, R_1) and the robustness property (Lemma 6) imply that

$$|c_v + r_v| \geq \kappa(\|c_v\| + \|r_v\|)\Delta$$

whenever $c_v + r_v$ is the local representation of $C_j + R_i$ at $v \in S_{ij}$. In particular, an exceptional vertex is such that $|c_v + r_v| \geq \alpha\kappa\Delta^2$. As before we denote by S_{ij}^e the set of exceptional vertices in S_{ij} .

5.1.2 Exceptional vertices are rare

Lemma 15. *Let $i = 0, 1$. Under the hypothesis $|S_{ij}| \leq \frac{\alpha\kappa\varepsilon}{4}|V_{00}|$, we have*

$$|S_{ii}^e| \leq \frac{2^8}{\alpha^2\kappa^2\varepsilon^2} \frac{1}{\Delta^2} |S_0| \quad \text{and} \quad |S_{ii}^e| \leq \frac{2^8}{\alpha^2\kappa^2\varepsilon^2} \frac{1}{\Delta^2} |S_1|.$$

Proof. The proof follows closely that of Lemma 10. We prove the upper bound for S_{00}^e , the other cases being similar. Viewing $(C_0 + R_0) \cap (C_1 + R_1)$ as a subgraph of $\mathcal{G}_0^\square$, we have that vertices of S_{00}^e have degree at least $\kappa\alpha\varepsilon\Delta^2/2$, by Lemma 14 and robustness.

By the Expander mixing Lemma in $\mathcal{G}_0^\square$, we therefore have

$$\frac{1}{2}\alpha\kappa\varepsilon\Delta^2|S_{00}^e| \leq |E(S_{00}^e, S_{11})| \leq \Delta^2 \frac{|S_{00}^e||S_{11}|}{|V_{00}|} + 4\Delta\sqrt{|S_{00}^e||S_{11}|}.$$

Writing $|S_{11}|/|V_{00}| \leq \frac{\alpha\kappa\varepsilon}{4}$ by assumption, we get

$$\begin{aligned} \frac{1}{4}\alpha\kappa\varepsilon\Delta^2|S_{00}^e|^{1/2} &\leq 4\Delta|S_{11}|^{1/2} \\ |S_{00}^e| &\leq \frac{2^8}{\alpha^2\kappa^2\varepsilon^2}\frac{1}{\Delta^2}|S_{11}|. \end{aligned}$$

which proves the claimed result, since $|S_{11}| \leq |S_0|$. $\square$

5.1.3 Ordinary columns and rows cluster

Lemma 16. *Let T be the set of vertices v of V_{ij} whose Q -neighbourhood $Q(v)$ contains a column that is a ordinary column for the neighbouring vertex $w \in V_{\bar{i}\bar{j}}$ whose local view shares this very column. If we have $|S_{i\bar{j}}| \leq \delta\varepsilon|V_{i\bar{j}}|/8$, then*

$$|T| \leq \frac{256}{\delta^2\varepsilon^2\Delta}|S_{i\bar{j}}|.$$

Exactly the same result holds for the set $T \subset S_{\bar{i}\bar{j}}$ of vertices on whose Q -neighbourhood $R_{\bar{i}}$ displays a non-zero row codeword of C_B that is shared by the local view of an ordinary vertex $w \in S_{\bar{i}\bar{j}}$.

Proof. We follow closely the proof of Lemma 11. We deal with the case when $i = j = 0$ and T is defined as the set of vertices of V_{00} whose Q -neighbourhoods share a non-zero column vector with an ordinary vertex of V_{10} . The other cases will hold by symmetry.

Again we keep only the two sets of vertices V_{00} and V_{01} , and collapse rows of local views to single edges, and we look at the graph induced by the squares of $(C_1 + R_0) \cap (C_0 + R_1)$ on the vertex set $T \cup S_{01}$. What the second claim of Lemma 14 tells us, is that the degree of any vertex of T in this subgraph is at least $\delta\Delta\frac{\varepsilon}{4}$.

As in the proof of Lemma 11, we apply the expander mixing Lemma in the double cover of $\text{Cay}(G, A)$ to obtain,

$$|T|\frac{\delta\varepsilon\Delta}{4} \leq |E(S_{01}, T)| \leq \Delta\frac{|S_{01}||T|}{|V_{01}|} + 2\sqrt{\Delta}\sqrt{|S_{01}||T|}.$$

Applying the hypothesis $|S_{01}| \leq \delta\varepsilon|V_{01}|/8$ we get

$$|T|\frac{\delta\varepsilon\Delta}{8} \leq 2\sqrt{\Delta}\sqrt{|S_{01}||T|}$$

and the result follows. $\square$

5.1.4 Obtaining a contradiction

Proof of Theorem 12. The hypothesis of the theorem translates into $|S_{ij}| \leq \frac{\alpha\kappa\varepsilon}{4}|V_{00}|$, since we defined $\alpha = \delta^2\varepsilon^2/2^{10}$. Therefore Lemma 15 holds, and so does Lemma 16, since clearly $\frac{1}{4}\alpha\varepsilon\kappa \leq \delta\varepsilon/8$.

