Higher-dimensional hypergraph-product codes

Leonid P. Pryadko and Weilei Zeng

UCR

(Dated: August 15, 2018)

A higher-dimensional version of the quantum hypergraph-product ansatz is described. Most important feature are the sharp lower and upper bounds for the minimum distance.

I. INTRODUCTION

The construction is useful for (a) analyzing repeated measurement in a stabilizer code in the problem of fault-tolerant (FT) quantum error correction (e.g., surface code[1], or more general LDPC code[2], (b) related problem of single-shot error correction[3], (c) analysis and generalization of transformations between QECCs, like the quantum code enlargement trick by Hastings [ref?], and (d) construction of asymmetric quantum CSS codes optimized for operation where X and Z channels are strongly asymmetric[4–9].

Origin: (a) Higher-dimensional toric codes [**ref**?] and (b) Hypergraph-product codes[10]. Present construction generalizes both.

II. CONSTRUCTION OVERVIEW

We want to generalize the QHP construction[10] to produce longer chain complexes from two, three, etc. arbitrary size binary matrices. Here are some examples of intended construction.

- Original QHP construction can be interpreted as a length-two chain complex $\mathcal{K}(A_1, A_2)$: $\{0\} \leftarrow C_0 \overset{A_1}{\leftarrow} C_1 \overset{A_2}{\leftarrow} C_2 \leftarrow \{0\}$ with boundary operators such that $A_1 A_2 = 0$ (below, the indices above and to the left of the double lines label the corresponding sectors),

$$A_1 = \left(\begin{array}{c||c|c} & x & y \\ \hline\hline \cdot & H_1\otimes E & E\otimes H_2 \end{array}\right), \quad A_2 = \left(\begin{array}{c||c} & xy \\ \hline\hline x & E\otimes H_2 \\ \hline y & H_1\otimes E \end{array}\right). \tag{1}$$

  Here H_i, $i \in \{1, 2\}$ are binary matrices and E denotes identity matrices of the sizes needed to make the block rows and block columns of matching sizes.

- The following three-chain complex $\mathcal{K}(B_1, B_2, B_3)$ $\{0\} \leftarrow C_0 \overset{B_1}{\leftarrow} C_1 \overset{B_2}{\leftarrow} C_2 \overset{B_3}{\leftarrow} C_3$ with boundary operators $B_1 B_2 = 0$, $B_2 B_3 = 0$ gives a generalization of the 3D toric codes' construction:

$$B_1 = (\cdot \| H_1\otimes E\otimes E | E\otimes H_2\otimes E | E\otimes E\otimes H_3) = (A_1\otimes E | E\otimes H_3)\,,$$

$$B_2 = \left(\begin{array}{c||c|c|c} & xy & xz & yz \\ \hline\hline x & E\otimes H_2\otimes E & E\otimes E\otimes H_3 & \\ \hline y & H_1\otimes E\otimes E & & E\otimes E\otimes H_3 \\ \hline z & & H_1\otimes E\otimes E & E\otimes H_2\otimes E \end{array}\right) = \left(\begin{array}{c|c} A_2\otimes E & E\otimes H_3 \\ \hline & A_1\otimes E \end{array}\right),$$

$$B_3 = \left(\begin{array}{c|c} & xyz \\ \hline\hline xy & E\otimes E\otimes H_3 \\ \hline xz & E\otimes H_2\otimes E \\ \hline yz & H_1\otimes E\otimes E \end{array}\right) = \left(\begin{array}{c} E\otimes H_3 \\ \hline A_2\otimes E \end{array}\right).$$

The simple rule is that the matrix H_1, H_2, or H_3 is placed at the position whose column label differs from the row label by x, y, or z, respectively. The construction has an obvious generalization to higher dimensions. In the following, it is more convenient to use the following (equivalent) recursive definition:

Definition 1 (Dimensional extension of a chain complex)**.** *Given an $(m-1)$-chain complex $\mathcal{A} \equiv \mathcal{K}(A_1,\ldots,A_{m-1})$: $\{0\} \leftarrow C_0 \stackrel{A_1}{\leftarrow} C_1 \stackrel{A_2}{\leftarrow} C_2 \ldots \stackrel{A_{m-1}}{\leftarrow} C_{m-1} \stackrel{0_{n_{m-1}\times 0}}{\leftarrow} \{0\}$ with non-trivial boundary operators A_j, $j=1,\ldots,m-1$, such that $A_jA_{j+1}=0$, and an $r\times c$ binary matrix P, the extended m-chain complex $\mathcal{B}\equiv\mathcal{K}(B_1,\ldots,B_m)$ is defined by the boundary operators*

$$B_1 = (A_1\otimes E_r | E_{n_0}\otimes P), \quad B_2 = \left(\begin{array}{c|c} A_2\otimes E_r & E_{n_1}\otimes P \\ \hline & A_1\otimes E_c \end{array}\right), \ldots \tag{2}$$

$$B_j = \left(\begin{array}{c|c} A_j\otimes E_r & E_{n_{j-1}}\otimes P \\ \hline & A_{j-1}\otimes E_c \end{array}\right), \quad \ldots \tag{3}$$

$$B_{m-1} = \left(\begin{array}{c|c} A_{m-1}\otimes E_r & E_{n_{m-2}}\otimes P \\ \hline & A_{m-2}\otimes E_c \end{array}\right), \tag{4}$$

$$B_m = \left(\begin{array}{c} E_{n_{m-1}}\otimes P \\ \hline A_{m-1}\otimes E_c \end{array}\right). \tag{5}$$

Here $E_r \equiv E(r)$ denotes the $r\times r$ identity matrix, and the original linear spaces C_i, $i\in\{0,\ldots,m-1\}$, have dimensions n_i (so that A_i is an $n_{i-1}\times n_i$ binary matrix), with the additional convention $n_j=0$ for $j<0$ and $j\geq m$. The dimension of thus defined j-th level extended linear space $\mathcal{C}'_j$ is $n'_j = n_jr + n_{j-1}c$, $j\in\{0,\ldots,m\}$.

The constructed matrices trivially satisfy the correct orthogonality conditions. The original non-recursive definitions can be recovered by starting with the two-chain complex (1) and extending it sequentially with the help of matrices H_3, H_4, etc.

Let us assume that the rank of the j-th homology group $\mathcal{H}_j \equiv \mathcal{H}_j(\mathcal{A})$ is $k_j = \mathrm{rank}(\mathcal{H}_j)$, and the corresponding *distance*, the minimum weight of a homologically non-trivial cycle in $\mathcal{H}_j$ is $d_j \geq 1$, $1\leq j<m$. Here we use the convention that $d_j=\infty$ if $k_j=0$. The distance d_j can be also expressed as the (left or z) distance of a quantum CSS code with generators $G_x = A_j$ and $G_z = A_{j+1}^T$, denoted as $\mathcal{Q}(A_j, A_{j+1}^T)$, that is,

$$d_j(\mathcal{A}) \equiv d_j(A_j, A_{j+1}^T) = \min_{\mathbf{e}\in\mathcal{C}_{A_j}^{\perp}\setminus\mathcal{C}_{A_{j+1}^T}} \mathrm{wgt}\,\mathbf{e}.$$

Here $\mathcal{C}_G$ and $\mathcal{C}_H^{\perp}$ denote the binary codes with the generator matrix G and the check matrix H, respectively.

The parameters of the CSS code $\mathcal{Q}(A_j, A_{j+1}^T)$ are thus $[[n_j, k_j, \min(d_j, \tilde{d}_j)]]$, where the distances in the corresponding co-chain complex generated by transposed matrices A_j^T, with the list taken in the opposite order,

$$\tilde{\mathcal{A}} \equiv \{0\} \leftarrow \mathcal{C}_{m-1} \stackrel{A_{m-1}^T}{\leftarrow} \ldots \stackrel{A_1^T}{\leftarrow} \mathcal{C}_0 \leftarrow \{0\},$$

are denoted with the tilde, $\tilde{d}_j \equiv d(A_{j+1}^T, A_j)$.

