Trapping Set Analysis of Finite-Length Quantum LDPC Codes

Nithin Raveendran* and Bane Vasić†
Department of Electrical and Computer Engineering
Center for Quantum Networks
University of Arizona, Tucson, AZ 85721
Email: *nithin@email.arizona.edu, †vasic@ece.arizona.edu

***Abstract*—Iterative decoders for finite length quantum low-density parity-check (QLDPC) codes are impacted by short cycles, detrimental graphical configurations known as trapping sets (TSs) present in a code graph as well as symmetric degeneracy of errors. In this paper, we develop a systematic methodology by which quantum trapping sets (QTSs) can be defined and categorized according to their topological structure. Conventional definition of a TS from classical error correction is generalized to address the syndrome decoding scenario for QLDPC codes. We show that QTS information can be used to design better QLDPC code and decoder. For certain finite-length QLDPC codes, frame error rate improvements of two orders of magnitude in the error floor regime are demonstrated without needing any post-processing steps.**

I. Introduction

Quantum low-density parity check (QLDPC) codes are an important class of quantum error correction (QEC) [2], [3] codes that can realize scalable fault-tolerant quantum computers (FTQCs) with a finite multiplicative overhead [4]. In addition, they have finite asymptotic rates with non-zero fault-tolerant thresholds [5], and support low-complexity iterative decoding. However, QLDPC codes implemented in practical QEC systems will be of finite length and will exhibit performance degradation due to the failure of iterative decoders to converge to a correct error pattern. This phenomenon specific to finite-length codes is well understood in classical literature, but its analysis and precise mathematical characterization in the QEC literature [6], [7] is our primary focus in this paper.

Error floor is attributed to the presence of specific topologies of sub-graphs in the Tanner graph generically referred to as *trapping sets* (TSs) [8] that are detrimental to iterative decoders. A typical approach in QEC literature to reduce the error floor is to couple iterative decoders with *ordered statistics decoding (OSD)* and post-processing [9], [10]. In contrast, understanding the failure configurations of classical LDPC codes using TS research allows to develop message-passing decoders which do not require a post-processing step to achieve strong error correction [11]. Such researches on failure configurations of QLDPC codes are relatively unknown. One major drawback of belief propagation (BP) as pointed out in [12] is that the decoding ability of BP is typically limited by the row weight of the parity-check matrix due to the stabilizer commutativity/symplectic inner product (SIP) constraint. Ref. [12] identifies pseudo-codeword structures for cycle codes and proposes modifications to the BP algorithm. However, the generalization from cycle codes to QLDPC codes is nontrivial.

This work is funded by the NSF under grants CCF-1855879 and NSF-ERC 1941583. An extended version of this paper is accessible at: https://arxiv.org/pdf/2012.15297.pdf [1].

For finite-length QLDPC codes, the SIP constraint for the parity check matrices introduces additional code construction constraints resulting in unavoidable cycles in the Tanner graph. Furthermore, QLDPC codes are known to be highly degenerate, i.e., their minimum distance is higher than the weight of their stabilizers. From the decoder perspective, this implies that the decoders need to account for degenerate errors, which has no equivalent in classical error correction. However, iterative algorithms based on BP are sub-optimal in the presence of cycles and, also are not capable of correcting all degenerate errors [6], [13]. Moreover, the inability to directly measure qubits for error correction requires iterative message-passing algorithms to make use of the syndrome information to infer the error introduced by the channel. How the classical trapping sets definition accommodates a syndrome-based decoder is not clearly understood. The approach presented in this paper accounts for these key differences and their implications.

In this paper, we define quantum trapping sets (QTSs) by investigating into failure configurations for syndrome-based iterative message passing algorithms. The quantum trapping set formulation is modified to the syndrome decoding scenario for QLDPC codes considering Pauli X and Z errors separately. We identify QTSs of prominent QLDPC codes and show that the QTSs must be analyzed in conjunction with the particular iterative decoder used along with their location in the Tanner graph [14]. We show that degenerate errors having no classical analogy introduce new failure configurations unique to the QLDPC codes. Message update rules and scheduling strategies are also identified that help the decoder escape from such trapping sets, thereby improving the error floor performance.

The rest of this paper is organized as follows. In Section II, we introduce QLDPC codes using the stabilizer formalism and then discuss the syndrome decoding problem and classical trapping sets. In Section III, we formally define QTSs and describe the methodology used to identify them specifically for Calderbank, Shor, Steane (CSS) codes [15]. Trapping sets of some classes of CSS codes are analyzed in Section IV. Based on these analyses, we discuss the trapping set-aware decoder improvements followed by concluding remarks in Section V.