Without loss of generality, let us suppose $|S_1| \geq |S_0|$ (otherwise invert their roles in the argument below) and $|S_{10}| \geq |S_{01}|$ (otherwise invert their roles).

Because there are so few exceptional vertices in S_{10} (by Lemma 15), the number of ordinary vertices of S_{10} is almost equal to $|S_{10}|$ namely it is at least $a|S_{10}|$ for any constant $a < 1$ and Δ large enough.

We obtain that either the number of ordinary rows or the number of ordinary columns in (the Q -neighbourhoods of vertices of) S_{10} is at least $a|S_{10}|/2$. Suppose without loss of generality that the number of ordinary columns is at least $a|S_{10}|/2$. Applying Lemma 16, they must cluster among the Q -neighbourhoods of a set $T \subset S_{00}$ of size at most $\frac{256}{\delta^2 \varepsilon^2 \Delta} |S_{01}| \leq \frac{256}{\delta^2 \varepsilon^2 \Delta} |S_{10}|$ (since we have supposed $|S_{01}| \leq |S_{10}|$). Therefore, the average number of non-zero columns of C_0 on the Q -neighbourhoods of the vertices of this set T is at least $a \frac{\delta^2 \varepsilon^2 \Delta}{512}$, which is close to twice the minimum norm $\alpha \Delta$ of a local view of $R_i + C_j$ for an exceptional vertex.

Therefore, a constant proportion of vertices of T must be exceptional vertices of S_{00} . Now since the Q -neighbourhood of a vertex can host at most Δ columns, we also have $|T| \geq \frac{1}{\Delta} \frac{a}{2} |S_{10}| \geq \frac{a}{4\Delta} |S_1|$ (since $|S_{10}| \geq |S_{01}|$), and since $|S_1| \geq |S_0|$ we have $|T| \geq \frac{a}{4\Delta} |S_0|$. Therefore, for large enough Δ , we have a contradiction with Lemma 15 which limits the number of exceptional vertices of S_{00} to not more than $O(1/\Delta^2)|S_0|$. $\square$

5.2 Proof of Theorem 13

Without loss of generality we may suppose the error $\mathbf{e}$ to be a σ_x -type error.

We need to guarantee that the upper bound on $|S_{ij}|$ required by Theorem 12 is satisfied throughout the decoding procedure, until we reach a zero mismatch $\hat{Z}$. Recall that S_{ij} is the set of active vertices of V_{ij} in a minimum decomposition of $\hat{Z}$. We will argue that $|S_{ij}| \leq \|\hat{Z}\|$, so we track the evolution of $\|\hat{Z}\|$ during the mismatch decomposition procedure.

Lemma 17. *If Z is the mismatch vector that the preprocessing phase associates to an error vector $\mathbf{e}$, then we have*

$$|Z| \leq 4|\mathbf{e}| \quad (6)$$

and

$$\|Z\| \leq \frac{4}{\kappa \Delta} |\mathbf{e}|.$$

Proof. During the preprocessing phase, we have that $Z = \sum_{v \in V_1} \varepsilon_v = \sum_{v \in V_1} x_v$, where $e_v = \varepsilon_v + x_v$ is the projection of the error vector $\mathbf{e}$ on $Q(v)$ and where $x_v = c_v + r_v$ is the dual tensor codeword that is the difference between e_v and the decoder's initial evaluation ε_v of the error. The minimality of $|\varepsilon_v| = |e_v + x_v|$ implies that $|\varepsilon_v| \leq |e_v|$ and therefore that $|x_v| \leq 2|e_v|$. This gives

$$|Z| \leq \sum_{v \in V_1} |x_v| \leq 2 \sum_{v \in V_{10}} |e_v| = 4|\mathbf{e}|$$

which proves the first point. Writing $\|x_v\| := \|c_v\| + \|r_v\|$ and applying robustness to x_v , we also have $\kappa\Delta\|x_v\| \leq |x_v| \leq 2|e_v|$, whence

$$\|x_v\| \leq \frac{2}{\kappa\Delta}|e_v|.$$

Summing over all vertices of V_1 , we obtain

$$\|Z\| \leq \sum_{v \in V_1} \|x_v\| \leq \sum_{v \in V_{01}} \|x_v\| + \sum_{v \in V_{10}} \|x_v\| \leq \frac{2}{\kappa\Delta} \sum_{v \in V_{01}} |e_v| + \frac{2}{\kappa\Delta} \sum_{v \in V_{10}} |e_v| = \frac{4}{\kappa\Delta}|e|$$