For the following, it is convenient to extend the definitions of the distance to also apply for the portions $\{0\} \leftarrow C_0$, as well as the portion $C_{m-1} \leftarrow \{0\}$. That is, we would like to formally define quantum CSS codes $\mathcal{Q}(0_{0\times n_0}, A_1^T)$ and $\mathcal{Q}(A_{m-1}, 0_{n_{m-1}\times 0})$. In the former case, we are looking at binary vectors in $\mathbb{F}_{n_0}$ orthogonal to a zero-row matrix (all 2^{n_0} vectors), except those that are linearly dependent with the rows of A_0^T. It is easy to check that the minimum distance of such a code is 1, except when A_0^T is a full-row-rank matrix, in which case $\mathcal{Q}(0, A_0^T)$ is empty. By convention, in the latter case, we are going to set $d = \infty$. Thus, we have $\boxed{d_0 \equiv d(0, A_0) = k_0 > 0 ? 1 : \infty,}$ that is, distance of the code $\mathcal{Q}(0, A_0)$ is one unless $n_0 = \operatorname{rank} A_0$, in which case it is infinite.

The situation is even simpler in the case of $\mathcal{Q}_m = \mathcal{Q}(A_{m-1}, 0)$, which is just a classical binary code with the check matrix A_{m-1} of dimension $n_{m-1} \times n_m$ (the zero matrix does not remove any vectors from $\mathcal{C}^{\perp}_{A(m-1)}$). The parameters of this classical code with the check matrix A_{m-1} are $[n_{m-1}, n_{m-1} - \operatorname{rank} A_{m-1}, d_m]$. Again, by convention, the distance d_m is infinite if the code is empty, that is, if A_{m-1} is a full-column-rank matrix.

Main result of this work are the exact parameters of the dimensionally-extended chain complex $\mathcal{B}$, expressed in terms of those of the original chain complex $\mathcal{A}$ and the parameters of the two binary codes with parity check matrices P and P^T, respectively: $\mathcal{C}_P^{\perp} = [c, \kappa, \delta]$ and $\mathcal{C}_{P^T}^{\perp} = [r, \tilde{\kappa}, \tilde{\delta}]$. Here P is an $r \times c$ binary matrix with $\operatorname{rank} P = u$, so that $\kappa = c - u$ and $\tilde{\kappa} = r - u$; and the distances $\delta \geq 1$, $\tilde{\delta} \geq 1$. Throughout this work the infinite-distance-empty-code convention is used: $\delta = \infty$ if $\kappa = 0$, and $\tilde{\delta} = \infty$ if $\tilde{\kappa} = 0$.

With these definitions, the parameters of the dimensionally-extended m-chain complex $\mathcal{B}$ (see Definition 1) are given by the following

Theorem 2. *The dimension of the space $\mathcal{C}'_j$ is $\boxed{n'_j = n_j r + n_{j-1} c,}$ the rank of the j th homology group $\mathcal{H}_j(\mathcal{B})$ is $\boxed{k'_j = k_j \tilde{\kappa} + k_{j-1}\kappa,}$ with the minimum distance [weight of the smallest homologically non-trivial cycle in $\mathcal{H}_j(\mathcal{B})$] $\boxed{d'_j = d_{j-1}\delta \text{ if } \tilde{\kappa} = 0, \text{ otherwise } d'_j = \min(d_j, d_{j-1}\delta).}$*

This theorem combines the results of Lemma 3, Theorem 4 (upper distance bounds), and Theorem 6 (lower distance bound) below. The value of n'_j directly follows from the Definition 1.

III. MATRIX RANKS AND DIMENSIONS OF CSS CODES.

Any pair of adjacent matrices A_j and A_{j+1}^T can obviously be used as generators of a quantum CSS code $\mathcal{Q}_j \equiv \mathcal{Q}(A_j, A_{j+1}^T)$, which we assume to encode k_j qubits in n_j. Alternatively, k_j is the rank of the j th homology group in the original chain complex $\mathcal{A}$.

What are the parameters $[[n'_j, k'_j, d'_j]]$ of the codes generated by dimensionally extended matrices, e.g., $\mathcal{Q}'_j \equiv \mathcal{Q}(B_j, B_{j+1}^T)$?

We already know the block length $n'_j = n_j r + n_{j-1} c$. To find k'_j, denote the original ranks $s_j = \operatorname{rank} A_j$, $u \equiv \operatorname{rank} P$, and prove:

Lemma 3. *Given $A_{j-1}A_j = 0$, the binary matrix*

$$B_j = \left(\begin{array}{c|c} A_j \otimes E_r & E_{n_{j-1}} \otimes P \\ \hline & A_{j-1} \otimes E_c \end{array}\right)$$

has the rank $s'_j \equiv \operatorname{rank} B_j = s_j(r-u) + s_{j-1}(c-u) + n_{j-1}u$.

Proof. Start by computing the ranks of the upper and lower row blocks using the trick from Ref. 10. Namely, for each block, use row transformations to form distinct zero combinations. The corresponding count for the upper row block is

$$\operatorname{lc} B'_j = (n_{j-1} - s_j)(r-u),$$

and for the lower row block, trivially, $\operatorname{lc} B''_j = (n_{j-2} - s_{j-1})c$. This gives $\operatorname{rank} B'_j = s_j r + n_{j-1}u - s_j u$, and $\operatorname{rank} B''_j = cs_{j-1}$. For matrices of this form, the number of zero linear combinations that involve both row blocks is just the product of the ranks in the right blocks,

$$\operatorname{lc}(B'_j, B''_j) = s_{j-1}u.$$

Overall, we get the stated rank of B_j,

$$\begin{aligned} s'_j = \operatorname{rank} B_j &= (n_{j-1}r + n_{j-2}c) - (n_{j-1} - s_j)(r-u) - (n_{j-2} - s_{j-1})c - s_{j-1}u \\ &= n_{j-1}r + n_{j-2}c - n_{j-1}r + rs_j + n_{j-1}u - s_j u - cn_{j-2} + cs_{j-1} - s_{j-1}u \\ &= s_j(r-u) + s_{j-1}(c-u) + n_{j-1}u. \end{aligned}$$

□

The same result can also be obtained considering column blocks. Check:

$$\begin{aligned} s'_j &= n_j r + n_{j-1}c - (n_{j-1} - s_{j-1})(c-u) - (n_j - s_j)r - s_j u \\ &= n_{j-1}u + cs_{j-1} - s_{j-1}u + rs_j - s_j u = \boxed{s_j(r-u) + s_{j-1}(c-u) + n_{j-1}u.} \end{aligned}$$

These expressions give

$$\begin{aligned} k'_j &= n'_j - \operatorname{rank} B_j - \operatorname{rank} B_{j+1} \\ &= n_j r + n_{j-1}c - s_j(r-u) - s_{j-1}(c-u) - n_{j-1}u - s_{j+1}(r-u) - s_j(c-u) - n_j u \\ &= (n_j - s_j - s_{j+1})(r-u) + (n_{j-1} - s_{j-1} - s_j)(c-u) \\ &= k_j\tilde{\kappa} + k_{j-1}\kappa, \end{aligned} \tag{6}$$

where $\kappa \equiv c-u$ and $\tilde{\kappa} \equiv r-u$ respectively are the dimensions of the binary codes using P and P^T as check matrices.