II. Preliminaries

Let $\mathcal{P}_1 = \{\mathrm{I}, \mathrm{Z}, \mathrm{Y}, \mathrm{X}\}$ be the set of Pauli matrices: $\mathrm{I} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right], \mathrm{X} = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right], \mathrm{Z} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, and $\mathrm{Y} = \imath \mathrm{XZ} = \left[\begin{smallmatrix} 0 & -\imath \\ \imath & 0 \end{smallmatrix}\right]$. Kronecker products of n single-qubit Paulis and scalars $\imath^\kappa$, where $\kappa \in \mathbb{Z}_4 = \{0, 1, 2, 3\}$ form the n-qubit Pauli group $\mathcal{P}_n$. The stabilizer group $\mathcal{S}$ is a commutative subgroup of $\mathcal{P}_n$ that contains only Hermitian Paulis and excludes $-\mathbb{1}$. An $[\![n, k, d]\!]$ quantum stabilizer code [16], [17] is defined as +1 common eigen states of $\mathcal{S}$. Stabilizer group generators $S = \{s_1, s_2, \ldots, s_m\}$ form the $m = n - k$ rows of the corresponding stabilizer matrix with entries from $\mathcal{P}_1$. Using the Pauli-binary isomorphism, an n-qubit Pauli operator can be written as a binary vector of length $2n$ of the form $\boldsymbol{p} = (\boldsymbol{p}_{\mathbf{X}}, \boldsymbol{p}_{\mathbf{Z}})$, where $\boldsymbol{p}_{\mathbf{X}}$ and $\boldsymbol{p}_{\mathbf{Z}}$ are of length n each with ones at positions of X- and Z-Pauli components respectively. Then, the binary representation H_b of the stabilizer matrix of dimension $m \times 2n$ is given by $H_b = \left[H_{\mathrm{X}} \mid H_{\mathrm{Z}}\right]$, where H_{X} and H_{Z} are binary parity check matrices. Although the syndrome decoding paradigm is applicable to any class of quantum codes, the trapping set analysis in this paper is focused on QLDPC families [2], [9], [18] representing the CSS class of codes [15]. An attractive property of CSS codes constructed from two classical codes $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C}_2^{\perp} \subseteq \mathcal{C}_1$, is that the parity check matrix can be written in a separable form: $H_b = \begin{bmatrix} H_{\mathrm{X}} & 0 \\ 0 & H_{\mathrm{Z}} \end{bmatrix}$, where H_{X} and H_{Z} are the parity check matrices of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Readers are referred to [17] for detailed description on stabilizer QEC codes.

A. Decoding Problem

For binary syndrome decoding of CSS codes, we consider the two independent binary symmetric channels (BSCs) rather than the depolarizing channel because it is a simplified model ignoring the correlation between bit flip (X) and phase flip (Z) errors. In this case, the BSCs for X and Z errors have a cross-over probability of $2p/3$ [2], decoded using H_{Z} and H_{X}, respectively. Let $\boldsymbol{e} = (\boldsymbol{e}_{\mathbf{X}}, \boldsymbol{e}_{\mathbf{Z}})$ be the binary representation of a Pauli error acting on the n qubits. The corresponding syndrome is computed as $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_{\mathbf{X}},\ \boldsymbol{\sigma}_{\mathbf{Z}}] = [H_{\mathrm{Z}}.\boldsymbol{e}_{\mathbf{X}}^{\mathrm{T}} (\mathrm{mod}\ 2), H_{\mathrm{X}}.\boldsymbol{e}_{\mathbf{Z}}^{\mathrm{T}} (\mathrm{mod}\ 2)]$. For CSS codes, we can perform error correction for the X and Z errors separately using H_{Z} and H_{X} matrices, respectively. For simplicity going forward, we use H, $\boldsymbol{\sigma}$ and $\boldsymbol{e}$ for the parity check matrix, input syndrome and channel error vector, respectively. A syndrome based decoder's task is to estimate the error pattern $\hat{\boldsymbol{e}}$ whose syndrome $\hat{\boldsymbol{\sigma}}$ matches with the initial input syndrome $\boldsymbol{\sigma}$. If $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}$, the estimated error pattern $\hat{\boldsymbol{e}}$ is applied to reverse the error $\boldsymbol{e}$ introduced by the channel. Error correction process is successful if $\hat{\boldsymbol{e}} = \boldsymbol{e} \oplus h$, where $h \in \mathrm{rowspace}(H)$, i.e., if the stabilized code state is recovered up to a stabilizer ($\hat{\boldsymbol{e}} \oplus \boldsymbol{e}$ is a stabilizer, where $\oplus$ denotes pairwise XOR). Error correction fails when the decoder is unable to find an error pattern that matches the syndrome $\boldsymbol{\sigma}$ or when the decoding process results in a logical or miscorrection error [7].