since $|e| = \sum_{v \in V_{01}} |e_v| + \sum_{v \in V_{10}} |e_v|$. $\square$

Assume that after the m^{th} round of sequential decoding, the decoder has flipped successively the local dual tensor codewords $x_1, x_2, \dots, x_m$. We must have in particular

$$(1 - \varepsilon)(|x_1| + |x_2| + \dots + |x_m|) \leq |Z|.$$

Now, every time we modify $\hat{Z}$ by adding some x_i to it, we have that $\|\hat{Z}\|$ is increased by at most $\|x_i\|$. Robustness implies that $\|x_i\| \leq \frac{1}{\kappa\Delta}|x_i|$ and we may therefore write

$$\|x_1\| + \|x_2\| + \dots + \|x_m\| \leq \frac{1}{\kappa\Delta(1 - \varepsilon)}|Z|. \quad (7)$$

Applying Lemma 17 we therefore obtain that

$$\|\hat{Z}\| \leq \|Z\| + \|x_1\| + \|x_2\| + \dots + \|x_m\| \leq \frac{4}{\kappa\Delta} \left(1 + \frac{1}{1 - \varepsilon}\right) |e| \leq \frac{8}{(1 - \varepsilon)\kappa\Delta} |e|$$

since $1 + 1/(1 - \varepsilon) \leq 2/(1 - \varepsilon)$. Therefore, if we impose the condition

$$|e| \leq \frac{(1 - \varepsilon)\kappa\Delta}{8} \frac{1}{2^{12}} \delta^2 \varepsilon^3 \kappa |V_{00}| = \frac{1}{2^{11}} \frac{\varepsilon^3}{16} \kappa^2 \delta^2 (1 - \varepsilon) \frac{n}{\Delta}$$

we obtain that $|S_{ij}| \leq \|\hat{Z}\|$ must always be bounded from above by $\frac{1}{2^{12}} \delta^2 \varepsilon^3 \kappa |V_{00}|$ throughout the decoding procedure, so that Theorem 12 always applies and decoding can always continue.

We also need to check that the output $\hat{e}$ of the decoder is correct. This will be the case provided that $|e + \hat{e}| < d_{\min}(Q)$. Recall that

$$|e + \hat{e}| = \left| \sum_{v \in V_{10}} (e_v + \varepsilon_v) + \hat{C}_0 + \hat{R}_1 \right| \quad (8)$$

and that $|\sum_{v \in V_{10}} (e_v + \varepsilon_v)| = |\sum_{v \in V_{10}} x_v| \leq 2|e|$. To evaluate $|\hat{C}_0 + \hat{R}_1|$, we make the remark that every local dual tensor codeword x_i that is used by an iteration of the sequential decoder contributes at most $\|x_i\|$ non-zero row and column vectors to $\hat{C}_0 + \hat{R}_1$. Therefore,

$$|\hat{C}_0 + \hat{R}_1| \leq \Delta \sum_i \|x_i\|$$

where the sum runs over all local increments used by the decoder. Applying (7), we obtain

$$|\hat{C}_0 + \hat{R}_1| \leq \frac{1}{\kappa(1-\varepsilon)}|Z|.$$

Writing $|Z| \leq 4|\mathbf{e}|$ (from (6)), we get

$$|\hat{C}_0 + \hat{R}_1| \leq \frac{4}{\kappa(1-\varepsilon)}|\mathbf{e}|.$$

Since $\kappa < 1$, we may write $2|\mathbf{e}| \leq \frac{4}{\kappa(1-\varepsilon)}|\mathbf{e}|$, and (8) now gives us

$$|\mathbf{e} + \hat{\mathbf{e}}| \leq \frac{4}{\kappa(1-\varepsilon)}|\mathbf{e}| + |\hat{C}_0 + \hat{R}_1| \leq \frac{8}{\kappa(1-\varepsilon)}|\mathbf{e}|$$

which is smaller than the minimum distance of the quantum code whenever $|\mathbf{e}| \leq \frac{\kappa}{2^{11}}\kappa^2\delta^2(1-\varepsilon)\frac{n}{\Delta}$. This concludes the proof of Theorem 13.

6 Parallel Decoding

Without loss of generality, we again assume the error $\mathbf{e}$ to be a σ_x -type error. The analysis of the parallel decoder rests upon the following lemma.