For the length-two chain complex (1) this gives (setting $s_0 = 0$):

$$k_0^{(2)} = r_1 r_2 - \operatorname{rank} B_1 = (r_1 - u_1)(r_2 - u_2) = \tilde{\kappa}_1\tilde{\kappa}_2, \tag{7}$$

$$k_1^{(2)} = (c_1 - u_1)(r_2 - u_2) + (r_1 - u_1)(c_2 - u_2) = \tilde{\kappa}_1\kappa_2 + \kappa_1\tilde{\kappa}_2, \tag{8}$$

$$k_2^{(2)} = c_1 c_2 - \operatorname{rank} B_2 = (c_1 - u_1)(c_2 - u_2) = \kappa_1\kappa_2, \tag{9}$$

where the k_0 and k_2 correspond to classical codes with check matrices B_1^T and B_2, respectively. The value of $k_1^{(2)}$ coincides with the result in Ref. 10. For length-3 chain complex one gets, explicitly:

$$\begin{aligned} k_0^{(3)} &= \tilde{\kappa}_1\tilde{\kappa}_2\tilde{\kappa}_3, \\ k_1^{(3)} &= \kappa_1\tilde{\kappa}_2\tilde{\kappa}_3 + \tilde{\kappa}_1\kappa_2\tilde{\kappa}_3 + \tilde{\kappa}_1\tilde{\kappa}_2\kappa_3, \\ k_2^{(3)} &= \kappa_1\kappa_2\tilde{\kappa}_3 + \kappa_1\tilde{\kappa}_2\kappa_3 + \tilde{\kappa}_1\kappa_2\kappa_3, \\ k_3^{(3)} &= \kappa_1\kappa_2\kappa_3. \end{aligned}$$

Generally, $k_j^{(m)}$ is a sum of all the products of distinct κ_s, $s \in \{1, \ldots, m\}$ with all possible placements of $(m-j)$ tildes.

IV. BOUNDS ON CODES' DISTANCES

For a CSS code with binary generator matrices P and Q, such that $PQ^T = 0$, consider two distances different by the order of the generators:

$$d(P,Q) = \min_{\mathbf{x}\in\mathcal{C}_P^\perp\setminus\mathcal{C}_Q} \operatorname{wgt}\mathbf{x}, \quad d(Q,P) = \min_{\mathbf{x}\in\mathcal{C}_Q^\perp\setminus\mathcal{C}_P} \operatorname{wgt}\mathbf{x},$$

so that the actual CSS code distance is the minimum of the two. We will use the convention[10] that an empty code has infinite distance, which is the same as stating $\min(\emptyset) = \infty$. Notice that if one takes the right matrix zero, one can also get the distances of the classical binary code associated with the matrix P (dimensions $r \times c$),

$$d_P = d(P, 0_{0\times c}).$$

In any case, the distance of the code (quantum or classical, empty or not) cannot be zero.

Again, consider an $(m-1)$-chain complex

$$\mathcal{A} = \{0\} \overset{0_{0\times n_0}}{\leftarrow} C_0 \overset{A_1}{\leftarrow} C_1 \leftarrow \ldots \leftarrow C_{m-2} \overset{A_{m-1}}{\leftarrow} C_{m-1} \overset{0_{n_{m-1}\times 0}}{\leftarrow} \{0\},$$

and its associated dimensionally-extended m-chain complex $\mathcal{B}$, see Definition 1. Let the CSS code $\mathcal{Q}(A_j, A_{j+1}^T)$ have parameters $[[n_j, k_j]]$ and the left and right distances $d(A_j, A_{j+1}^T) = d_j$, $d(A_{j+1}^T, A_j) = \tilde{d}_j$, and the binary codes with the check matrices P and P^T have distances δ and $\tilde{\delta}$, respectively. We have the following upper bounds on the distances in K':

Theorem 4. *The following upper bounds apply to codes in $\mathcal{B}$:*

(a) If $\tilde{\kappa} \equiv r - u > 0$, then $d'_j \equiv d(B_j, B_{j+1}^T) \le d_j \equiv d(A_j, A_{j+1}^T)$.

(b) If $k_0 \equiv n_0 - \operatorname{rank} A_1 > 0$, then $d'_1 \le \delta$.

(c) For $j > 1$, $d'_j \equiv d(B_j, B_{j+1}^T) \le d_{j-1}\delta \equiv d(A_{j-1}, A_j^T)\,\delta$.

The proofs are based on Lemma 3 and its generalization, the following Lemma:

Lemma 5. *Consider two pairs of matrices: A, B of dimensions $n_0 \times n_1$, $n_1 \times n_2$ respectively, such that $AB = 0$, and P, Q of dimensions $m_0 \times m_1$, $m_1 \times m_2$ respectively, such that $PQ = 0$. Then the rank of the following block matrix*

$$M = \left(\begin{array}{c|c} B_{n_1\times n_2} \otimes E(m_0) & E(n_1) \otimes P_{m_0\times m_1} \\ \hline A_{n_0\otimes n_1} \otimes E(m_1) & E(n_0) \otimes Q_{m_1\times m_2} \end{array}\right) \quad \textit{is}$$

$$\begin{aligned} \operatorname{rank} M = {} & m_0 \operatorname{rank} B + n_1 \operatorname{rank} P - \operatorname{rank} B \operatorname{rank} P \\ & + m_1 \operatorname{rank} A + n_0 \operatorname{rank} Q - \operatorname{rank} A \operatorname{rank} Q - \operatorname{rank} A \operatorname{rank} P. \end{aligned}$$

In this expression, the first and second three terms are the ranks of the upper and lower row blocks, respectively; the last term is the number of rows that enter non-trivial linear combinations between the two blocks.

Proof of Theorem 4. For convenience, quote the definitions

$$B_1 = (A_1 \otimes E_r | E_{n_0} \otimes P),\ B_2 = \left(\begin{array}{c|c} A_2 \otimes E_r & E_{n_1} \otimes P \\ & A_1 \otimes E_c \end{array} \right),\ B_j = \left(\begin{array}{c|c} A_j \otimes E_r & E_{n_{j-1}} \otimes P \\ & A_{j-1} \otimes E_c \end{array} \right), \ldots$$

Part *(a)*: We only need to consider the case where $k_j \neq 0$. In this case we can find a minimum-weight codeword $\mathbf{c} \in \mathbb{F}_2^{n_j}$, $\mathrm{wgt}(\mathbf{c}) = d_j$, such that $A_j \mathbf{c}^T = 0$ but $\mathbf{c} \neq \alpha A_{j+1}^T$ for any $\alpha \in \mathbb{F}_2^{n_{j+1}}$. Clearly, $\mathbf{x} = (\mathbf{c} \otimes \mathbf{y} | 0)$ satisfies $B_j \mathbf{x}^T = 0$ with every $\mathbf{y} \in \mathbb{F}_2^r$. We want a $\mathbf{y}$ of weight one, such that $\mathbf{x}$ is not a linear combination of rows of B_{j+1}^T. To see that this is guaranteed by the condition $\tilde{\kappa} > 0$, consider the matrix B'_{j+1} constructed from the original A_j and a modified matrix A'_{j+1} of dimension $n_j \times (n_{j+1} + 1)$, which is A_{j+1} with an added column $\mathbf{c}^T$; one has $\mathrm{rank}\, A'_{j+1} = \mathrm{rank}\, A_{j+1} + 1$. By construction, $A_j A'_{j+1} = 0$. According to Lemma 3, the modified matrix B'_{j+1} has $\mathrm{rank}\, B'_{j+1} = \mathrm{rank}\, B_{j+1} + r - u$, which guarantees that at least $\tilde{\kappa} = r - u > 0$ of the added columns are indeed linearly independent from the columns of the original B_{j+1}. That is, we can find a weight-one vector $\mathbf{y}$ so that $\mathbf{x}$ is a valid codeword of weight d_j in $\mathcal{Q}(B_j, B_{j+1})$, which proves the upper bound.