B. Tanner Graph and Iterative Syndrome Decoder

The stabilizer generator matrix H is the bi-adjacency matrix of a bipartite Tanner graph $G = (V \cup C, E)$, where V represents the set of n qubit/variable nodes (VNs), C is the set of m stabilizer generators/check nodes (CNs) and E is the set of edges between them. CN $c_i \in C$ and VN $v_j \in V$ are neighbors if there is an edge $(v_j, c_i) \in E$ between the nodes, corresponding to the non-zero entry in the matrix, i.e., $H_{c_i, v_j} = 1$. Denote the set of CNs connected to a VN v_j by $\mathcal{N}(v_j)$, and $|\mathcal{N}(v_j)|$, where $|\cdot|$ denotes cardinality, is referred to as the degree of the VN v_j. Similarly, we can define the neighbor set and the degree of a CN c_i as $\mathcal{N}(c_i)$ and $|\mathcal{N}(c_i)|$, respectively. For a subset of VNs, say $K \subseteq V$, $\mathcal{N}(K)$ denotes the set of CN neighbors. The induced subgraph $\mathcal{G}(K)$ is the graph containing the nodes $K \cup \mathcal{N}(K)$ along with the edges $\{(x, y) \in F : x \in K, y \in \mathcal{N}(K)\}$. The girth, g, of the Tanner graph G is the length of the shortest cycle in G. Denote the number of cycles of length g, $g+2, \ldots$ by χ_g, $\chi_{g+2}, \ldots$, respectively. If G has χ_g, $\chi_{g+2}, \ldots$ cycles of length $g, g+2, \ldots$, then the cycle enumerator series $\mathrm{CYC}(x) = \sum_{r \geq 0} \chi_r x^r$ defines the cycle profile of G.

Starting from an input syndrome $\boldsymbol{\sigma}$ and an all-zero error vector estimate, a syndrome based iterative decoder $\mathcal{D}_s$ performs a finite number ℓ_{max} of iterations of message passing decoding over the Tanner graph. Decoder update rules and message alphabet size can be of varying complexity ranging from the simplest binary message passing algorithms such as Gallager-B [19] to MSA or BP using floating point messages [2]. Also, schedule of message passing in $\mathcal{D}_s$ can be implemented with a flooding/parallel schedule or a layered/serial schedule [20]. Trapping set analysis presented here is applicable for all such decoder implementations. Based on the update rules, $\mathcal{D}_s$ outputs an error vector estimate $\hat{\boldsymbol{e}}^{(\ell)} = (\hat{e}_1^{(\ell)}, \hat{e}_2^{(\ell)}, \ldots, \hat{e}_n^{(\ell)})$ and corresponding output syndrome $\hat{\boldsymbol{\sigma}}^{(\ell)} = (\hat{\sigma}_1^{(\ell)}, \hat{\sigma}_2^{(\ell)}, \ldots, \hat{\sigma}_m^{(\ell)})$. We refer to $\hat{e}_j^{(\ell)}/\hat{\sigma}_i^{(\ell)}$ as the value of the variable/check node v_j/c_i at iteration $\ell \leq \ell_{max}$. We conclude that a syndrome based iterative decoder $\mathcal{D}_s$ is successful if the output syndrome $\hat{\boldsymbol{\sigma}}^{(\ell)}$ is equal to the input syndrome $\boldsymbol{\sigma}$ (we also say that syndromes are *matched*). The iterative procedure is halted if successfully matched or if ℓ_{max} number of iterations is reached.

C. Classical Trapping Sets

Extensive works on classical trapping set: enumeration and their harmfulness in terms of their critical number μ and strength s are summarized in [8] and references within. A classical iterative decoder is said to converge correctly if the decoder output word for any $\ell \leq \ell_{max}$, where ℓ denotes the number of iterations, matches to the transmitted codeword and fails otherwise. A variable node v_j is *eventually correct* if there exists a positive integer I_j such that for all iterations $\ell \geq I_{\mathrm{j}}$, the decoder's estimate of v_j is equal to the transmitted bit.