Lemma 18. *Let $\varepsilon \in (0, 1/6)$, and assume that the state of the current mismatch Z is such that the sequential decoder of parameter ε converges. Then after one round of parallel decoding, the weight of the mismatch has been reduced by at least $\frac{1}{4}(\frac{1}{4} - \frac{3}{2}\varepsilon)|Z|$.*

To obtain that the parallel decoder converges in a logarithmic number of steps, it will remain to show that, provided the initial error vector is of sufficiently small weight, the mismatch will remain in a decodable state by the sequential decoder of parameter ε after an arbitrary number of parallel decoding steps. The result will then follow from applying Lemma 18 iteratively.

Proof of Lemma 18. Let $x_1, x_2, \dots, x_i \dots x_m$ be a sequence of local non-zero dual tensor codewords such that, for every $i \geq 1$,

$$|Z_{i-1}| - |Z_{i-1} + x_i| \geq (1 - \varepsilon)|x_i| \tag{9}$$

where $Z_0 = Z$ and $Z_i = Z_{i-1} + x_i$. That the sequential decoder of parameter ε converges means that whenever $x_1, \dots, x_i$ satisfy (9) and $Z_i \neq \emptyset$, then the sequence $x_1, \dots, x_i$ can be augmented by some x_{i+1} satisfying (9), until eventually $Z_m = \emptyset$.

It will be useful to keep in mind that the condition (9) is equivalent to the condition $|Z_{i-1} \cap x_i| \geq (1 - \varepsilon/2)|x_i|$.

Let us define $x'_1 = x_1 \cap Z$ and for $i \geq 2$, $x'_i = x_i \cap (Z \setminus \bigcup_{j < i} x_j)$, so that the x'_i are disjoint and partition Z . It may happen that some x'_i are empty.

Let G (good) be the subset of indices $1 \leq i \leq m$ for which it holds that

$$|x'_i| \geq \left(1 - \frac{3}{2}\varepsilon\right) |x_i| \quad (10)$$

and let B (bad) be the set of remaining indices. Let us first prove:

Claim 1. We have

$$|Z \cap \bigcup_{i \in G} x'_i| \geq |Z|/2.$$

Let us write the mismatch vector after addition of x_i as $Z_i = Z'_i \cup A_i$, where the union is disjoint and where we have set

$$Z'_i = Z \setminus \bigcup_{j \leq i} x'_j.$$

The situation is illustrated on Figure 2.

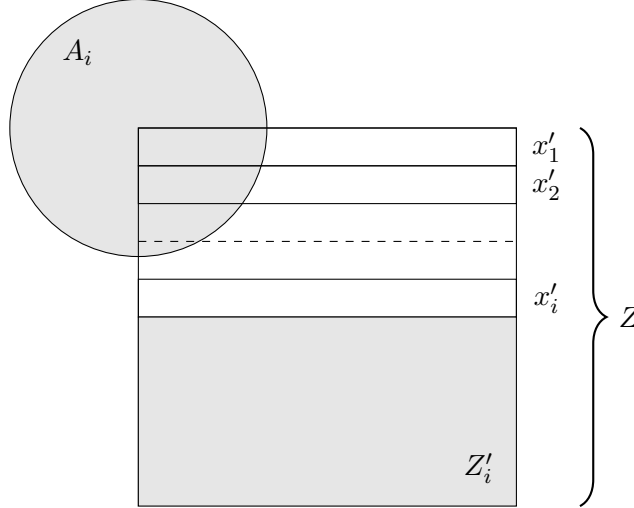


Figure 2: The state $Z_i = Z'_i \cup A_i$ of the mismatch after i steps (shaded): The supports of $x'_1, \dots, x'_i$ have been removed from the original mismatch Z , and an additional set A_i has been added.

Whenever x_i is added to the current mismatch Z_{i-1} , x'_i is removed from the support of Z'_i , and the set of 'additional' bits/coordinates A_{i-1} is modified to become A_i . We remark that for every i , the number of bits that are added to A_{i-1} , *i.e.* $|A_i \setminus A_{i-1}|$, is at most $\frac{\varepsilon}{2}|x_i|$, since x_i is an ε -sequential decoder increment. Furthermore, for $i \in B$, the condition (10) implies that

$$|A_{i-1} \setminus A_i| \geq \varepsilon|x_i|$$

so that $|A_{i-1}| - |A_i| \geq \frac{\varepsilon}{2}|x_i|$. Summarising, we have

$$\begin{aligned} |A_i| - |A_{i-1}| &\leq \frac{\varepsilon}{2}|x_i| \quad \text{for } i \in G, \\ |A_i| - |A_{i-1}| &\leq -\frac{\varepsilon}{2}|x_i| \quad \text{for } i \in B, \end{aligned}$$

from which we obtain

$$0 = |A_m| - |A_0| = \sum_{i=1}^m |A_i| - |A_{i-1}| \leq \frac{\varepsilon}{2} \left(\sum_{i \in G} |x_i| - \sum_{i \in B} |x_i| \right),$$

that is,

$$\sum_{i \in B} |x_i| \leq \sum_{i \in G} |x_i|.$$

From the definitions of the sets B and G , we have therefore

$$\sum_{i \in B} |x'_i| \leq \sum_{i \in B} (1 - \frac{3}{2}\varepsilon)|x_i| \leq \sum_{i \in G} (1 - \frac{3}{2}\varepsilon)|x_i| \leq \sum_{i \in G} |x'_i|$$

which proves the claim since the x'_i partition Z .