Proof of Part *(b)* is similar, except now the trial vector has the form $\mathbf{x} = (0, \mathbf{y} \otimes \mathbf{b})$, where $P\mathbf{b}^T = 0$ (we only need to consider the case $\kappa > 0$). Any such vector clearly satisfies $B_1 \mathbf{x}^T = 0$. We want $\mathbf{y} \in \mathbb{F}_2^{n_0}$ of unit weight, such that $\mathbf{x}^T$ be linearly independent from the columns of B_2. Instead of analyzing each possible $\mathbf{y}$, consider the rank of the matrix B'_2 obtained from B_2 by adding a block $E(n_0) \otimes \mathbf{b}^T$ in the second block row (and a zero block of size $n_{j-1} r \times n_0$ above it). Resulting matrix B'_2 satisfies the conditions of Lemma 5, which gives $\mathrm{rank}\, B'_2 = \mathrm{rank}\, B_2 + (n_0 - \mathrm{rank}\, A_1)$; indeed, whenever $k_0 > 0$, we can find a codeword of weight δ.

Part *(c)*: Again, we only need to consider the case $k_{j-1} > 0$ and $\kappa > 0$, so that minimum-weight codewords $\mathbf{c} \in \mathcal{Q}(A_{j-1}, A_j^T)$ and $\mathbf{b} \in \mathcal{C}_P^{\perp}$ cab be found; $d_{j-1} = \mathrm{wgt}(\mathbf{c})$, $\delta = \mathrm{wgt}(\mathbf{b})$. The trial codeword has the block form $\mathbf{x} = (0 | \mathbf{c} \otimes \mathbf{b})$, it satisfies $B_j \mathbf{x}^T = 0$. Also, since $\mathbf{c}^T$ is linearly independent from the columns of A_{j-1}, it follows from Lemma 5 that $\mathbf{x}^T$ is linearly independent from the columns of B_{j+1}. Indeed, it is easy to see that if we took instead of the $\mathbf{c}^T$ a linear combination of the columns of A_{j-1}, the additional block would not affect the rank since it may be eliminated by column transformations. If the same were true for the vector $\mathbf{c}^T$ linearly independent from columns of A_{j-1}, we would get a contradiction with Lemma 5. □

Notice that parts (b) and (c) in Theorem 4 can be united if we take $d_0 = 1$ with $k_0 > 0$. In the following, we assume this to be the case.

Theorem 6. *The left minimum distance $d'_j = d(B_j, B_{j+1}^T)$ of the CSS code with generators B_j and B_{j+1}^T satisfies the following lower bound: (i) if $\tilde{\kappa} = 0$, $d'_j \geq d_{j-1}\delta$. (ii) Otherwise, if $\tilde{\kappa} > 0$, $d'_j \geq \min(d_j, d_{j-1}\delta)$.*

The proof relies on the following Lemma, a generalization of a statement used in the proof of the minimum-distance bound for the hypergraph-product codes[10].

Lemma 7. *Consider the matrices B_1 and B_2, such that $B_1 B_2 = 0$, constructed from the matrices $A_1^{(n_0 \times n_1)}$, $A_2^{(n_1 \times n_2)}$, and $P^{(r \times c)}$ as in Eq. (2). Let $I_1 \subseteq \{1, 2, \ldots, n_1\}$ and $I_2 \subseteq \{1, 2, \ldots, c\}$ denote two arbitrary index sets, and the support of each of the vectors $\mathbf{a}_i \in \mathbb{F}_2^{n_1}$, $i \leq r$, and $\mathbf{b}_j \in \mathbb{F}_2^c$, $j \leq n_0$, respectively, be contained inside of I_1 and I_2. Consider $\mathbf{e} = (\sum_{i=1}^r \mathbf{a}_i \otimes \mathbf{y}_i^{(r)} | \sum_{j=1}^{n_0} \mathbf{y}_j^{(n_0)} \otimes \mathbf{b}_j)$ where $\mathbf{y}_i^{(s)} \in \mathbb{F}_2^s$ is a vector with the only non-zero*

element at the position i, such that $B_1\mathbf{e}^T = 0$. Consider matrices B'_1 and B'_2 similarly constructed from A'_1, A'_2, and P', where A'_1 and P' are constructed from A_1 and P by keeping only the columns in I_1 and I_2, respectively, and A'_2 is a generator matrix of the code $\mathcal{C}_{A_2}$ shortened *to I_2 [That is, we take a subcode of $\mathcal{C}_{A_2}$ consisting of codewords with $c_i = 0$ outside of I_2, and* puncture *it at these positions.] Also, define vectors $\mathbf{a}'_i$ and $\mathbf{b}'_j$ by dropping the (all-zero) components outside of I_1 and I_2, respectively, and the corresponding vector $\mathbf{e}'$ which satisfies $G'_x(\mathbf{e}')^T = 0$. With these definitions, if $\mathbf{e}'$ is a linear combination of rows of B'_2, then $\mathbf{e}$ is a linear combination of rows of B_2.*

Proof of Theorem 6. For the reference, we are looking at

$$B_j = \left(\begin{array}{c|c} A_j \otimes E_r & E_{n_{j-1}} \otimes P \\ \hline & A_{j-1} \otimes E_c \end{array}\right), \quad B_{j+1}^T = \left(\begin{array}{c|c} A_{j+1}^T \otimes E_r & \\ \hline E_{n_j} \otimes P^T & A_j^T \otimes E_c \end{array}\right).$$

Consider a two-block vector $\mathbf{e} = (\mathbf{e}_1|\mathbf{e}_2)$, with $\mathbf{e}_1 \in \mathbb{F}_2^{n_j r}$, $\mathbf{e}_2 \in \mathbb{F}_2^{n_{j-1}c}$, where $w_1 \equiv \mathrm{wgt}(\mathbf{e}_1) < d_j$, and $w_2 \equiv \mathrm{wgt}(\mathbf{e}_2) < d_{j-1}\delta$, and assume $B_j\mathbf{e}^T = 0$. We are going to show that $\mathbf{e}$ is a linear combination of rows of B_{j+1}^T by using Lemma 7 twice.

step 1: given $\mathbf{e}_1$, mark the columns in A_j which are incident on non-zero positions in $\mathbf{e}$. Denote the corresponding index set and the submatrix of A_j, respectively, as $I_1 \subset \{1, 2, \ldots, n_j\}$ and A'_j. As in Lemma 7, denote A'_{j+1} the generator matrix of the code $\mathcal{C}_{A_{j+1}}$ shortened at the positions outside of I_1. By assumption, I_1 is an erasable set in $\mathcal{Q}(A_j, A_{j+1}^T)$; this implies that $\mathcal{Q}\big(A'_j, (A'_{j+1})^T\big)$ encodes no qubits. Take $P' = P$, and construct the corresponding matrices B'_1 and B'_2, the shortened vectors $\mathbf{a}'_i$, as well as the corresponding vector $\mathbf{e}' \equiv (\mathbf{e}'_1|\mathbf{e}_2)$ which satisfies $B'_1(\mathbf{e}'_1)^T = 0$. The point of the first reduction is that the code $\mathcal{Q}' = \mathcal{Q}(A'_j, (A'_{j+1})^T)$ encodes no qubits, so that the weight of the first block in $\mathbf{e}$ no longer matters.

Step 2: Consider the representation of the vector

$$\mathbf{e}_2 = \sum_{\ell=1}^{c} \mathbf{f}_\ell^{(n_{j-1})} \otimes \mathbf{y}_\ell^{(c)}, \tag{10}$$

where the assumed identity $B_j(\mathbf{e}')^T = 0$ implies $A_{j-1}\mathbf{f}_\ell^T = 0$ for any $1 \le \ell \le c$. For those ℓ where $\mathbf{f}_\ell$ is linearly dependent with the rows of A_j^T, $\mathbf{f}_\ell = \alpha_\ell A_j^T$, render this vector to zero by the linear transformation

$$\mathbf{e}' \to \mathbf{e}' + (0|\alpha_\ell \otimes \mathbf{y}_\ell^{(c)}) \cdot (B'_{j+1})^T.$$

Such a transformation only affects one vector $\mathbf{f}_\ell$. The resulting vector $\bar{\mathbf{e}}' = (\mathbf{e}'_1|\mathbf{e}'_2)$ has the second block of weight $\mathrm{wgt}(\mathbf{e}'_2) \le \mathrm{wgt}(\mathbf{e}_2) < d_{j-1}\delta$, it satisfies $B'_j(\bar{\mathbf{e}}')^T = 0$, and in the corresponding block representation (10) the remaining non-zero vectors $\mathbf{f}_\ell^{(n_{j-1})}$ all have weights d_{j-1} or larger.