Definition 1 ([8]): A trapping set $\mathcal{T}$ for an iterative decoder $\mathcal{D}$ is a non-empty set of variable nodes in a Tanner graph G that are not eventually correct. Trapping set $\mathcal{T}$ is conventionally labeled as an (a, b) trapping set if the induced sub-graph $\mathcal{G}(\mathcal{T})$ induced has a VNs and b odd degree CNs.

In the next section, we make use of this classical TS definition to accommodate for the degeneracy of QLDPC codes and $\mathcal{D}_s$, iterative syndrome decoder.

III. Quantum Trapping Sets

A. Definition of a Quantum Trapping Set

After pre-defined number of iterations, ℓ_{max}, of iterative syndrome decoding, we declare that the decoder $\mathcal{D}_s$ failed for a particular input syndrome/error pattern if the decoder is not able to find an error pattern with a syndrome equal to the input syndrome. More precisely, a decoder failure is said to have occurred if there does not exist $\ell \leq \ell_{max}$ such that $\text{supp}(\hat{\boldsymbol{\sigma}}^{(\ell)} + \boldsymbol{\sigma}) = \emptyset$, where supp denotes the support set (indices of non-zero elements). During iterative decoding, check node c_i is *eventually satisfied* if there exists a positive integer I_{i} such that for all $\ell \geq I_{\text{i}}$, $\hat{\sigma}_i^{(\ell)} = \sigma_i$. We say that VN v_j, where $1 \leq j \leq n$, has *eventually converged* if there exists a positive integer I_{j} such that for all $\ell \geq I_{\text{j}}$, $\hat{e}_j^{(\ell)} = \hat{e}_j^{(\ell-1)}$. Note that the $\hat{e}_j^{(\ell)}$ is not necessarily the correct estimate of error on the j^{th}-VN. With these definitions, we define quantum TSs as follows:

Definition 2: A trapping set $\mathcal{T}_s$ for a syndrome-based iterative decoder $\mathcal{D}_s$ is a non-empty set of variable nodes in a Tanner graph G that are not eventually converged or are neighbors of the check nodes that are not eventually satisfied.

Remark 1: If the sub-graph $\mathcal{G}(\mathcal{T}_s)$ induced by such a set of variable nodes has a VNs and b unsatisfied CNs, then $\mathcal{T}_s$ is conventionally labeled as an (a, b) trapping set.

The QTSs similar to the TSs in classical LDPC codes have exactly the same definition as the first criterion, and we refer to them as *classical-type* trapping sets [8]. The second class of trapping sets are specifically the harmful degenerate errors observed within the stabilizers classified as *symmetric stabilizer trapping sets*. We will see that in such trapping sets, even though the VNs eventually converge to some error pattern, there exist CNs that are not eventually satisfied. Next these two classes of TSs are discussed assuming $\mathcal{D}_s$ is the well-known Gallager-B decoding algorithm [8], [19] (mostly for pedagogical reasons and simplicity).

1) Classical-type trapping set: We show how classical TSs are also failure configurations of syndrome decoders in the longer version of the paper [1]. In addition to classical-type TSs, iterative decoders on QLDPC codes fail for specific degenerate errors which deserve a closer look as they are specific to QEC and not observed in classical LDPC codes.

2) Symmetric stabilizer trapping set: In quantum decoding, we say error vectors $\boldsymbol{e}$ and $\boldsymbol{f}$ are a pair of *degenerate errors* if $\boldsymbol{e} \oplus \boldsymbol{f}$ is a stabilizer, which makes it equivalent to output any one of the degenerate errors as the candidate error pattern for matching the syndrome. A symmetric topology of the stabilizer sub-graph that contains degenerate error patterns $\boldsymbol{e}$ and $\boldsymbol{f}$ of equal weight will result in an iterative decoding failure. This failure can be attributed to the symmetry of both the stabilizer and the decoder message update rules. Hence, such errors are referred to as symmetric degenerate errors and corresponding sets of VNs as symmetric stabilizer TSs or just symmetric stabilizers.

Definition 3: A symmetric stabilizer is a stabilizer with the set of variable nodes, whose induced sub-graph has no odd-degree check nodes, and that can be partitioned into an even number of disjoint subsets, so that: (a) sub-graphs induced by these subsets of variable nodes are isomorphic, and (b) each subset has the same set of odd degree check node neighbors in its induced sub-graph.

Example 1: Consider the Fig. 1(a) with the stabilizer sub-graph induced by ten VNs that are partitioned into two disjoint sets with the coloring ● and ●. The sub-graphs in Fig. 1(b) and Fig. 1(c) are isomorphic and have the same odd-degree checks represented using dark squares ■. Hence, the stabilizer shown in Fig. 1(a) satisfies the definition of a symmetric stabilizer.

