Distance-preserving stabilizer measurements in hypergraph product codes

Argyris Giannisis Manes and Jahan Claes

Departments of Physics and Applied Physics, Yale University, New Haven, CT 06520, USA
Yale Quantum Institute, Yale University, New Haven, Connecticut 06511, USA

Unlike the surface code, quantum low-density parity-check (QLDPC) codes can have a finite encoding rate, offering potentially lower overhead. However, finite-rate QLDPC codes have nonlocal stabilizers, making it difficult to design stabilizer measurement circuits that are low-depth and preserve the effective distance. Here, we demonstrate that a popular construction of finite-rate QLDPC codes, the hypergraph product, has the convenient property of distance-robustness: any stabilizer measurement circuit preserves the effective distance. In particular, we prove a previously proposed depth-optimal circuit is also optimal in terms of effective distance.

1 Introduction

Building a large-scale quantum computer will likely require encoding information in quantum error-correcting codes. Quantum low-density parity-check (QLDPC) codes are a broad family of quantum stabilizer codes in which errors are detected by measuring Pauli stabilizers of weight $O(1)$, and each qubit is included in $O(1)$ stabilizers [1]. A quantum code on n physical qubits encoding k logical qubits that can detect up to $(d-1)$ errors is denoted an $[[n,k,d]]$ code. The most widely studied QLDPC code is the surface code [2, 3], in which qubits are arranged in 2D Euclidean space so the stabilizers are geometrically local, and whose $[[n,k,d]]$ parameters satisfy $kd^2 = O(n)$. Indeed, it has been proven that any local QLDPC code in 2D Euclidean space necessarily has $kd^2 = O(n)$ [4, 5], and improving on these parameters requires some amount of non-local connectivity [6, 7] or non-Euclidean geometry [8]. Thus, to achieve a finite-rate code (in which $k = O(n)$ and $d > O(1)$), we must go beyond local 2D codes. [1]

There have been several recent constructions of non-local QLDPC codes with increasingly better parameters [8–17], culminating in the construction of "good" QLDPC codes with $k, d = \Theta(n)$ [15][see also 16, 17]. However, there is still considerable interest in hypergraph product (HGP) codes [12], an earlier construction achieving $k = \Theta(n)$, $d = \Theta(\sqrt{n})$. While HGP codes have worse d scaling than "good" QLDPC codes, the scaling for HGP codes is sufficient to prove a threshold exists [18, 19] and that large-scale quantum error correction requires only a constant overhead factor [19]. In addition, the HGP codes' relatively tractable construction has led to further development of decoders [20–26], logical gates [27–29], and low-depth stabilizer measurement circuits [30, 31], and it has been recently demonstrated that HGP codes can be designed to be compatible with modular architectures [32] and reconfigurable atom arrrays [33].

For any QLDPC code, an immediate question is: how do we measure the stabilizers? One typically wants to measure stabilizers using ancilla qubits and low-depth quantum circuits, but these circuits must be optimized to simultaneously measure overlapping stabilizers [34–37]. Ref. [30] introduced low-depth circuits for general CSS-type QLDPC codes, as well as even lower-depth circuits that measured all stabilizers in parallel for particular HGP codes. Besides the depth of measurement circuits, one should also optimize how measurement circuits propagate errors. A single error on an ancilla qubit during a circuit measuring a weight-ω stabilizer can propagate to a weight $\lfloor \omega/2 \rfloor$ error on the data qubits, which may reduce the effective distance of the code from d to $\lceil d/\lfloor \omega/2 \rfloor \rceil$ in the worst case [38]. The original surface code has the convenient property that no measurement circuit reduces the effective distance [34], but the rotated surface code requires a carefully chosen measurement schedule to avoid reducing the distance to $\lceil d/2 \rceil$ [38]. For a general QLDPC code with many logical operators and no local structure, there may not exist a circuit that preserves the distance. Indeed, a recent work [39] has proposed stabilizer measurement circuits for a finite-rate QLDPC code family closely related to HGP codes known as quasicyclic codes [40]. In that work, it was found that randomly chosen stabilizer measurement circuits sharply reduced the code distance, and even optimized circuits could not fully preserve the distance [39].