This means that, for sure, there remains fewer than δ of these non-zero vectors $\mathbf{f}_\ell$. Therefore, in the representation $\mathbf{e}'_2 = \sum_{j=1}^{n_0} \mathbf{y}_j^{(n_0)} \otimes \mathbf{b}'_j$ compatible with Lemma 7, the union of supports of vectors $\mathbf{b}'_j$, I'_2, has cardinality $|I'_2| < \delta$. Indeed, I'_2 is just the set of the indices ℓ corresponding to the remaining non-zero vectors $\mathbf{f}_\ell^{(n_{j-1})}$.

Finally, in step 3, trim the columns of P, keeping only the positions inside I'_2. Since there are fewer then δ columns left, $c' = |I'_2| < \delta$, the resulting classical code contains no non-zero vectors, $c' = \mathrm{rank}\, P'$. Now, after we trimmed the columns of both A_j and of P, according to

Eq. (6), $\mathcal{Q}\big(B''_j, \check{(}B_{j+1})^T\big)$ encodes no qubits; thus the corresponding vector $\mathbf{e}''$ which satisfies $B''_j(\mathbf{e}'')^T = 0$, is a linear combination of the rows of $(B''_{j+1})^T$.

We can now use Lemma 7 to show that the vector $\bar{\mathbf{e}}'$ is a linear combination of the rows of $(B'_{j+1})^T$; this remains true for the vector $\mathbf{e}'$. Using Lemma 7 again we see that the original two-block vector $\mathbf{e}$ with the block weights $w_1 < d_j$ and $w_2 < d_{j-1}\delta$ which satisfies $B_j\mathbf{e}^T = 0$ is necessarily a linear combination of the rows of B^T_{j+1}. This guarantees $d'_j \geq \min(d_j, d_{j-1}\delta)$.

To complete the proof, consider the case $\tilde{\kappa} = 0$ separately. Here, step 1 can be omitted; the code resulting from steps 2 and 3 alone would encode no qubits, regardless of the weight $\mathrm{wgt}(\mathbf{e}_1)$ of the first block. Thus, in this case we get the lower bound $d'_j \geq d_{j-1}\delta$. □

V. EXPLICIT CODE PARAMETES

Consider the special case of a four-dimensional construction, from four matrices P_j, $j \in \{1, 2, 3, 4\}$, of dimension $r_j \times c_j$, with rank $u_j > 0$, so that the binary codes with the check matrices P_j and P'_j have distances δ_j and $\tilde{\delta}_j$, respectively. The parameters of thus constructed codes are given in the Table I on page 9

VI. ROW AND COLUMN WEIGHT OF CSS CODE

In many quantum error models, the probability of measurement error is related to the number of gate operations or qubits involved in each measurement. The row and column weight of the matrices in the complex chain can be used to determine upper and lower bound of the weight of check operators, and so the measurement error probability. It can easily be derived from the parameters of P.

Let ω_j be the maximum weight of rows in P_j, and $\tilde{\omega}_j$ be the maximum weight of columns in P_j. The maximun row weight of $A_j^{(m)}$ would be $\omega_j^{(m)} = \max(\omega_j^{(m-1)} + \omega_m, \omega_{j-1}^{(m-1)})$, and the maximun column weight $\tilde{\omega}_j^{(m)} = \max(\tilde{\omega}_{j-1}^{(m-1)} + \tilde{\omega}_m, \tilde{\omega}_j^{(m-1)})$ for $j = 1, ..., m$. Let $\omega_0^{(m)} = \omega_{m+1}^{(m)} = \tilde{\omega}_0^{(m)} = \tilde{\omega}_{m+1}^{(m)} = 0$ to complete the rules.

Apply the iteration rule, one can get the general expresion: $\omega_j^{(m)} = \max_{\{I^{(m-j+1)}\}} S^{(m)}(\{I^{(m-j+1)}\})$, $\tilde{\omega}_j^{(m)} = \max_{\{I^{(j)}\}} \tilde{S}^{(m)}(\{I^{(j)}\})$. The j-term summation is given by $S^{(m)}(\{I^{(j)}\}) = \sum_{i=1}^{j} \omega^{(m)}_{I_i^{(j)}}$, $\tilde{S}^{(m)}(\{I^{(j)}\}) = \sum_{i=1}^{j} \tilde{\omega}^{(m)}_{I_i^{(j)}}$, where $\{I^{(j)}\}$ is an arbitrary size-j subset of the index set $\{1, 2, ..., m\}$

For the CSS code $\mathcal{Q}^{(m)}(A_j, A^T_{j+1})$, the max row weight is $\boxed{\max(\omega_j^{(m)}, \tilde{\omega}_{j+1}^{(m)})}$, which is the upper bound on weight of check operators. And the max column weight is $\max(\tilde{\omega}_j^{(m)}, \omega_{j+1}^{(m)})$.

In order to find the min weight ω', replace max with min and the same rules will apply. The difference is, when finding max weight, combination of rows or columns are not allowed. But when finding min weight, one can combine row or columns to get smaller weight. The reason is that we use the former to find weight but use the latter to determine the distance.