Now, we discuss some degenerate error pairs within the symmetric stabilizer are harmful for iterative decoders. Let the error pattern $\boldsymbol{e}$ be located on the ● VNs in Fig. 1(b). They result in unsatisfied check shown as ■. Note, however, that the stabilizer sub-graph is symmetric, and therefore each erroneous node has a ● twin. The set of all ● twins form an alternative error pattern $\boldsymbol{f}$ in Fig. 1(c). The existing iterative decoders fail as they simultaneously attempt to converge to both these error patterns. For example, during the iterations of the Gallager-B decoder, every unsatisfied CN ■ sends binary message, one back to the VNs. Because of the symmetry, the VNs in both $\boldsymbol{e}$ and $\boldsymbol{f}$ receive exactly the same messages, thus converging to $\boldsymbol{e} \oplus \boldsymbol{f}$, the symmetric stabilizer.

Based on the Definition 2, the set of VNs involved in the symmetric stabilizer form a QTS and the sub-graph in Fig. 1(a) is a (10,0) TS by convention. We can easily prove the following lemma pertaining to the general case.

Lemma 1: A symmetric stabilizer is an $(a, b = 0)$ trapping set, and a is even.

Remark 2: When there are more than a pair (an even number greater than two) of disjoint VN sets, the symmetric stabilizer can be split into smaller symmetric stabilizers.

Notably, the harmfulness of symmetric stabilizers associated with decoders is also different from classical-type TSs, as summarized in the following lemma.

Lemma 2: For an $(a, 0)$ symmetric stabilizer TS with any iterative decoder with a critical number $a/2$, no error pattern on more than $a/2$ nodes of the symmetric stabilizer is a failure configuration.

Proof Consider an $(a, 0)$ symmetric stabilizer with critical number $a/2$ for a syndrome decoder $\mathcal{D}_s$. By the definition of the critical number, any error pattern of weight smaller than the critical number $a/2$ with support on the symmetric stabilizer is corrected by the decoder $\mathcal{D}_s$. Error patterns of weight larger than the critical number $a/2$ with support on

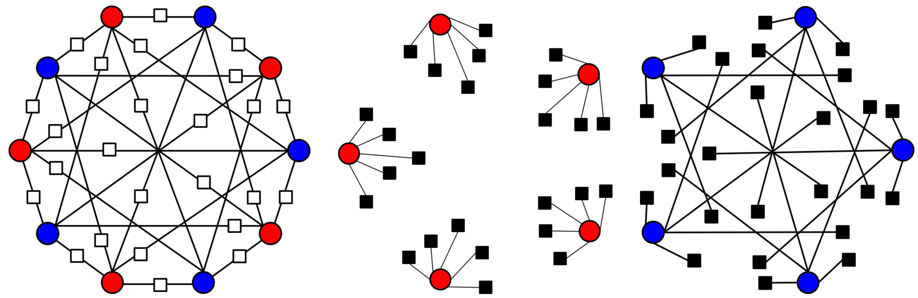


Fig. 1. Tanner graph representation of a $(10, 0)$ symmetric stabilizer trapping set is shown in 1(a). The induced sub-graphs from ● and ● VNs - 1(b) and 1(c), are isomorphic and have the same odd-degree checks represented using dark squares ■.

the symmetric stabilizer are decoded correctly, converging to their respective low-weight degenerate error pattern. □

The strength of a $(a, 0)$ symmetric stabilizer TS with critical number $a/2$ is given by twice the number of possible partitions into two disjoint subsets of VNs that satisfy Definition 3. Each of such partition (distinct by their unsatisfied checks) contributes two error patterns each to the TS decoder failure.

B. Searching for Quantum Trapping Sets

Using the definition of a QTS, one can search for small sub-graphs in the Tanner graph of the QLDPC code to identify and enumerate the QTSs using efficient algorithms for TS search [21], [22]. Unlike the classical-type TSs, the search for symmetric stabilizer TSs requires a different approach of finding low-weight codeword sub-graphs [23] with additional symmetry constraints. In the case of CSS codes, the H_{Z} even-weight stabilizer generators are examples of symmetric stabilizer TSs for iterative decoding over the Tanner graph of H_{X} matrix and vice-versa. After obtaining the list of relevant QTSs, we can perform decoder simulation with an iterative decoder $\mathcal{D}_s$ to verify their relative harmfulness. In the next section, we find and enumerate trapping sets of CSS based QLDPC codes, the generalized bicycle codes [9] and hypergraph product (HP) codes [18] in particular.