In this paper, we prove that HGP codes have the same distance robustness as the original surface code:

Argyris Giannisis Manes: agiannisismanes@uchicago.edu
Jahan Claes: jahan.claes@yale.edu

[1] Note if we do not have $d > O(1)$, the fraction of errors that are correctable will vanish as $n \to \infty$.

any valid stabilizer-measurement circuit does not reduce the effective distance of the code. As HGP codes are a generalization of the surface code, our result is a generalization of the robustness of the surface code. Our result shows that the "product coloration circuit" that is most directly compatible with reconfigurable atom arrays [33] can be used without reducing the distance. In addition, our result establishes that the "cardinal" measurement circuit for HGP codes introduced in [30] is in some sense optimal: the circuit has the lowest possible depth and does not reduce the effective code distance, satisfying the two typical desiderata for measurement circuits [36, 38].

A natural question raised by our proof is: Can it be applied to the lifted product [15, 41], a generalization of the HGP with linear distance scaling? Unfortunately, for general lifted product codes, no such proof is possible. Lifted product codes contain all generalized bicycle codes [41], and generalized bicycle codes contain both a subset of rotated toric codes which are not distance robust [40, example 2] as well as the above-mentioned quasicyclic codes [39]. We note that it is still an interesting question whether some subset of lifted products can be found with both linear distance scaling and distance-robustness.

2 Preliminaries: HGP Construction

The HGP construction [12] takes two classical linear codes, described by parity check matrices H_1 and H_2 with n_i bits and r_i parity checks, and defines a quantum CSS code [42, 43]. Note that the rows of H_i may be linearly dependent, i.e. we allow redundant checks. We illustrate the construction in Fig. 1. Each classical code is described by a Tanner graph, where bits are represented by circles and parity checks are represented by squares connected to the circles they act on. To form the HGP code, we take the Cartesian product of the two classical Tanner graphs. The qubits of the quantum code come in two types, given by the product of the bits and bits or checks and checks (hereafter, bit- and check-type qubits). The $X(Z)$-stabilizers of the code are given by the product of checks and bits (bits and checks), and act on the qubits they are connected to in the Cartesian product graph. The X- and Z-stabilizers are then guaranteed to commute. Note that every stabilizer in Fig. 1 acts on the qubits in the same row or column as that stabilizer. $X(Z)$ stabilizers act on bit(check)-type qubits above and below them, and check(bit)-type qubits to their left and right.

Algebraically, this construction corresponds to check matrices $H_X(H_Z)$ whose rows describe the $X(Z)$-stabilizers

$$\begin{aligned} H_X &= \left(H_1 \otimes I_{n_2} \quad , \quad I_{r_1} \otimes H_2^T\right) \\ H_Z &= \left(I_{n_1} \otimes H_2 \quad , \quad H_1^T \otimes I_{r_2}\right) \end{aligned} \tag{1}$$

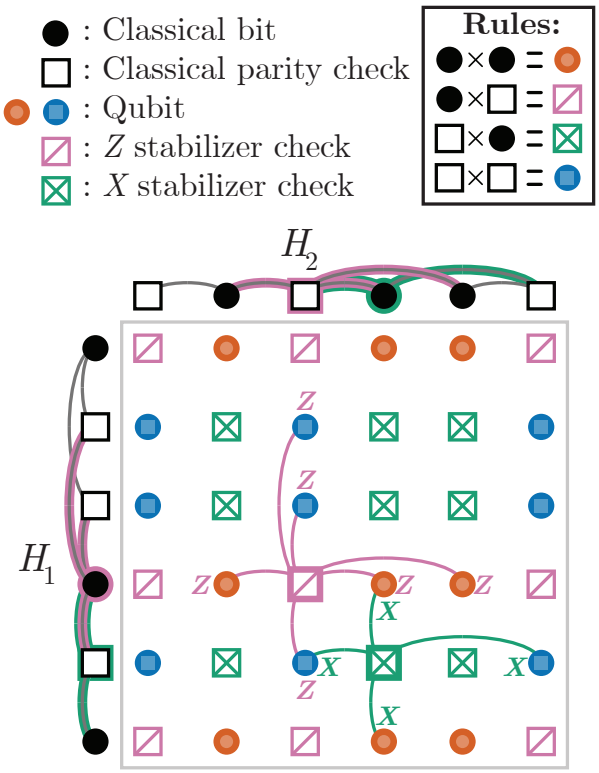

Figure 1: Lattice illustration of an HGP code as the Cartesian product of two Tanner graphs of classical linear codes H_1, H_2. The black circles and squares correspond to the bits and checks of the classical codes. The product of a bit with a bit (check with a check) is a bit-type (check-type) qubit, illustrated in orange (blue). The product of a bit with a check (check with a bit) is a $Z(X)$-stabilizer. The stabilizers are supported on connected qubits in both directions; the connections are determined from the classical parity check matrices. Two example stabilizers are illustrated.

where the comma separates concatenated matrices. The $n_1 n_2$ columns before the comma correspond to the bit-type qubits, and the remaining $r_1 r_2$ columns after the comma correspond to the check-type qubits.