m	$n_0^{(m)} = n_0^{(m-1)} r_m$	$n_1^{(m)} = n_1^{(m-1)} r_m + n_0^{(m-1)} c_m$	$n_2^{(m)} = n_2^{(m-1)} r_m + n_1^{(m-1)} c_m$	n_3	n_4
1	r_1	c_1			
2	$r_1 r_2$	$r_1 c_2 + c_1 r_2$	$c_1 c_2$		
3	$r_1 r_2 r_3$	$r_1 r_2 c_3 + r_1 c_2 r_3 + c_1 r_2 r_3$	$r_1 c_2 c_3 + c_1 r_2 c_3 + c_1 c_2 r_3$	$c_1 c_2 c_3$	
4	$r_1 r_2 r_3 r_4$	$r_1 r_2 r_3 c_4 + r_1 r_2 c_3 r_4$ $r_1 c_2 r_3 r_4 + c_1 r_2 r_3 r_4$	$c_1 r_2 r_3 c_4 + c_1 r_2 c_3 r_4 + c_1 c_2 r_3 r_4$ $r_1 c_2 r_3 c_4 + r_1 c_2 c_3 r_4 + r_1 r_2 c_3 c_4$	$r_1 c_2 c_3 c_4 + c_1 r_2 c_3 c_4$ $+ c_1 c_2 r_3 c_4 + c_1 c_2 c_3 r_4$	$c_1 c_2 c_3 c_4$
m	$k_0^{(m)} = k_0^{(m-1)} \tilde{\kappa}_m$	$k_1 = k_0^{(m-1)} \kappa_m + k_1^{(m-1)} \tilde{\kappa}_m$	k_2	k_3	k_4
1	$\tilde{\kappa}_1$	κ_1			
2	$\tilde{\kappa}_1 \tilde{\kappa}_2$	$\tilde{\kappa}_1 \kappa_2 + \kappa_1 \tilde{\kappa}_2$	$\kappa_1 \kappa_2$		
3	$\tilde{\kappa}_1 \tilde{\kappa}_2 \tilde{\kappa}_3$	$\kappa_1 \tilde{\kappa}_2 \tilde{\kappa}_3 + \tilde{\kappa}_1 \kappa_2 \tilde{\kappa}_3 + \tilde{\kappa}_1 \tilde{\kappa}_2 \kappa_3$	$\kappa_1 \kappa_2 \tilde{\kappa}_3 + \kappa_1 \tilde{\kappa}_2 \kappa_3 + \tilde{\kappa}_1 \kappa_2 \kappa_3$	$\kappa_1 \kappa_2 \kappa_3$	
4	$\tilde{\kappa}_1 \tilde{\kappa}_2 \tilde{\kappa}_3 \tilde{\kappa}_4$	$\kappa_1 \tilde{\kappa}_2 \tilde{\kappa}_3 \tilde{\kappa}_4 + \tilde{\kappa}_1 \kappa_2 \tilde{\kappa}_3 \tilde{\kappa}_4$ $+ \tilde{\kappa}_1 \tilde{\kappa}_2 \kappa_3 \tilde{\kappa}_4 + \tilde{\kappa}_1 \tilde{\kappa}_2 \tilde{\kappa}_3 \kappa_4$	$\kappa_1 \kappa_2 \tilde{\kappa}_3 \tilde{\kappa}_4 + \kappa_1 \tilde{\kappa}_2 \kappa_3 \tilde{\kappa}_4 + \kappa_1 \tilde{\kappa}_2 \tilde{\kappa}_3 \kappa_4$ $+ \tilde{\kappa}_1 \kappa_2 \kappa_3 \tilde{\kappa}_4 + \tilde{\kappa}_1 \kappa_2 \tilde{\kappa}_3 \kappa_4 + \tilde{\kappa}_1 \tilde{\kappa}_2 \kappa_3 \kappa_4$	$\kappa_1 \kappa_2 \kappa_3 \tilde{\kappa}_4 + \kappa_1 \kappa_2 \tilde{\kappa}_3 \kappa_4$ $+ \kappa_1 \tilde{\kappa}_2 \kappa_3 \kappa_4 + \tilde{\kappa}_1 \kappa_2 \kappa_3 \kappa_4$	$\kappa_1 \kappa_2 \kappa_3 \kappa_4$
	d_0	d_1	d_2	d_3	d_4
1	$\tilde{\kappa}_1 > 0 ? 1 : \infty$	δ_1			
2	$\tilde{\kappa}_2 > 0 ? d_0^{(m=1)} : \infty$	$\tilde{\kappa}_2 > 0 ? \min\big(d_1^{(1)}, d_0^{(1)} \delta_2\big) : d_0^{(1)} \delta_2$ $\tilde{\kappa}_1 \tilde{\kappa}_2 > 0 : \ \min(\delta_1, \delta_2)$ $\tilde{\kappa}_1 = 0 : \ \delta_2$ $\tilde{\kappa}_2 = 0 : \ \delta_1$	$\delta_1 \delta_2$		
3	$\tilde{\kappa}_3 > 0 ? d_0^{(m=2)} : \infty$	$\tilde{\kappa}_3 > 0 : \ \min(d_1^{(m=2)}, d_0^{(m=2)} \delta_3)$ $\tilde{\kappa}_3 = 0 : \ d_0^{(m=2)} \delta_3$	$\tilde{\kappa}_3 > 0 : \ \min(d_2^{(m=2)}, d_1^{(m=2)} \delta_3)$ $\tilde{\kappa}_3 = 0 : \ d_1^{(m=2)} \delta_3$	$\delta_1 \delta_2 \delta_3$	
4	$\tilde{\kappa}_4 > 0 ? d_0^{(m=3)} : \infty$	$\tilde{\kappa}_4 > 0 : \ \min(d_1^{(m=3)}, d_0^{(m=3)} \delta_4)$ $\tilde{\kappa}_4 = 0 : \ d_0^{(m=3)} \delta_4$	$\ldots$	$\ldots$	$\delta_1 \delta_2 \delta_3 \delta_4$

TABLE I. Parameters of the first four chain complexes. The distance is minimized over all non-zero combinations of the products of δ_j and $\tilde{\delta}_j$ corresponding to the products of κ_j and $\tilde{\kappa}_j$ that actually contribute to the total k.

row weight	$\omega_1^{(m)}$	$\omega_2^{(m)}$	$\omega_3^{(m)}$	$\omega_4^{(m)}$
1	ω_1			
2	$\omega_1+\omega_2$	$\max(\omega_1,\omega_2)$		
3	$\omega_1+\omega_2+\omega_3$	$\max(\omega_1\omega_2,\omega_1\omega_3,\omega_2\omega_3)$	$\max(\omega_1,\omega_2,\omega_3)$	
4	$\omega_1+\omega_2+\omega_3+\omega_4$	$\max(\omega_1+\omega_2+\omega_4,$ $\omega_1+\omega_3+\omega_4,$ $\omega_2+\omega_3+\omega_4,$ $\omega_1+\omega_2+\omega_3)$	$\max(\omega_1+\omega_2,\omega_1+\omega_3,$ $\omega_2+\omega_3,\omega_1+\omega_4,$ $\omega_2+\omega_4,\omega_3+\omega_4)$	$\max(\omega_1,\omega_2,\omega_3,\omega_4)$
col weight	$\tilde{\omega}_1^{(m)}$	$\tilde{\omega}_2^{(m)}$	$\tilde{\omega}_3^{(m)}$	$\tilde{\omega}_4^{(m)}$
1	$\tilde{\omega}_1$			
2	$\max(\tilde{\omega}_1,\tilde{\omega}_2)$	$\tilde{\omega}_1+\tilde{\omega}_2$		
3	$\max(\tilde{\omega}_1,\tilde{\omega}_2,\tilde{\omega}_3)$	$\max(\tilde{\omega}_1\tilde{\omega}_2,\tilde{\omega}_1\tilde{\omega}_3,\tilde{\omega}_2\tilde{\omega}_3)$	$\tilde{\omega}_1+\tilde{\omega}_2+\tilde{\omega}_3$	
4	$\max(\tilde{\omega}_1,\tilde{\omega}_2,\tilde{\omega}_3,\tilde{\omega}_4)$	$\max(\tilde{\omega}_1+\tilde{\omega}_2,\tilde{\omega}_1+\tilde{\omega}_3,$ $\tilde{\omega}_2+\tilde{\omega}_3,\tilde{\omega}_1+\tilde{\omega}_4,$ $\tilde{\omega}_2+\tilde{\omega}_4,\tilde{\omega}_3+\tilde{\omega}_4)$	$\max(\tilde{\omega}_1+\tilde{\omega}_2+\tilde{\omega}_4,$ $\tilde{\omega}_1+\tilde{\omega}_3+\tilde{\omega}_4,$ $\tilde{\omega}_2+\tilde{\omega}_3+\tilde{\omega}_4,$ $\tilde{\omega}_1+\tilde{\omega}_2+\tilde{\omega}_3)$	$\tilde{\omega}_1+\tilde{\omega}_2+\tilde{\omega}_3+\tilde{\omega}_4$

TABLE II. upper bound on row and column weight of matrices $A_j^{(m)}$ in the chain complex. ω_i and $\tilde{\omega}_i$ are the row and column weight of P_i. To get the lower bound $\omega_i'^{(m)}$ and $\tilde{\omega}_i'^{(m)}$, just replace max by min

VII. DISTANCE OF DATA SYNDROME CODE

From a quantum code with redundant parity check matrix $G_{n_1\times n_2}$, we define the data syndrome code as $Q=(G|E_{n_1})$. There exists full rank matrix $T_{\kappa\times n_1}$ such that $TG=0$. T is the parity check matrix for the classical syndrome code. Then the generating matrix for syndrome code is F such that $TF^T=0$. One can see F is just the full rank matrix of G^T. Then the minimum row weight of F is just the minimum weight of G^T, which is the distance of the syndrome code d_s. Then $[n_1,k,d_s]$ is the parameters of syndrome code with parity check matrix T. $k=n_1-\kappa$ is the rank of G.