IV. Trapping Set Analysis of CSS codes

A. Generalized bicycle codes

Bicycle codes [2] were generalized by Kovalev and Pryadko in [24] as follows: Consider two binary $n/2 \times n/2$ matrices A and B that commute ($AB = BA$). Let $H_{\mathrm{X}} = [A, B]$ and $H_{\mathrm{Z}} = [B^T, A^T]$. The SIP condition is clearly satisfied by definition, and in [24], A and B are chosen as binary circulant matrices so that they commute. In [9], Panteleev and Kalachev use binary polynomials over rings to define the circulant matrices for constructing $[[n, k]]$ family of generalized bicycle codes. The choice of circulant matrices determines the properties of both the classical-type TSs and the symmetric stabilizers in the code. Hence, the observations on QTSs in the following example can be generalized to the code family.

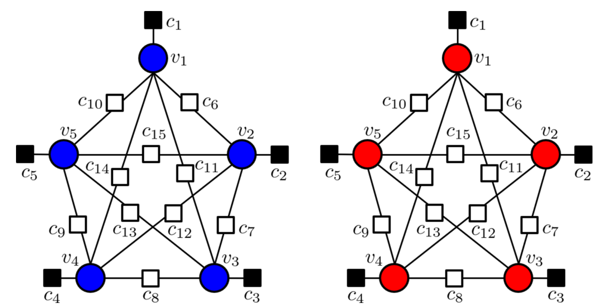


Fig. 2. A $(5, 5)$ TS with 5 variable nodes and 5 odd degree check nodes (the shaded squares represent the odd-degree checks). The blue and red shaded circles indicate erroneous variable nodes in their relative position in the H_{X} matrix: in the circulant matrices A and B, respectively.

Example 2: For the purpose of illustration in our TS analysis, we chose the $A1[[254, 28]]$ code, where the circulant size is 127, $a(x) = 1 + x^{15} + x^{20} + x^{28} + x^{66}$ and $b(x) = 1 + x^{58} + x^{59} + x^{100} + x^{121}$ as given in Appendix B in [9]. The girth of the Tanner graph is six, CN degree $= 10$ and VN degree $= 5$.

1) Classical-type TSs: Based on our QTS definition, we search for QTSs of small size (upto $a = 5$) present in the $A1$ code in Ex. 2. Fig. 2 shows the dense $(5, 5)$ TS present in both the circulant matrices A and B which is the smallest harmful sub-graph present in the Tanner graph. From their cyclic property, we can locate 127 (equal to the circulant size) isomorphic $(5, 5)$ TSs in each of them. In the Fig. 2, blue and red shaded circles for the VNs indicate their relative position in the H_{X} matrix, from A and B respectively.

2) Symmetric stabilizer TSs: The pair of symmetric degenerate error patterns of weight five in the Ex. 2 code shown in Fig. 1(b) and Fig. 1(c) form a $(10, 0)$ TS in Fig. 1(a). Interestingly, the blue and red shaded circles indicates the VNs relative position as before in case of the $(5, 5)$ TS coloring. Also, these error patterns induce isomorphic sub-graphs - trees (without any cycles), quite distinct from the error patterns in classical-type TSs which are usually composed of one or more cycles. Using the cyclic property of the circulant matrices in the code, we can easily locate 127 isomorphic symmetric stabilizers present in the Tanner graph of H_{X}, and similarly

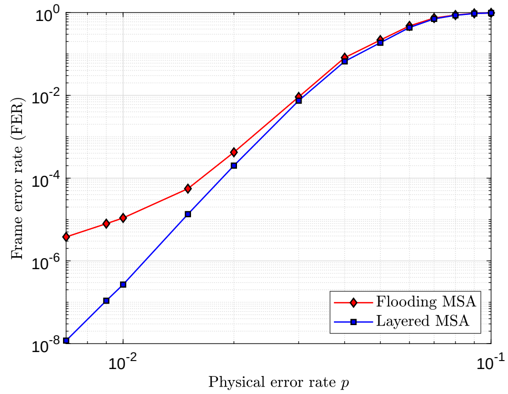


Fig. 3. Figure shows the FER performance comparison for the $A1[[254, 28]]$ code using the min-sum algorithm (MSA) for two different schedules: flooding/parallel and layered schedule. The layered schedule is able to decode all the symmetric stabilizer TSs and numerous classical-type TSs correctly leading to two orders of magnitude improvement in the error floor regime.

for H_{Z}. These symmetric trapping sets have a clear distinction of red and blue nodes with respect to the cyclic matrices A and B, which is used to implement a layered decoder schedule that can break such TSs as shown next.