The parameters $[[n, k, d]]$ of the HGP code can be expressed in terms of the classical code parameters. Let the classical code H_i have parameters $[n_i, k_i, d_i]$; let the classical transposed code H_i^T, formed by swapping the role of checks and bits in the Tanner graph, have parameters $[r_i, k_i^T, d_i^T]$. Then the HGP code has

$$n = n_1 n_2 + r_1 r_2 \tag{2}$$

$$k = k_1 k_2 + k_1^T k_2^T \tag{3}$$

$$d = \begin{cases} \min(d_1, d_2) & , \ d_1^T = \infty \text{ or } d_2^T = \infty \\ \min(d_1^T, d_2^T) & , \ d_1 = \infty \text{ or } d_2 = \infty \\ \min(d_1, d_2, d_1^T, d_2^T) & , \ \text{else.} \end{cases} \tag{4}$$

The formula for n is obvious, while k can be straightforwardly derived by computing the ranks of H_X and H_Z. We prove the formula for d while proving Thm. 2 below. Choosing classical LDPC codes with

$k_i, d_i \propto n_i$ gives sparse HGP codes with $k \propto n$ and $d \propto \sqrt{n}$ [12].

3 HGP Logical Operators

To understand the distance of the HGP code, we explicitly construct the k logical operators. These logical operators are formed by taking the product of classical codewords of H_i or H_i^T with certain weight-one vectors.

Following [12, 25], we define two classes of logical Z operators, illustrated in Fig. 2. The first is generated by a basis $\{x_i\}$ of classical codewords of H_1 and a set $\{y_j\}$ of unit vectors whose linear combinations are outside the image of H_2^T. Clearly, $|\{x_i\}| = k_1$. Since $\dim\left[\mathrm{Im}(H_2^T)\right] = (n_2 - k_2)$, we can always find k_2 unit vectors whose span never falls inside $\mathrm{Im}(H_2^T)$, so that $|\{y_j\}| = k_2$. The first $k_1 k_2$ logical Z operators are then given by

$$\begin{pmatrix} x_i \otimes y_j \\ 0 \end{pmatrix}. \tag{5}$$

These operators are oriented vertically in Fig. 2 and act only on bit-type qubits. They have weight $\geq d_1$, with at least k_1 having weight d_1 (provided $k_2 \neq 0$). Pictorially, it's clear they commute with X-stabilizers: each X-stabilizer intersects the Z operator either zero times, or the same number of times its corresponding classical check in H_1 intersects the classical codeword x_i. Because x_i is a codeword of H_1, this intersection must be even.

The second class is generated by a basis $\{b_m\}$ of classical codewords of H_2^T and a set $\{a_\ell\}$ of unit vectors whose linear combinations are outside $\mathrm{Im}(H_1)$. We similarly have $|\{b_m\}| = k_2^T$, $|\{a_\ell\}| = k_1^T$. Then the second $k_1^T k_2^T$ logical Z operators are given by

$$\begin{pmatrix} 0 \\ a_\ell \otimes b_m \end{pmatrix}. \tag{6}$$

which are oriented horizontally and act only on check-type qubits. They have weight $\geq d_2^T$, with at least k_1^T having weight d_2^T (provided $k_2^T \neq 0$). These operators commute with X-stabilizers for similar reasons. We have thus enumerated $k = (k_1 k_2 + k_1^T k_2^T)$ logical Z-operators.

To ensure that these Z-operators are independent, we need to establish that no combination of Z-operators can be written as a combination of Z stabilizers. WLOG, we consider a combination of Z operators which acts on at least one row of bit-type qubits (e.g., highlighted row of Fig. 2). We can restrict to this row and view it as a copy of the classical code H_2. The combination of Z-operators restricted to this row is made up of a combination of $\{y_j\}$. Any Z stabilizer restricted to this row is an element of $\mathrm{Im}(H_2^T)$. Thus, since no combination of $\{y_j\}$ is in $\mathrm{Im}(H_2^T)$, no combination of Z stabilizers can equal the logical operator.