Apply it to code $\mathcal{Q}^{(m)}(A_j,A_{j+1}^T)$, I will get $\boxed{[n_{j-1}^{(m)},\mathrm{rank}(A_j),\tilde{\omega}_j'^{(m)}]}$ for the left part, here prime mean minimum instead of maximum. For the right part, the syndrome code has parameters $\boxed{[n_{j+1}^{(m)},\mathrm{rank}(A_{j+1}),\omega_{j+1}'^{(m)}]}$. Combine them, I will get $\boxed{[n_{j-1}^{(m)}+n_{j+1}^{(m)},n_j^{(m)}-k_j^{(m)},\min(\tilde{\omega}_j'^{(m)},\omega_{j+1}'^{(m)})]}$

To avoid confusion in my notation, the data syndrome code means one decode qubit error and syndrome error globally/simultaneously. If one decode syndrome error first, and then use the decoded syndrome to decode the qubit error, the syndrome part is called syndrome code, and the qubit part is the original quantum code.

Theorem 8. *(1) If the quantum code has distance d_q, and the syndrome code has distance d_s, then the overall code (decode syndrome error first and then qubit error) would have distance* $\boxed{d=\min(d_q,d_s)}$.

(2) If we decode the qubit error and syndrome error simultaneously, the data syndrome code would have distance $\boxed{d_{ds}=\min \mathrm{wgt}\left((E_{n_2}|G^T)\setminus(H|0)\right)\geq\min(d_q,1+d_s)}$*, where H is the other parity check matrix.*

Proof. Part (1) looks obvious.

For part (2), the dual matrix of $Q = (G|E_{n_1})$ is $(E_{n_2}|G^T)$. We can transform it into $\begin{pmatrix} H & 0 \\ C & 0 \\ A & B \end{pmatrix}$, where $GH^T = 0$, $GC^T = 0$, H is the other parity check matrix, C is the codeword generating matrix, A and B are full rank matrices, which refer to detectable qubit errors and corresponding syndrome in the original quantum code. The distance of the data syndrome code is the minimum row weight of $(H|0)$'s coset in $\begin{pmatrix} C & 0 \\ A & B \end{pmatrix}$. We can write the matrix as $(E|G^T) \setminus (H|0)$. Use $\min \mathrm{wgt}(C|0) = \min \mathrm{wgt}(C) = d_q$ and

$$\min \mathrm{wgt}(A|B) = \min \mathrm{wgt}\left((E|G^T) \setminus \begin{pmatrix} H & 0 \\ C & 0 \end{pmatrix}\right) \geq 1 + \min \mathrm{wgt}(G^T) = 1 + d_s$$

The we have the following lower bound

$$d_{ds} = \min\left(d_q, \min \mathrm{wgt}(A|B)\right) \geq \min(d_q, 1 + d_s)$$

□

Example: For the cubic code with side length L, the G_X with plaquette check operators has $d_q = L^2$ and $d_s = 4$. In part (1), it will lower the overall distance to 4. In part (2), the distance becomes 5. Both match the lower bounds exactly. And both become very weak, compared to the quantum code distance. Then this kind of code only apply to the case where measurement error probability are much lower than quantum error probability.

For the $(A|B)$ part, we consider it as bad/logical errors. But the measurement error will disappear in the next round and then the qubit error can get detected. Then d_{ds} would be higher than the lower bound especially when $d_q \gg d_s$.

Then we can get a better lower bound by looking at the row weight distribution of $(E|G^T)$

Definition 9 (robustness)**.** *define $g(x)$ as the min weight of error e_q s.t. $|\sigma(e_q)| = x$. We call it robustness. Then* $\min \mathrm{wgt}\left((E|G^T) \setminus \begin{pmatrix} H & 0 \\ C & 0 \end{pmatrix}\right) = \min_x(x + g(x)) = d_r$,

$\boxed{d_{ds} = \min(d_q, d_r)}$

In order to get large distance, we want $g(x)$ to be large. Also, we prefer $g(x)$ to have an inverse relation with x, such as $g(x) = d_q - x$. **This gives a standard to choose a set of parity check generators, that maximize the distance of the data syndrome code.** Fujiwara[11] gives several examples of this. Increasing robustness $g(x)$ is the generalization of those examples, and is a special method to increase the column weight of G. I believe we can design an optimization algorithm by calculating the row weight distribution of G^T, to make the code have best distance. The output would be a transformation matrix for the rows of G.

As for the *soundness* defined by Earl, he wants make $g(x)$ as small as possible, to avoid large weight residual errors resulted from measurement error. This applies in the case that one decode syndrome error first and then qubit error. This reduces the decoding complexity.

But If we do a global decoding for syndrome error and qubit error together, It is not necessary to have good soundness. Conversely, large minimum weight are prefered to increase the distance.

As Earl suggested, if the residual error has weight $\mathrm{wgt}(e_{res}) < d_q/2$, we consider it as a good error, which can hopefully get fixed in the next round. Then we can have a modified definition of the robustness, which increases the lower bound compare to the previous definition.

Definition 10 (modified robustness)**.** *define $g'(x)$ as the min weight of error e_q s.t. $|\sigma(e_q)| = x$ and $\mathrm{wgt}(e_q) \geq d_q/2$. If such error doesn't exist, then $g(x) = \infty$. Then* $\min \mathrm{wgt}\left((E|G^T) \setminus \begin{pmatrix} H & 0 \\ C & 0 \end{pmatrix}\right) = \min_x(x + g'(x)) = d'_r \geq 1 + d_q/2$, $d_{ds} = \min(d_q, d'_r) \geq 1 + d_q/2$

VIII. GENERALIZATION TO NON CSS CODE

This section generalizes the robustness theory to non CSS code, and show an example of optimizing the $[[7, 1, 3]]$ CSS code. For a general code with parity check matrix $G = (G_X|G_Z)$, $\tilde{G}G^T = 0$. C is the codeword generating matrix $\tilde{G}C^T = 0$. Define the data syndrome code as $D = (G_X|G_Z|E)$. An error $e = (e_X|e_Z|e_s)$ is a codeword iff $D(e_Z|e_X|e_s)^T = 0$. The dual matrix of D is $J = \begin{pmatrix} E & G_X^T \\ & G_Z^T \end{pmatrix}$. The codeword generating matrix of D is $J \setminus \begin{pmatrix} C_X & C_Z & 0 \\ G_X & G_Z & 0 \end{pmatrix}$. The row weight (supposition of X and Z error on a qubit is a Y error of weight 1) of the codeword generating matrix give the distance of the code.

Fujiwara modify the $[[7, 1, 3]]$ CSS code to make it a non CSS code which can fix single measurement error[11] . The parameters of the quantum code doesn't change. The original CSS code has parity check matrix

$$(G_X|G_Z) = \left(\begin{array}{ccccccc|ccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 & & & & & & & \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & & & & & & & \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & & & & & & & \\ & & & & & & & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ & & & & & & & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & & & & & & & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right)$$

Transpose of the dual matrix is

$$\begin{pmatrix} E & G_X^T \\ & G_Z^T \end{pmatrix}^T = \left(\begin{array}{ccccccc|ccccccc} 1&0&0&1&0&1&1 & & & & & & & \\ 0&1&0&1&1&0&1 & & & & & & & \\ 0&0&1&0&1&1&1 & & & & & & & \\ & & & & & & & 1&0&0&1&0&1&1 \\ & & & & & & & 0&1&0&1&1&0&1 \\ & & & & & & & 0&0&1&0&1&1&1 \\ \hline 1& & & & & & & & & & & & & \\ &1& & & & & & & & & & & & \\ & &1& & & & & & & & & & & \\ & & &1& & & & & & & & & & \\ & & & &1& & & & & & & & & \\ & & & & &1& & & & & & & & \\ & & & & & &1& & & & & & & \\ \hline & & & & & & & 1& & & & & & \\ & & & & & & & &1& & & & & \\ & & & & & & & & &1& & & & \\ & & & & & & & & & &1& & & \\ & & & & & & & & & & &1& & \\ & & & & & & & & & & & &1& \\ & & & & & & & & & & & & &1 \end{array}\right)$$

Eliminating the parity check operators, the min column weight is 2. Then the distance of the syndrome code is $d = 2$. It is not able to fix single measurement error.

