Power and Limitations of Linear Programming Decoder for Quantum LDPC Codes

Shouzhen Gu
Applied physics, Yale University, New Haven, Connecticut 06520, USA and
Yale Quantum Institute, Yale University, New Haven, Connecticut 06520, USA

Mehdi Soleimanifar
California Institute of Technology, Pasadena, California 91125, USA

Decoding quantum error-correcting codes is a key challenge in enabling fault-tolerant quantum computation. In the classical setting, linear programming (LP) decoders offer provable performance guarantees and can leverage fast practical optimization algorithms. Although LP decoders have been proposed for quantum codes, their performance and limitations remain relatively underexplored. In this work, we uncover a key limitation of LP decoding for quantum low-density parity-check (LDPC) codes: certain constant-weight error patterns lead to ambiguous fractional solutions that cannot be resolved through independent rounding. To address this issue, we incorporate a post-processing technique known as ordered statistics decoding (OSD), which significantly enhances LP decoding performance in practice. Our results show that LP decoding, when augmented with OSD, can outperform belief propagation with the same post-processing for intermediate code sizes of up to hundreds of qubits. These findings suggest that LP-based decoders, equipped with effective post-processing, offer a promising approach for decoding near-term quantum LDPC codes.

I. INTRODUCTION

Quantum error correction (QEC) is widely believed to be necessary to protect quantum information from the inevitable noise and imperfections that arise in quantum systems [1–3]. Effective QEC codes should have a large distance and high rate so that they can tolerate a significant number of errors while maintaining low overhead. Another desirable feature is that the stabilizer measurements used to obtain information about errors have low weight, allowing for efficient and fault-tolerant syndrome extraction. Quantum low-density parity-check (LDPC) codes [4] are families of codes that satisfy this sparsity property, and they hold the potential for enabling large-scale quantum computation with low overhead [5–9]. Recent constructions of quantum LDPC codes have led to improved parameters both in the asymptotic regime [10–16] and for finite code sizes [6, 7].

To make LDPC codes practically useful, they must be paired with efficient *decoding algorithms* capable of determining errors that have affected the quantum state using information from the stabilizer measurements. Computationally efficient and provably correct decoders have been introduced for certain families of LDPC codes, including quantum expander codes [17, 18] and asymptotically good codes [16, 19–21]. However, these theoretical algorithms are not always practical to implement. While they achieve linear runtime in the blocklength, the constant prefactors, which may be exponential in the code's check weight, could be significant. Alternatively, heuristic algorithms have been developed that are efficient and accurate in practice [22–24].

One promising direction for developing decoding algorithms for quantum codes is to adapt successful decoders for classical codes to the quantum setting. For example, the flip algorithm for expander codes [25] was modified to small-set-flip [17] and its variants [16, 19, 20], and belief propagation (BP) for classical LDPC codes [26, 27] was enhanced with ordered statistics decoding (OSD) post-processing (BP+OSD) [22, 23].

Linear programming (LP) is another widely used decoder for classical LDPC codes [28, 29]. LP decoding is based on formulating the problem of finding the minimum-weight error consistent with a given syndrome as an integer program (IP) and relaxing it to a continuous optimization problem. Thus, highly optimized LP solvers [30] can be used to enable fast LP decoding. LP decoders have provable performance guarantees for some classical LDPC codes [31, 32], such as expander codes, and provide certificates confirming minimum-weight solutions. Although LP decoders remain relatively unexplored in the quantum setting, prior works have considered their direct extension, demonstrating promising performance for certain codes [33–35].

However, a significant challenge in applying LP decoders to quantum codes is the occurrence of *fractional solutions* that do not correspond to valid physical corrections. Such behavior was shown to be problematic for the surface code, codes with poor soundness [33], and hypergraph product (HGP) codes with carefully tuned parameters [35]. In this work, we ask:

Do fractional solutions arise
generically in quantum LDPC codes?

We answer this in the affirmative. In Sec. III, we show that quantum codes generally allow for small, often constant-sized, error patterns which are uncorrectable by LP. Such patterns, related to cycles in the code's Tanner graph, can prevent LP from attaining a threshold.

One approach for handling a fractional solution produced by the LP decoder is to round each entry of the solution independently [33, 35]. While this method guarantees an integral solution, it is still not able to correct the problematic errors we identify. Furthermore, the correction after independent rounding may not even have the same syndrome as the error, which our simulations show is the case in the vast majority of LP failures (Fig. 6). Motivated by this observation, we ask:

Are there more sophisticated rounding techniques that can improve the performance of LP decoders?