We summarize this discussion in the following theorem and formal proof:

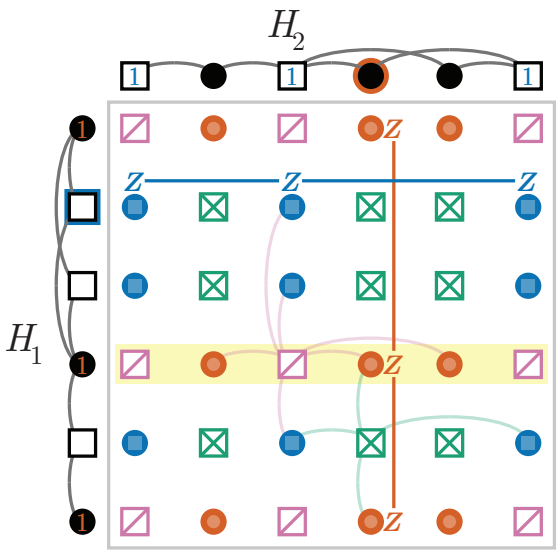

Figure 2: Examples of elementary logical Z operators for an HGP code, as given in Eqs 5 and 6. The logical operators are obtained by combining a codeword of one of the classical codes (in this case, 111) with a unit vector that lies outside the image of the other code's transpose check matrix.

Theorem 1 ([12, 25]). *The set of operators given by Eqs. 5 and 6 are a complete set of independent logical Z operators.*

Proof. We need to show that the set of operators given by Eqs. 5 and 6 are a basis for $\ker(H_X)/\mathrm{Im}(H_Z^T)$. We note that this space is k-dimensional, so if the k vectors enumerated by Eqs. 5 and 6 are in $\ker(H_X)$ and are linearly independent in $\ker(H_X)/\mathrm{Im}(H_Z^T)$, they are a basis. We therefore consider an arbitrary non-trivial linear combination of such operators:

$$z = \begin{pmatrix} \sum_{ij} \lambda_{ij} x_i \otimes y_j \\ \sum_{\ell m} \kappa_{\ell m} a_\ell \otimes b_m \end{pmatrix}, \tag{7}$$

where λ_{ij} and $\kappa_{\ell m}$ are arbitrary binary values.

First, we show $z \in \ker(H_X)$. We have (see Eq. 1 for H_X)

$$\begin{aligned} H_X z &= \sum_{ij} \lambda_{ij} \left(H_1 x_i\right) \otimes y_j + \sum_{\ell m} \kappa_{\ell m} a_\ell \otimes \left(H_2^T b_m\right) \\ &= 0 \end{aligned} \tag{8}$$

where we have used $H_1 x_i = 0$ ($H_2^T b_m = 0$), since x_i (b_m) are codewords of H_1 (H_2^T).

Second, we show that $z \notin \mathrm{Im}(H_Z^T)$, so that z is nonzero in $\ker(H_X)/\mathrm{Im}(H_Z^T)$. We establish this by contradiction. If $z \in \mathrm{Im}(H_Z^T)$, there would exist some vector w such that $z = H_Z^T w$, or

$$z = \begin{pmatrix} \sum_{ij} \lambda_{ij} x_i \otimes y_j \\ \sum_{\ell m} \kappa_{\ell m} a_\ell \otimes b_m \end{pmatrix} = \begin{pmatrix} (I_{n_1} \otimes H_2^T) w \\ (H_1 \otimes I_{r_2}) w \end{pmatrix} \tag{9}$$

We can define new codewords $x'_j := \sum_i \lambda_{ij} x_i$ and $b'_\ell := \sum_m \kappa_{\ell m} b_m$. We note that if some $\lambda_{ij} \neq 0$ then

the corresponding $x'_j \neq 0$, since the x_i are linearly independent. Similarly, if some $\kappa_{\ell m} \neq 0$ then the corresponding $b'_\ell \neq 0$. Without loss of generality, let's assume $\lambda_{ij} \neq 0$ and $x'_j \neq 0$; the case where all $\lambda_{ij} = 0$ and some $\kappa_{\ell m} \neq 0$ can be treated similarly. Expand w as $w = \sum_k w_k \otimes \tilde{w}_k$ (such a decomposition is always possible by choosing $\{w_k\}$ to be a basis of the first tensor product factor). Then the top row of Eq. 9 becomes

$$\sum_j x'_j \otimes y_j = \sum_k w_k \otimes \left(H_2^T \tilde{w}_k\right). \tag{10}$$