3) Layered Decoding to break QTSs: The layered schedule employed to break the symmetric stabilizers in the $A1[[254, 28]]$ code is based on the circulant-size of the cyclic matrices A and B given by a straight forward column-update order: $v_1, \ldots, v_{127}$ followed by $v_{128}, \ldots, v_{254}$. The column-layered decoder (MSA with $\ell_{max} = 20$ iterations) is able to decode all the symmetric stabilizer TSs and numerous classical-type TSs correctly leading to two orders of magnitude improvement in the error floor regime (low physical error rates) compared to the flooding MSA decoder as shown in Fig. 3.

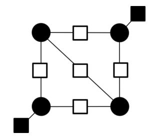
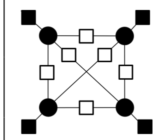
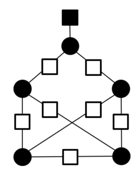
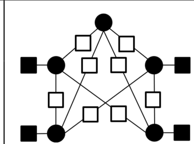
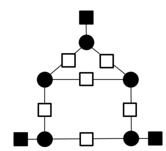
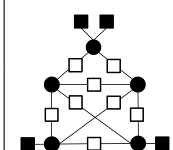
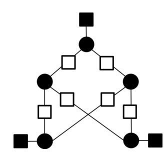
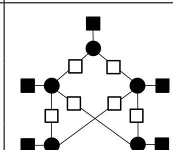
B. Hypergraph product codes

HP codes by Tillich and Zemor [18] and their improvements by Kovalev and Pryadko [25] are constructed by taking Kronecker product (denoted as $\otimes$) of two classical LDPC codes. Using two classical parity check matrices H_1 and H_2 of dimensions $m_1 \times n_1$ and $m_2 \times n_2$ respectively, we have $H_{\mathrm{X}} = \left[\, H_1 \otimes \mathrm{I}_{n_2} \mid \mathrm{I}_{m_1} \otimes H_2^T \,\right]$ and $H_{\mathrm{Z}} = \left[\, \mathrm{I}_{n_1} \otimes H_2 \mid H_1^T \otimes \mathrm{I}_{m_2} \,\right]$.

Example 3: We use the example of a $[[900, 36, 10]]$ HP code given in [10] using a symmetric Kronecker product of a single $(n = 24, k = 6, d = 10)$ classical code.

In Table I, we enumerate all the smallest QTSs (with $a \leq 5, b \leq a$) present in the $[[900, 36, 10]]$ HP code having no CN with degree > 2 in their induced sub-graphs (computationally efficient task). These values for a and b are chosen as such classical-type TSs are typically the most harmful for iterative decoders. For symmetric HP codes, the QTS enumeration of H_{X} and H_{Z} is identical. The QTSs listed in Table I contribute to the poor iterative decoding performance [10] using BP decoding, where error patterns within the QTSs result in decoding failures. Using this QTS information, in [1], we proposed improved code design strategies by designing constituent classical codes devoid of these harmful QTSs.

TABLE I
QTS ENUMERATION IN $H_{\mathrm{X}}/H_{\mathrm{Z}}$ OF $[[900, 36, 10]]$ HP CODE [10]

Quantum TS	**Parameters** [1] (a, b) $(\mathrm{CYC}(x))$ Count	**Quantum TS**	**Parameters** (a, b) $(\mathrm{CYC}(x))$ Count
	(4,2) $(2x^6 + x^8)$ 720		(4,4) $(4x^6 + 3x^8)$ 72
	(5,1) $(2x^6 + 3x^8 + 2x^{10})$ 240		(5,4) $(4x^6 + 5x^8 + 4x^{10})$ 36
	(5,3) $(x^6 + x^8 + x^{10})$ 4080		(5,4) $(5x^6 + 5x^8 + 2x^{10})$ 90
	(5,3) $(3x^8)$ 360		(5,5) $(3x^8)$ 5184

V. Summary and Future Work

In this paper, we identified and classified quantum trapping sets using their definition adapted from the classical error correction to address the syndrome decoding scenario for QLDPC codes. The knowledge of QTSs is shown to significantly improve stabilizer code/decoder designs and also decoder performance in the error floor regimes of practical finite-length QLDPC codes. Analysis of failure configurations of the QLDPC codes, which are generalization of surface codes will have near-future implications in surface code designs and their decoders. In future work, we plan to modify the expansion-contraction method [14] to QLDPC codes to obtain the exact set of most harmful configurations that should be avoided in the Tanner graph of QLDPC codes. In addition, the extension of QTS definition to consider X and Z type errors together (correlated errors) and non-CSS stabilizer codes in general will set up the framework to study and explore non-binary quantum trapping sets.