To finish the proof of the lemma, consider the vector u , equal to the *union* of all the bits that have been flipped by the decoder after all four substeps of the parallel decoding step. Consider a local dual tensor codeword x_i from the above sequence $x_1, \dots, x_m$, for $i \in G$. We wish to prove that a constant fraction of its support is included in the support of u . The local vector x_i sits in the local view of some vertex v . Let $y = |u \cap x'_i|$. We must have $|x'_i| - y < \frac{3}{4}|x_i|$. Indeed, suppose otherwise: let z_v be the local dual tensor codeword that the parallel local decoder at v has flipped when it was its turn to decode (with possibly $z_v = 0$). Note that we must have $z_v \subset u$ by definition of u . We have

$$|\hat{Z}| - |\hat{Z} + z_v| \geq |z_v|/2 \tag{11}$$

where $\hat{Z}$ is the local view of the mismatch at v , at the moment when it is v 's turn to decode. Now since $x'_i \setminus u$ has been left untouched by the decoder throughout the round of parallel decoding, we must have $(x'_i \setminus u) \subset \hat{Z}$ and $(x'_i \setminus u) \subset \hat{Z} + z_v$. Therefore, it must be the case that if, just after v has flipped z_v , we flip x_i , we remove at least $|x'_i| - y \geq \frac{3}{4}|x_i|$ coordinates from the current mismatch $\hat{Z} + z_v$ and therefore add at most $|x_i|/4$ coordinates to it. In other words,

$$|\hat{Z} + z_v| - |\hat{Z} + z_v + x_i| \geq |x_i|/2.$$

Adding to this inequality the inequality (11), we get

$$|\hat{Z}| - |\hat{Z} + z_v + x_i| \geq |x_i|/2 + |z_v|/2 \geq |x_i + z_v|/2.$$

Since we have at least $|x'_i \setminus u| \geq \frac{3}{4}|x_i|$ coordinates of x_i that are disjoint from u and hence disjoint from z_v , we must have $|x_i + z_v| > |z_v|$, which means that the local decoder at v

would have preferred $x_i + z_v$ over z_v when it was its turn to decode, since it is instructed to maximise the Hamming weight of the local codeword it flips. This establishes the contradiction and proves therefore

$$|x'_i| - y < \frac{3}{4}|x_i|.$$

This implies

$$y > |x'_i| - \frac{3}{4}|x_i| \geq (1 - \frac{3}{2}\varepsilon)|x_i| - \frac{3}{4}|x_i| = (\frac{1}{4} - \frac{3}{2}\varepsilon)|x_i|$$

since $i \in G$, and therefore, since the x'_i are disjoint,

$$|u| \geq \left(\frac{1}{4} - \frac{3}{2}\varepsilon\right) \sum_{i \in G} |x_i|.$$

By writing $\sum_{i \in G} |x_i| \geq \sum_{i \in G} |x'_i|$ and applying Claim 1, we get that

$$|u| \geq \left(\frac{1}{4} - \frac{3}{2}\varepsilon\right) \frac{1}{2}|Z|.$$

We conclude the proof by noticing that if the total number of bits that are flipped by the decoder is N (where a bit may be flipped multiple times, everytime contributing to N), then the mismatch weight is decreased by at least $N/2$, by definition of the decoding criterion; so the claim follows since the total number of bits flipped by the decoder is at least $|u|$. $\square$

Lemma 19. *Let $\varepsilon \in (0, 1/2)$. If the Hamming weight $|e|$ is less than*

$$\frac{1}{2^{16}} \varepsilon^3 \kappa^2 \delta^2 \frac{n}{\Delta},$$

Then sequential decoding with parameter $1/2$ is guaranteed to terminate. Furthermore, at any stage during the mismatch decoding procedure, if the sequential decoding parameter is switched from $1/2$ to ε , then mismatch decomposition is also guaranteed to terminate.