As there exists an $x'_j \neq 0$, it is possible to pick an n_1-dimensional unit vector e_t such that $e_t \cdot x'_j \neq 0$. If we apply the linear operator $e_t^T \otimes I_{n_2}$ (an $n_2 \times n_1 n_2$ dimensional matrix) to both sides, we have

$$\sum_j (e_t \cdot x'_j) y_j = \sum_k (e_t \cdot w_k) H_2^T \tilde{w}_k \tag{11}$$

which says a nonzero combination of $\{y_j\}$ is in $\mathrm{Im}(H_2^T)$, contradicting our choice for the $\{y_j\}$. Thus $z \notin \mathrm{Im}(H_Z^T)$. □

The logical X operators can be found similarly, by swapping the roles of H_1 and H_2.

4 Effective Code Distance

When measuring a stabilizer using a single ancilla qubit and a set of two-qubit entangling gates, errors can propagate from the ancilla qubit to multiple code qubits. For instance, consider the circuit measuring a Z stabilizer of weight 4 in Fig. 3. In this case, if a Z error occurs on the ancilla qubit during the measurement, it can propagate to two physical qubits. Such an error is often called a "hook error" in the literature [34]

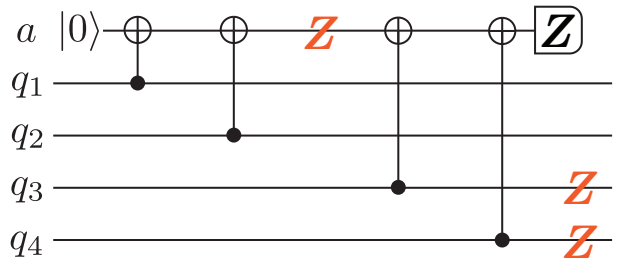

Figure 3: An example circuit for measuring a Z stabilizer supported on code qubits $q_1, \ldots, q_4$. A single Z error on the ancilla qubit halfway through the circuit propagates to Z errors on code qubits q_3 and q_4, potentially reducing the number of physical errors that can be corrected.

In general, when measuring a stabilizer of weight ω, a single ancilla error can propagate to any subset of code qubits in the support of the stabilizer (the specific subsets will depend on the order we apply the two-qubit gates). Because we can always multiply an error by a stabilizer to reduce its weight, a single error will propagate to up to $\lfloor \frac{\omega}{2} \rfloor$ code qubits. This error propagation may reduce the number of physical errors required for an undetectable logical error by a factor of up to $\lfloor \frac{\omega}{2} \rfloor$ [36–38].

For a given code and measurement circuit, we define the **effective distance** to be the minimum number of physical errors required for an undetectable logical error. We note that only ancilla errors propagating to data qubits can reduce the effective distance, as data qubit errors never spread to neighboring data qubits..

In the particular case of HGP codes, error propagation never reduces the distance of the code. Specifically, if we assume we measure each row of $H_Z(H_X)$ using a circuit as in Fig. 3 and allow a $Z(X)$ error on the ancilla qubit to introduce a $Z(X)$ error on any subset of the qubits in the support of that stabilizer, we will show that at least d physical errors are still needed for an undetectable logical error[2]. This is summarized in the following theorem.

Theorem 2. *Let d be the distance of an HGP code. The effective distance of the code using any stabilizer measurement order for the rows of H_Z and H_X is also d.*

Before formally proving this statement, we provide an intuitive explanation. Any combination z of logical operators from Eqs. 5 and 6 will have at least one column (row) with at least d_1 (d_2^T) Z operators. WLOG, assume it is a column, as in the vertical logical operator in Fig. 2. z then has support on at least d_1 rows. Restricted to any one of these rows, z is a combination of vectors $\{y_i\}$. Therefore, any combination of z with stabilizers still has support on this row, since any stabilizer restricted to this row is in $\mathrm{Im}(H_2^T)$ and a combination of $\{y_i\}$ can never be in $\mathrm{Im}(H_2^T)$. Thus, even if we multiply z by Z stabilizers, we cannot have a logical operator with support in fewer than d_1 rows. If we multiply z by a partial Z stabilizer induced by an ancilla error, at worst this stabilizer can cancel the support of z on one row, since each stabilizer only acts on a single row of bit-type qubits. Thus, an ancilla error can only reduce the weight of the logical by 1, just like a direct data qubit error.