To explore this question, we draw inspiration from BP. BP is a fast message-passing decoder [27] that is also known to perform well for classical codes [36–38] but has convergence issues for quantum codes due to short cycles in the Tanner graph [39–41]. A common approach to address the non-convergence of BP is to apply OSD as a post-processing step. In BP+OSD, the less reliable bits identified by BP are erased and Gaussian elimination is then used to recover a valid correction consistent with the given syndrome [22, 23]. By interpreting the fractional outputs of LP as the reliability of the solution, we propose using OSD as a post-processing step in LP decoding (LP+OSD) to round the output to an integral solution satisfying the syndrome. We show that the problematic error patterns identified in our analysis in Sec. III can be corrected by OSD post-processing.

Another motivation for combining LP with OSD is that BP+OSD is already bottlenecked by the $O(n^3)$ runtime of the matrix inversion step in OSD. Therefore, replacing BP with LP, which also has complexity $O(n^3)$, does not increase the overall asymptotic complexity. Since standalone LP decoding performs as well as—or better than—BP for many classical [28] and quantum codes [33, 35], one might expect that LP+OSD achieves lower logical error rates than BP+OSD. In this work, we find that this is the case for some mid-sized LDPC codes of up to 400 physical qubits (Figs. 3 and 4).

The remainder of the paper is structured as follows. After introducing some background on quantum LDPC codes and the LP and OSD decoders in Sec. II, we describe the uncorrectable error patterns for LP in Sec. III. In Sec. IV, we introduce our proposed LP+OSD decoder and explain why OSD post-pocessing helps in handling these problematic error patterns. We present the results of our numerical simulations in Sec. V. Finally, we conclude in Sec. VI and discuss possible future directions.

II. PRELIMINARIES

We first review some facts about quantum LDPC codes and give an overview of the LP and OSD decoders.

II.1. Classical and quantum codes

A classical linear code is a subspace $C \subseteq \mathbb{F}_2^A$, where A is a set indexing the bits. The classical code is often defined using a parity check matrix $H \in \mathbb{F}_2^{B\times A}$ with the code defined as the kernel: $C = \ker H$. Here, the rows B index the set of parity checks that must be satisfied for a vector $x \in \mathbb{F}_2^A$ to be in the codespace. To visualize the code, we can represent it using a Tanner graph $T(C)$, which is a bipartite graph with vertices $A \sqcup B$. The edges are defined to connect $a \in A$ and $b \in B$ if the matrix element $H_{ba} = 1$.

A quantum stabilizer code $\mathcal{C}$ is a subspace of the n-qubit Hilbert space $\left(\mathbb{C}^2\right)^{\otimes n}$, defined as the simultaneous $+1$-eigenspace of an abelian subgroup S of the n-qubit Pauli group not containing $-I$, called the stabilizer group. When S can be generated by operators from two sets S_X and S_Z consisting of Pauli operators of X and Z type, respectively, we call $\mathcal{C}$ a Calderbank-Shor-Steane (CSS) code [42, 43]. We often identify X- or Z-type Pauli operators, their supports as subsets, and the corresponding vectors in $\mathbb{F}_2^n$ with each other. Associated with any CSS code are two parity check matrices $H_X \in \mathbb{F}_2^{m_X\times n}$ and $H_Z \in \mathbb{F}_2^{m_Z\times n}$, where the rows represent (supports of) the stabilizer generators of each type. When H_X and H_Z have at most a constant number of ones in each row and column, we say that the code is LDPC.

The number of logical qubits encoded in a CSS code is the dimension $k = n - \operatorname{rank} H_X - \operatorname{rank} H_Z$, and the rate is k/n. The distance is the weight of the smallest nontrivial logical operator, which is given by $d = \min\{d_X, d_Z\}$ with $d_X = \min\{|x| : x \in \ker H_Z \setminus \operatorname{row} H_X\}$ and $d_Z = \min\{|x| : x \in \ker H_X \setminus \operatorname{row} H_Z\}$. Here, row A is the rowspace of a matrix A and $|x|$ is the Hamming weight of a binary vector.

We can also associate a Tanner graph $G = T(\mathcal{C}) = (V, E)$ with a CSS code $\mathcal{C}$. The vertices of G split into three sets $V = C_Z \sqcup Q \sqcup C_X$, where Q consists of the n qubits and C_Z and C_X consist of the m_Z and m_X stabilizer generators of the corresponding type. The edges of G are $\{i, c\}$ where $i \in Q$, $c \in C_Z \sqcup C_X$, and the qubit i is in the support of the check c. Thus, G has constant degree when the code is LDPC. We will sometimes work with the subgraphs G_Z and G_X, which are the bipartite graphs generated by $C_Z \sqcup Q$ and $Q \sqcup C_X$, respectively.