What the conclusion of Lemma 19 means is that, if $Z = \hat{Z}_0$ is the original mismatch vector after preprocessing, denoting by $\hat{Z}_\ell$ the mismatch vector after ℓ sequential decoding steps, *i.e.*

$$\hat{Z}_\ell = Z + x_1 + \dots + x_\ell,$$

where $x_1, \dots, x_\ell$ is the sequence of local dual tensor codeword increments, satisfying

$$|\hat{Z}_j| - |\hat{Z}_j + x_{j+1}| \geq |x_{j+1}|/2$$

for $j < \ell$, then it is possible to extend the sequences $x_1, \dots, x_\ell$ and $\hat{Z}_0, \dots, \hat{Z}_\ell$ to $x_1, \dots, x_i, \dots, x_m$ and $\hat{Z}_0, \dots, \hat{Z}_i, \dots, \hat{Z}_m = 0$, with

$$|\hat{Z}_j| - |\hat{Z}_j + x_{j+1}| \geq (1 - \varepsilon)|x_{j+1}|$$

for $j = \ell \dots m - 1$.

Proof of Lemma 19. Assume that after a certain number of sequential decoding steps, the decoder has flipped successively the local dual tensor codewords $x_1, x_2, \dots, x_\ell$. Since for every codeword x_j that is flipped the mismatch decreases by at least $|x_j|/2$, we must have

$$\frac{1}{2}(|x_1| + |x_2| + \dots + |x_\ell|) \leq |Z|.$$

where $|Z|$ is the original mismatch vector computed after the preprocessing phase. Robustness implies that $\|x_j\| \leq \frac{1}{\kappa\Delta}|x_j|$ and we may therefore write

$$\|x_1\| + \|x_2\| + \dots + \|x_\ell\| \leq \frac{2}{\kappa\Delta}|Z|. \quad (12)$$

Applying Lemma 17 we therefore obtain that

$$\|\hat{Z}_\ell\| \leq \|Z\| + \|x_1\| + \|x_2\| + \dots + \|x_\ell\| \leq \frac{4}{\kappa\Delta}(1+2)|e| \leq \frac{12}{\kappa\Delta}|e|$$

where $\hat{Z}_\ell$ is the current mismatch after $x_1, \dots, x_\ell$ have been flipped, and we used that $\kappa < 1$. Therefore, if we impose the condition

$$|e| \leq \frac{1}{2^{16}}\varepsilon^3\kappa^2\delta^2\frac{n}{\Delta} < \frac{1}{2^{12}}\frac{\varepsilon^3}{12}\kappa^2\delta^2\frac{n}{\Delta} = \frac{\kappa\Delta}{12}\frac{1}{2^{12}}\delta^2\varepsilon^3\kappa|V_{00}|,$$

we obtain that $\|\hat{Z}_\ell\|$ must be bounded from above by $\frac{1}{2^{12}}\delta^2\varepsilon^3\kappa|V_{00}|$. Since the set S of active vertices for $\|\hat{Z}_\ell\|$ in any one of the four types of vertices $V_{00}, V_{01}, V_{10}, V_{11}$ must satisfy $|S| \leq |\hat{Z}_\ell|$, we have that Theorem 12 applies and there exists $x_{\ell+1}$ such that $|\hat{Z}_\ell| - |\hat{Z}_\ell + x_{\ell+1}| \geq (1-\varepsilon)|x_{\ell+1}|$. Since $\varepsilon \leq 1/2$, we have that $x_{\ell+1}$ is also a valid choice for the sequential decoder with parameter $1/2$, and we may therefore reapply the very same argument iteratively to obtain the required sequence $x_{\ell+1}, \dots, x_m$. $\square$

Theorem 20. Fix $\varepsilon \in (0, 1/6)$. If the Hamming weight $|e|$ is less than

$$\frac{1}{2^{12}} \min\left(\frac{\varepsilon^3}{16}, \kappa\right) \delta^2 \kappa^2 \frac{n}{\Delta},$$

then the parallel decoder returns a valid correction in logarithmic time.

Proof. We make the remark that a substep of parallel decoding has the same effect on the mismatch as that of a sequence of successive increments by the sequential decoder of parameter $1/2$. Therefore, after any number of rounds of parallel decoding, we are in the same situation as if a sequential decoder of parameter $1/2$ had been applied, and Lemma 19 applies, which in turn implies that Lemma 18 applies, so that the parallel decoder terminates in a logarithmic number of rounds.

It remains to prove that the output $\hat{e}$ of the decoder is such that $|e + \hat{e}|$ is smaller than the minimum distance of the quantum code. This argument is very close to that of the end of the proof of Theorem 13.