Fujiwara apply the recombination of parity check operators, the paritycheck matrix becomes

$$(G'_X|G'_Z) = \left(\begin{array}{ccccccc|ccccccc} 1&0&0&1&0&1&1 & 1&0&0&1&0&1&1 \\ 0&1&0&1&1&0&1 & 1&0&0&1&0&1&1 \\ 0&0&1&0&1&1&1 & 1&0&0&1&0&1&1 \\ 1&1&1&0&0&0&1 & 0&1&1&1&0&1&0 \\ 1&1&1&0&0&0&1 & 1&0&1&1&1&0&0 \\ 1&1&1&0&0&0&1 & 1&1&0&0&1&1&0 \end{array}\right)$$

Then the transpose of its dual becomes

$$\begin{pmatrix} E & G_X'^T \\ & G_Z'^T \end{pmatrix}^T = \left(\begin{array}{ccccccc|ccccccc} 1&0&0&1&0&1&1&1&0&0&1&0&1&1 \\ 0&1&0&1&1&0&1&1&0&0&1&0&1&1 \\ 0&0&1&0&1&1&1&1&0&0&1&0&1&1 \\ 1&1&1&0&0&0&1&0&1&1&1&0&1&0 \\ 1&1&1&0&0&0&1&1&0&1&1&1&0&0 \\ 1&1&1&0&0&0&1&1&1&0&0&1&1&0 \\ \hline 1&&&&&&&&&&&&&\\ &1&&&&&&&&&&&&\\ &&1&&&&&&&&&&&\\ &&&1&&&&&&&&&&\\ &&&&1&&&&&&&&&\\ &&&&&1&&&&&&&&\\ &&&&&&1&&&&&&&\\ \hline &&&&&&&1&&&&&&\\ &&&&&&&&1&&&&&\\ &&&&&&&&&1&&&&\\ &&&&&&&&&&1&&&\\ &&&&&&&&&&&1&&\\ &&&&&&&&&&&&1&\\ &&&&&&&&&&&&&1 \end{array}\right)$$

Eliminating the parity check operators, the min column weight is 3. Then the distance of the syndrome code is $d = 3$. It is able to fix all single measurement error.

To see the change from the view of robustness, lets calculate it for both codes.

For the original CSS code, the first 3 columns show there exists a single qubit error for that syndrome bit. So $g(1) = 1$. The column 4 show that we get a single qubit error corresponding to 2 syndrome bit, then $g(2) = 1$, etc. Then $d = 1 + g(1) = 2$.

For the modified code, adding column 1, 2, 7, we got $g(1) = 3$. The column 4 shows $g'(1) = 2$. Column 4 and column 5 show that $g'(2) = 1$, etc. Then $d' = 1+g'(1) = 2+g'(2) = 3$.

Fujiwara optimize the code in a way that all single errors (both qubit and syndrome bit) has distinguished syndrome result. In our language, we are improving the robustness $\min_x(x + g(x)$ from 2 to 3. When designing program, this could be achieved by applying random/particular transform matrix on G and evaluate the robustness. I think there is a way to design such transform matrix, in order to eliminate those columns that give the minimum value of $(x + g(x))$

IX. EXAMPLE: REPETITION CODE AND 2D TORIC CODE

For the 1D repetition code without redundancy, the parity check matrix is $\hat{a} = \begin{pmatrix} 1&1&&\\ &1&1&\\ &&1&1\\ &&&1&1 \end{pmatrix}$, the first and last column gives $g(1) = 1$, and other columns give $g(2) = 1$, which implement $d_{ds} = \min_x(x + g(x)) = 2$. If we add one redundancy to make it the circulant repetition

matrix $a = \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix}$, then we still have $g(2) = 1$, but $g(1) = \infty$. Now $d_{ds} = 3$.

Similarly, for the 2D toric code without redundancy, for most single qubit error, there are two plaquettes connected, then $g(2) = 1$. But for the one plaquette removed by redundancy, the single qubit error on one of the bonds of that plaquette is connected with only one plaquette. Then $g(1) = 1$. Hence $d_{ds} = 2$. But if we add that removed plaquette, then all single error are connected with two plaquettes. Then $g(2) = 1$, $g(1) = \infty$, and $d_{ds} = 3$

In these two cases, we add extra checks to increase d_{ds}. But in some cases, we only need to do recombination of checks. For example $a' = \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & 1 & & 1 & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix}$, where I add row 2 and row 3 in the circulant repetition matrix a. Column 3 in a' shows $g(1) = 1$ and $d_{ds} = 2$. By converting it back to code a, the distance get improved.

These examples show that, if a data syndrome code has only a small number of codewords that have weight matches d_{ds}, then we may eliminate those codewords and increase d_{ds}, by paying a small cost on increasing number of parity checks and weight of parity checks. This case maybe popular when we construct hypergraph product code from random matrices.

For the 2D toric code with redundancy that have $d_{ds} = 3$, since all single qubit error connected with two plaquette match this minimum weight, it is much harder to increase d_{ds}. Lets try an example. Let

$b = a^T = \begin{pmatrix} 1 & & & 1 \\ 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \\ & & & 1 & 1 \end{pmatrix}$, $T = \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ 1 & & & & 1 \\ 1 & & & & \end{pmatrix}$.

The 2D toric code with one redundancy is defined as $G = (a \otimes E | e \otimes b)$. Then define a new code $G' = \begin{pmatrix} (T \otimes E)G \\ (E \otimes T)G \end{pmatrix}$, where we combine any two adjacent plaquettes to form a rectangular check operator. The top matrix means horizontal rectangles, and the bottom matrix means vertical rectangles. Now, each single bond is connected to 6 rectangle, $g(6) = 1$. Each two bonds could connect to 8 rectangles, $g(8) = 2$. Then $d_{ds} = 7$. Here we double the number of parity checks. As a comparison, if we just repeat the measurement twice, then a single qubit error will be connected to 4 plaquettes. Hence $g(4) = 1$ and $d_{ds} = 5$.

---

[1] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," J. Math. Phys. **43**, 4452 (2002).

[2] A. A. Kovalev and L. P. Pryadko, "Fault tolerance of quantum low-density parity check codes with sublinear distance scaling," Phys. Rev. A **87**, 020304(R) (2013).

[3] Earl T. Campbell, "A theory of single-shot error correction for adversarial noise," (2018), unpublished, 1805.09271.

[4] Lev Ioffe and Marc Mézard, "Asymmetric quantum error-correcting codes," Phys. Rev. A **75**, 032345 (2007).

[5] Z. W. E. Evans, A. M. Stephens, J. H. Cole, and L. C. L. Hollenberg, "Error correction optimisation in the presence of x/z asymmetry," (2007), arXiv:0709.3875.

[6] Ashley M. Stephens, Zachary W. E. Evans, Simon J. Devitt, , and Lloyd C. L. Hollenberg, "Asymmetric quantum error correction via code conversion," Phys. Rev. A **77**, 062335 (2008).

[7] Panos Aliferis and John Preskill, "Fault-tolerant quantum computation against biased noise," Phys. Rev. A **78**, 052331 (2008).

[8] Pradeep Kiran Sarvepalli, Andreas Klappenecker, and Martin Rötteler, "Asymmetric quantum codes: constructions, bounds and performance," Proc. R. Soc. A **465**, 1645–72 (2009).

[9] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia, "Ultrahigh error threshold for surface codes with biased noise," Phys. Rev. Lett. **120**, 050505 (2018).

[10] J.-P. Tillich and G. Zemor, "Quantum LDPC codes with positive rate and minimum distance proportional to $\sqrt{n}$," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* (2009) pp. 799–803.

[11] Yuichiro Fujiwara, "Ability of stabilizer quantum error correction to protect itself from its own imperfection," Physical Review A **90**, 062304 (2014).