We now turn to our formal proof.

Proof (See also [25, Prop. 2]). An elementary Z stabilizer (row of H_Z) is given by $s_{t\tilde{t}} := H_Z^T(e_t \otimes \tilde{e}_{\tilde{t}})$, where $e_t \otimes \tilde{e}_{\tilde{t}}$ is a unit vector, so

$$s_{t\tilde{t}} = \begin{pmatrix} e_t \otimes H_2^T \tilde{e}_{\tilde{t}} \\ H_1 e_t \otimes \tilde{e}_{\tilde{t}} \end{pmatrix}. \tag{12}$$

An ancilla error may propagate Z errors to a subset of the qubits in the support of the stabilizer. However, we see that a stabilizer $s_{t\tilde{t}}$ is supported on qubits $e_t \otimes$

[2]Note that a Y error on an ancilla qubit propagates as either an X or Z error during measurement, so we do not need to separately consider Y errors.

$(\cdot)$ and $(\cdot)\otimes\tilde{e}_{\tilde{t}}$. Pictorally, this says that a Z stabilizer $s_{t\tilde{t}}$ only act on bit-type qubits in the row e_t and check-type qubits in the column $\tilde{e}_{\tilde{t}}$, as illustrated in Fig. 1. If every logical Z operator has support in at least d_1 distinct rows e_t or d_2^T distinct columns $\tilde{e}_{\tilde{t}}$, we still require d ancilla errors to cover a logical operator.

Consider an arbitrary logical Z operator consisting of a product of logical Z operators and Z stabilizers

$$z = \begin{pmatrix} \sum_j x'_j \otimes y_j + \sum_k w_k \otimes H_2^T \tilde{w}_k \\ \sum_\ell a_\ell \otimes b'_\ell + \sum_k H_1 w_k \otimes \tilde{w}_k \end{pmatrix} := \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{13}$$

where we have again made use of the definitions of x'_j and b'_ℓ introduced above, and again expanded w as $\sum_k w_k \otimes \tilde{w}_k$. Without loss of generality, assume some $x'_j \neq 0$ (the case where all $x'_j = 0$ and some $b'_\ell \neq 0$ proceeds similarly).

Since $x'_j \neq 0$, it must have hamming weight $\geq d_1$ and there are therefore at least d_1 distinct unit vectors e_t such that $e_t \cdot x'_j = 1$. We again apply $e_t^T \otimes I_{n_2}$ to the top row of Eq. 13:

$$(e_t^T \otimes I_{n_2}) z_1 = \sum_j (e_t \cdot x'_j) y_j + \sum_k (e_t \cdot w_k) H_2^T \tilde{w}_k. \tag{14}$$

This equation cannot be zero, because at least one of the $(e_t \cdot x'_j) \neq 0$, and no nonzero linear combination of y_j is in the image of H_2^T. We then see even when allowing for multiplication by stabilizers, z_1 has nonzero support in each row corresponding to e_t. Thus, we still require d_1 physical errors to cover the logical operator z.

The proof for X logical operators is identical up to exchanging the roles of H_1 and H_2. □

Acknowledgments

Note added— After preparing the initial version of our manuscript, we became aware that the fact that logical operators have support in at least d rows/columns has also been independently shown in [25, Prop 2] in a different context. However, our overall proof is distinct from theirs, and our application to distance robustness is novel.

Acknowledgments— We are grateful to Shruti Puri and Shilin Huang for helpful discussions. This material is based on work supported by the National Science Foundation (NSF) under Award No. 2137740. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF.

References

[1] Nikolas P Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2(4):040101, 2021. DOI: https://doi.org/10.1103/PRXQuantum.2.040101.

[2] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997. DOI: https://doi.org/10.1070/RM1997v052n06ABEH002155.

[3] Sergey B Bravyi and A Yu Kitaev. Quantum codes on a lattice with boundary. *arXiv preprint quant-ph/9811052*, 1998. DOI: https://doi.org/10.48550/arXiv.quant-ph/9811052.