[1] In Table I, the parameters-(a,b), $\mathrm{CYC}(x)$, and Count are listed row-wise under the column header-Parameters for each QTS.

REFERENCES

[1] N. Raveendran and B. Vasić, "Trapping sets of quantum LDPC codes," *arXiv:2012.15297*, 2020.

[2] D. MacKay, G. Mitchison, and P. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. on Inform. Theory*, vol. 50, no. 10, pp. 2315–2330, Oct. 2004.

[3] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct. 1995.

[4] D. Gottesman, "Fault-tolerant quantum computation with constant overhead," *Quantum Inform. and Computation*, vol. 14, no. 15–16, pp. 1338–1372, Nov. 2014.

[5] 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.

[6] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quantum Inform. and Computation*, vol. 8, no. 10, pp. 987–1000, Nov. 2008.

[7] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *IEEE Access*, vol. 3, pp. 2492–2519, 2015.

[8] B. Vasić, D. Nguyen, and S. K. Chilappagari, "Chapter 6 - failures and error floors of iterative decoders," in *Channel Coding: Theory, Algorithms, and Applications: Academic Press Library in Mobile and Wireless Commun.* Oxford: Academic Press, 2014, pp. 299–341.

[9] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," *arXiv:1904.02703*, 2019.

[10] J. Roffe, D. R. White, S. Burton, and E. T. Campbell, "Decoding across the quantum LDPC code landscape," *arXiv:2005.07016*, 2020.

[11] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasić, "Finite alphabet iterative decoders, Part I: Decoding beyond belief propagation on the binary symmetric channel," *IEEE Trans. on Commun.*, vol. 61, no. 10, pp. 4033–4045, Nov. 2013.

[12] J. X. Li and P. O. Vontobel, "Pseudocodeword-based decoding of quantum stabilizer codes," *arXiv:1903.01202*, 2019.

[13] A. Rigby, J. C. Olivier, and P. Jarvis, "Modified belief propagation decoders for quantum low-density parity-check codes," *Phys. Rev. A*, vol. 100, p. 012330, Jul. 2019. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.100.012330

[14] N. Raveendran, D. Declercq, and B. Vasić, "A sub-graph expansion-contraction method for error floor computation," *IEEE Trans. on Commun.*, vol. 68, no. 7, pp. 3984–3995, 2020.

[15] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug. 1996.

[16] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. New York, NY, USA: Cambridge University Press, 2011.

[17] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Institute of Technology, 1997.

[18] J.-P. Tillich and G. Zemor, "Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$," *Proc. IEEE Intl. Symp. on Inform. Theory*, pp. 799–803, Jul. 2009.

[19] N. Raveendran, P. J. Nadkarni, S. S. Garani, and B. Vasić, "Stochastic resonance decoding for quantum LDPC codes," in *Proc. IEEE Intl. Conf. on Commun.*, May 2017, pp. 1–6.

[20] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop on Signal Processing Systems*, 2004, pp. 107–112.

[21] M. Karimi and A. Banihashemi, "Efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inform. Theory*, vol. 58, no. 11, pp. 6942–6958, Nov. 2012.

[22] D. V. Nguyen, S. Chilappagari, M. Marcellin, and B. Vasić, "On the construction of structured LDPC codes free of small trapping sets," *IEEE Trans. Inform. Theory*, vol. 58, no. 4, pp. 2280–2302, Apr. 2012.

[23] S. M. Khatami, L. Danjean, D. V. Nguyen, and B. Vasić, "An efficient exhaustive low-weight codeword search for structured LDPC codes," in *Proc. Inform. Theory and Applications Workshop*, San Diego, CA, USA, Feb. 10–15 2013, pp. 1 – 10.

[24] A. A. Kovalev and L. P. Pryadko, "Quantum kronecker sum-product low-density parity-check codes with finite rate," *Phys. Rev. A*, vol. 88, p. 012311, Jul 2013. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.88.012311

[25] A. Kovalev and L. Pryadko, "Improved quantum hypergraph-product LDPC codes," in *Proc. IEEE Intl. Symp. on Inform. Theory*, Jul. 2012, pp. 348–352.