Recall that

$$|\mathbf{e} + \hat{\mathbf{e}}| = \left| \sum_{v \in V_{10}} (e_v + \varepsilon_v) + \hat{C}_0 + \hat{R}_1 \right| \quad (13)$$

where e_v is the initial error vector restricted to the local view at v , and $\hat{C}_0, \hat{R}_1$ are the output of the decoder. We have $|\varepsilon_v| \leq |e_v|$ so that

$$|\mathbf{e} + \hat{\mathbf{e}}| \leq 2|\mathbf{e}| + |\hat{C}_0 + \hat{R}_1|.$$

To evaluate $|\hat{C}_0 + \hat{R}_1|$, we make the remark that every local dual tensor codeword x_i that is used by an iteration of the parallel decoder contributes at most $\|x_i\|$ non-zero row and column vectors to $\hat{C}_0 + \hat{R}_1$. Therefore,

$$|\hat{C}_0 + \hat{R}_1| \leq \Delta \sum_i \|x_i\|$$

where the sum runs over all increments applied by the decoder. Applying robustness $\|x_i\| \leq \frac{1}{\kappa \Delta} |x_i|$ and

$$\frac{1}{2} \sum_i |x_i| \leq |Z|$$

where Z is the initial mismatch vector, we obtain

$$|\hat{C}_0 + \hat{R}_1| \leq \frac{2}{\kappa} |Z|.$$

Writing $|Z| \leq 4|\mathbf{e}|$ from Lemma 17, we get

$$|\hat{C}_0 + \hat{R}_1| \leq \frac{8}{\kappa} |\mathbf{e}|.$$

Since $\kappa < 1$, we may write $2|\mathbf{e}| \leq \frac{2}{\kappa} |\mathbf{e}|$, and now we have

$$|\mathbf{e} + \hat{\mathbf{e}}| \leq \frac{2}{\kappa} |\mathbf{e}| + |\hat{C}_0 + \hat{R}_1| \leq \frac{10}{\kappa} |\mathbf{e}|$$

which is smaller than the minimum distance of the quantum code whenever $|\mathbf{e}| \leq \frac{\kappa^3 \delta^2}{2^{12}} \frac{n}{\Delta} < \frac{\kappa^3 \delta^2}{10 \cdot 2^8} \frac{n}{\Delta}$. $\square$

Acknowledgements

We thank Ting-Chun Lin for bringing up an issue with the parallelisation of the decoder in a previous version of this manuscript. We acknowledge support from the Plan France 2030 through the project ANR-22-PETQ-0006. GZ also acknowledges support from the ANR through the project QUDATA, ANR-18-CE47-0010.

References

- [1] D. Gottesman, “Fault-tolerant quantum computation with constant overhead,” *Quantum Information & Computation*, vol. 14, no. 15-16, pp. 1338–1372, 2014.
- [2] A. A. Kovalev and L. P. Pryadko, “Fault tolerance of quantum low-density parity check codes with sublinear distance scaling,” *Phys. Rev. A*, vol. 87, p. 020304, Feb 2013.
- [3] O. Fawzi, A. Grospellier, and A. Leverrier, “Constant overhead quantum fault-tolerance with quantum expander codes,” in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 743–754.
- [4] M. A. Tremblay, N. Delfosse, and M. E. Beverland, “Constant-overhead quantum error correction with thin planar connectivity,” *arXiv preprint arXiv:2109.14609*, 2021.
- [5] N. P. Breuckmann and J. N. Eberhardt, “Quantum low-density parity-check codes,” *PRX Quantum*, vol. 2, p. 040101, Oct 2021.
- [6] J.-P. Tillich and G. Zémor, “Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength,” *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2014.
- [7] M. B. Hastings, J. Haah, and R. O’Donnell, “Fiber Bundle Codes: Breaking the $n^{1/2}\text{polylog}(n)$ Barrier for Quantum LDPC Codes,” in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 1276–1288.
- [8] N. P. Breuckmann and J. N. Eberhardt, “Balanced product quantum codes,” *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6653–6674, 2021.
- [9] P. Panteleev and G. Kalachev, “Quantum LDPC Codes With Almost Linear Minimum Distance,” *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 213–229, 2022.
- [10] —, “Asymptotically good quantum and locally testable classical ldpc codes,” in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 375–388.
- [11] A. Leverrier and G. Zémor, “Quantum Tanner codes,” *arXiv preprint arXiv:2202.13641*, 2022, to appear in the proceedings of FOCS 2022.
- [12] C. Monroe and J. Kim, “Scaling the ion trap quantum processor,” *Science*, vol. 339, no. 6124, pp. 1164–1169, 2013.