[4] Sergey Bravyi and Barbara Terhal. A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes. *New Journal of Physics*, 11(4):043029, 2009. DOI: https://doi.org/10.1088/1367-2630/11/4/043029.

[5] Sergey Bravyi, David Poulin, and Barbara Terhal. Tradeoffs for reliable quantum information storage in 2d systems. *Physical Review Letters*, 104(5):050503, 2010. DOI: https://doi.org/10.1103/PhysRevLett.104.050503.

[6] Nouédyn Baspin and Anirudh Krishna. Quantifying nonlocality: how outperforming local quantum codes is expensive. *Physical Review Letters*, 129(5):050505, 2022. DOI: https://doi.org/10.1103/PhysRevLett.129.050505.

[7] Nouédyn Baspin, Venkatesan Guruswami, Anirudh Krishna, and Ray Li. Improved rate-distance trade-offs for quantum codes with restricted connectivity. *arXiv preprint arXiv:2307.03283*, 2023. DOI: https://doi.org/10.48550/arXiv.2307.03283.

[8] Nikolas P Breuckmann and Barbara M Terhal. Constructions and noise threshold of hyperbolic surface codes. *IEEE transactions on Information Theory*, 62(6):3731–3744, 2016. DOI: https://doi.org/10.1109/TIT.2016.2555700.

[9] Michael H Freedman, David A Meyer, and Feng Luo. Z2-systolic freedom and quantum codes. *Mathematics of quantum computation, Chapman & Hall/CRC*, pages 287–320, 2002.

[10] Larry Guth and Alexander Lubotzky. Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds. *Journal of Mathematical Physics*, 55(8):082202, 2014. DOI: https://doi.org/10.1063/1.4891487.

[11] Vivien Londe and Anthony Leverrier. Golden codes: quantum LDPC codes built from regular tessellations of hyperbolic 4-manifolds. *Quantum Information & Computation*, 19(5&6), 2019. DOI: https://doi.org/10.26421/QIC19.5-6-1.

[12] Jean-Pierre Tillich and Gilles Zémor. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2013. DOI: https://doi.org/10.1109/TIT.2013.2292061.

[13] Matthew B Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: Break-

ing the $n^{1/2}$/polylog(n) barrier for quantum LDPC codes. pages 1276–1288, 2021. DOI: https://doi.org/10.1145/3406325.3451005.

[14] Nikolas P Breuckmann and Jens N Eberhardt. Balanced product quantum codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021. DOI: https://doi.org/10.1109/TIT.2021.3097347.

[15] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 375–388, 2022. DOI: https://doi.org/10.48550/arXiv.2111.03654.

[16] Anthony Leverrier and Gilles Zémor. Quantum tanner codes. *arXiv preprint arXiv:2202.13641*, 2022. DOI: https://doi.org/10.48550/arXiv.2202.13641.

[17] Ting-Chun Lin and Min-Hsiu Hsieh. Good quantum ldpc codes with linear time decoder from lossless expanders. *arXiv preprint arXiv:2203.03581*, 2022. DOI: https://doi.org/10.48550/arXiv.2203.03581.

[18] Alexey A Kovalev and Leonid P Pryadko. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Physical Review A*, 87(2):020304, 2013. DOI: https://doi.org/10.1103/PhysRevA.87.020304.

[19] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *arXiv preprint arXiv:1310.2984*, 2013. DOI: https://doi.org/10.48550/arXiv.1310.2984.

[20] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 810–824. IEEE, 2015. DOI: https://doi.org/10.1109/FOCS.2015.55.

[21] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Efficient decoding of random errors for quantum expander codes. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 521–534, 2018. DOI: https://doi.org/10.1145/3188745.3188886.

[22] Antoine Grospellier and Anirudh Krishna. Numerical study of hypergraph product codes. *arXiv preprint arXiv:1810.03681*, 2018. DOI: https://doi.org/10.48550/arXiv.1810.03681.

[23] Joschka Roffe, David R White, Simon Burton, and Earl Campbell. Decoding across the quantum low-density parity-check code landscape. *Physical Review Research*, 2(4):043423, 2020. DOI: https://doi.org/10.1103/PhysRevResearch.2.043423.

[24] Antoine Grospellier, Lucien Grouès, Anirudh Krishna, and Anthony Leverrier. Combining hard and soft decoders for hypergraph product codes. *Quantum*, 5:432, 2021. DOI: https://doi.org/10.22331/q-2021-04-15-432.

[25] Armanda O Quintavalle and Earl T Campbell. Reshape: A decoder for hypergraph product codes. *IEEE Transactions on Information Theory*, 68(10):6569–6584, 2022. DOI: https://doi.org/10.1109/TIT.2022.3184108.

[26] Julien Du Crest, Francisco Garcia-Herrero, Mehdi Mhalla, Valentin Savin, and Javier Valls. Layered decoding of quantum ldpc codes. *arXiv preprint arXiv:2308.13377*, 2023. DOI: https://doi.org/10.48550/arXiv.2308.13377.

[27] Anirudh Krishna and David Poulin. Fault-tolerant gates on hypergraph product codes. *Physical Review X*, 11(1):011023, 2021. DOI: https://doi.org/10.1103/PhysRevX.11.011023.

[28] Armanda O Quintavalle, Paul Webster, and Michael Vasmer. Partitioning qubits in hypergraph product codes to implement logical gates. *Quantum*, 7:1153, 2023. DOI: https://doi.org/10.22331/q-2023-10-24-1153.

[29] Lawrence Z Cohen, Isaac H Kim, Stephen D Bartlett, and Benjamin J Brown. Low-overhead fault-tolerant quantum computing using long-range connectivity. *Science Advances*, 8(20):eabn1717, 2022. DOI: https://doi.org/10.1126/sciadv.abn1717.

[30] Maxime A Tremblay, Nicolas Delfosse, and Michael E Beverland. Constant-overhead quantum error correction with thin planar connectivity. *Physical Review Letters*, 129(5):050504, 2022. DOI: https://doi.org/10.1103/PhysRevLett.129.050504.

[31] Nicolas Delfosse, Michael E Beverland, and Maxime A Tremblay. Bounds on stabilizer measurement circuits and obstructions to local implementations of quantum ldpc codes. *arXiv preprint arXiv:2109.14599*, 2021. DOI: https://doi.org/10.48550/arXiv.2109.14599.

[32] Armands Strikis and Lucas Berent. Quantum low-density parity-check codes for modular architectures. *PRX Quantum*, 4(2):020321, 2023. DOI: https://doi.org/10.1103/PRXQuantum.4.020321.

[33] Qian Xu, J. Pablo Bonilla Ataides, Christopher A. Pattison, Nithin Raveendran, Dolev Bluvstein, Jonathan Wurtz, Bane Vasic, Mikhail D. Lukin, Liang Jiang, and Hengyun Zhou. Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *arXiv preprint arXiv:2308.08648*, 2023. DOI: https://doi.org/10.48550/arXiv.2308.08648.

[34] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002. DOI: https://doi.org/10.1063/1.1499754.

[35] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical

large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012. DOI: https://doi.org/10.1103/PhysRevA.86.032324.

[36] Andrew J Landahl, Jonas T Anderson, and Patrick R Rice. Fault-tolerant quantum computing with color codes. *arXiv preprint arXiv:1108.5738*, 2011. DOI: https://doi.org/10.48550/arXiv.1108.5738.

[37] Michael E Beverland, Aleksander Kubica, and Krysta M Svore. Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes. *PRX Quantum*, 2(2):020341, 2021. DOI: https://doi.org/10.1103/PRXQuantum.2.020341.

[38] Yu Tomita and Krysta M Svore. Low-distance surface codes under realistic quantum noise. *Physical Review A*, 90(6):062320, 2014. DOI: https://doi.org/10.1103/PhysRevA.90.062320.

[39] Sergey Bravyi, Andrew W Cross, Jay M Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, 2024. DOI: https://doi.org/10.1038/s41586-024-07107-7.

[40] Alexey A Kovalev and Leonid P Pryadko. Quantum kronecker sum-product low-density parity-check codes with finite rate. *Physical Review A*, 88(1):012311, 2013. DOI: https://doi.org/10.1103/PhysRevA.88.012311.

[41] Pavel Panteleev and Gleb Kalachev. Quantum ldpc codes with almost linear minimum distance. *IEEE Transactions on Information Theory*, 68(1):213–229, 2021. DOI: https://doi.org/10.1109/TIT.2021.3119384.

[42] A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996. DOI: https://doi.org/10.1103/PhysRevA.54.1098.

[43] Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, 1996. DOI: https://doi.org/10.1098/rspa.1996.0136.