- [13] T. Rudolph, “Why I am optimistic about the silicon-photonics route to quantum computing,” *APL Photonics*, vol. 2, no. 3, p. 030901, 2017.
- [14] M. Morgado and S. Whitlock, “Quantum simulation and computing with rydberg-interacting qubits,” *AVS Quantum Science*, vol. 3, no. 2, p. 023501, 2021.
- [15] P. Panteleev and G. Kalachev, “Degenerate Quantum LDPC Codes With Good Finite Length Performance,” *Quantum*, vol. 5, p. 585, Nov. 2021.
- [16] A. Grospellier, L. Grouès, A. Krishna, and A. Leverrier, “Combining hard and soft decoders for hypergraph product codes,” *Quantum*, vol. 5, p. 432, Apr. 2021.
- [17] J. Roffe, D. R. White, S. Burton, and E. Campbell, “Decoding across the quantum low-density parity-check code landscape,” *Phys. Rev. Research*, vol. 2, p. 043423, Dec 2020.
- [18] A. O. Quintavalle, M. Vasmer, J. Roffe, and E. T. Campbell, “Single-shot error correction of three-dimensional homological product codes,” *PRX Quantum*, vol. 2, p. 020340, Jun 2021.
- [19] N. Delfosse, V. Londe, and M. E. Beverland, “Toward a union-find decoder for quantum LDPC codes,” *IEEE Transactions on Information Theory*, vol. 68, no. 5, pp. 3187–3199, 2022.
- [20] N. P. Breuckmann and V. Londe, “Single-shot decoding of linear rate LDPC quantum codes with high performance,” *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 272–286, 2022.
- [21] J. du Crest, M. Mhalla, and V. Savin, “Stabilizer inactivation for message-passing decoding of quantum ldpc codes,” *arXiv preprint arXiv:2205.06125*, 2022.
- [22] A. Leverrier, J.-P. Tillich, and G. Zémor, “Quantum expander codes,” in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 2015, pp. 810–824.
- [23] S. Evra, T. Kaufman, and G. Zémor, “Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders,” in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, 2020, pp. 218–227.
- [24] A. Leverrier and G. Zémor, “Efficient decoding up to a constant fraction of the code length for asymptotically good quantum codes,” *arXiv preprint arXiv:2206.07571*, 2022, to appear in the proceedings of SODA 2023.
- [25] S. Gu, C. A. Pattison, and E. Tang, “An efficient decoder for a linear distance quantum LDPC code,” *arXiv preprint arXiv:2206.06557*, 2022.
- [26] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, “Good quantum LDPC codes with linear time decoders,” *arXiv preprint arXiv:2206.07750*, 2022.

- [27] T.-C. Lin and M.-H. Hsieh, “Good quantum LDPC codes with linear time decoder from lossless expanders,” *arXiv preprint arXiv:2203.03581*, 2022.
- [28] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [29] M. Sipser and D. A. Spielman, “Expander codes,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.
- [30] I. Dinur, S. Evra, R. Livne, A. Lubotzky, and S. Mozes, “Locally testable codes with constant rate, distance, and locality,” in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 357–374.
- [31] A. Steane, “Multiple-particle interference and quantum error correction,” *Proc. R. Soc. Lond. A*, vol. 452, no. 1954, pp. 2551–2577, 1996.
- [32] R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Physical Review A*, vol. 54, pp. 1098–1105, Aug 1996.
- [33] G. Kalachev and P. Panteleev, “Two-sided robustly testable codes,” *arXiv preprint arXiv:2206.09973*, 2022.
- [34] M. Hopkins and T.-C. Lin, “Explicit Lower Bounds Against $\Omega(n)$ -Rounds of Sum-of-Squares,” *arXiv preprint arXiv:2204.11469*, 2022, to appear in the proceedings of FOCS 2022.
- [35] M. H. Freedman and M. B. Hastings, “Quantum systems on non-k-hyperfinite complexes: a generalization of classical statistical mechanics on expander graphs,” *Quantum Information & Computation*, vol. 14, no. 1-2, pp. 144–180, 2014.
- [36] A. Anshu, N. P. Breuckmann, and C. Nirkhe, “NLTS Hamiltonians from good quantum codes,” *arXiv preprint arXiv:2206.13228*, 2022.
- [37] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown, “Low-overhead fault-tolerant quantum computing using long-range connectivity,” *Science Advances*, vol. 8, no. 20, p. eabn1717, 2022.
- [38] S. Hoory, N. Linial, and A. Wigderson, “Expander graphs and their applications,” *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.
- [39] V. Guruswami, “Expander codes and their decoding,” <https://www.cs.cmu.edu/~venkatg/teaching/codingtheory/notes/notes8.pdf>, 2010.
- [40] D. Gottesman, “Stabilizer codes and quantum error correction,” Ph.D. dissertation, California Institute of Technology, 1997.

- [41] A. M. Steane, “Error correcting codes in quantum theory,” *Physical Review Letters*, vol. 77, pp. 793–797, Jul 1996.
- [42] O. Goldreich, “On the Locally Testable Code of Dinur et al.(2021).” in *Electron. Colloquium Comput. Complex.*, vol. 28, 2021, p. 175.