We will use the convention that qubits in Q are labeled i, checks in C_X are labeled j, and checks in C_Z are labeled k. The supports of X and Z stabilizers will be denoted f and g, respectively, with generators labeled f_j and g_k. We use the notation $N_X(\cdot)$ to denote the neighborhood in G_X. For example, $N_X(i)$ is a subset of X checks.

II.2. Code constructions

In this work, we will consider several families of quantum LDPC codes, including the surface code [44], HGP codes [10], lifted/balanced product codes [12, 13], and bivariate bicycle (BB) codes [6].

An HGP code [10] $\mathcal{C}$ is defined by two classical codes C, C' with parity-check matrices $H \in \mathbb{F}_2^{B\times A}$, $H' \in \mathbb{F}_2^{B'\times A'}$, respectively. The qubits of $\mathcal{C}$ are $A \times A' \sqcup B \times B'$, the X checks are associated with $A \times B'$, and the Z checks are associated with $B \times A'$. The parity check matrices of $\mathcal{C}$ are $H_X = \left[\, \mathbb{1}_A \otimes H' \,\middle|\, H^\top \otimes \mathbb{1}_{B'} \,\right]$ and $H_Z =$

$[\, H \otimes \mathbb{1}_{A'} \,|\, \mathbb{1}_B \otimes H'^{\top} \,]$. Alternatively, whenever $\{a, b\}$ and $\{a', b'\}$ are edges in the Tanner graphs of the respective classical codes, then $\{(a, a'), (a, b')\}$, $\{(b, b'), (a, b')\}$ are edges in G_X and $\{(b, a'), (a, a')\}$, $\{(b, a'), (b, b')\}$ are edges in G_Z.

The surface code [44] can be thought of as the hypergraph product of two classical repetition codes. We will also consider the more qubit-efficient, rotated version of the surface code.

A lifted/balanced product code [12, 13] $\overline{\mathcal{C}}$ is a quotient of an HGP code $\mathcal{C}$. More precisely, let $\mathcal{C}$ be defined by classical codes C and C'. Suppose C has a right group action for a group G; that is, A and B both have right G-actions and $\{ag, bg\}$ is an edge in in $T(C)$ whenever $\{a, b\}$ is an edge in $T(C)$. Similarly, suppose C' has a left G-action. Then $\overline{\mathcal{C}}$ is defined by identifying the qubit or check (xg, y) with (x, gy) for all $g \in G$, where $x \in A \sqcup B$ and $y \in A' \sqcup B'$. Thus, the number of qubits, X checks, and Z checks are each reduced by a factor of $|G|$ if the action is free.

A class of lifted product codes are BB codes [6], which are defined by a quotient over the abelian group $\mathbb{Z}_\ell \times \mathbb{Z}_m$. They can be alternatively be described by parity-check matrices $H_X = [\, h \,|\, h' \,]$, $H_Z = [\, h'^{\top} \,|\, h^{\top} \,]$, where $h, h' \in \mathbb{F}_2[x, y]/\langle x^\ell - 1, x^m - 1\rangle$ are bivariate polynomials which are typically the sums of three monomials. Concretely, $x = S_\ell \otimes I_m$ and $y = I_\ell \otimes S_m$, where $S_r \in \mathbb{F}_2^{r \times r}$ denotes a cyclic shift matrix.

II.3. LP decoder

In the decoding problem, some unknown (Pauli) error E is applied to the code state. By measuring stabilizers, we obtain the syndrome, which is a classical bit string s telling us which generators anticommute with E. A decoder is a classical algorithm that takes s as input and outputs a correction $\tilde{E}$. The algorithm succeeds if applying $\tilde{E}$ to the corrupted state returns the original code state, i.e., if $\tilde{E}E \in S$.

For CSS codes, we can consider decoding X and Z errors separately—although this may not be optimal as it ignores correlations due to Y errors. For Z errors, we identify E with its support as a vector $e \in \mathbb{F}_2^n$, and the relevant syndrome will be obtained from the X stabilizer generators as $s = H_X e \in \mathbb{F}_2^{m_X}$. Using s, the decoder should output $\tilde{e}$ such that $e + \tilde{e} \in \operatorname{row} H_Z$. As the X errors are decoded similarly, we will only consider Z errors.

For independent and identically distributed Z errors, the most likely error giving a syndrome s is the one of lowest Hamming weight. This can be expressed as the solution of the (nonlinear) IP

$$\begin{aligned} \min_{x \in \{0,1\}^n} \quad & \sum_{i \in Q} x_i \\ \text{s.t.} \quad & H_X x = s \mod 2\,. \end{aligned} \tag{1}$$

As IPs are generally intractable [45], a *linear programming (LP) relaxation* can instead be considered. An LP is an optimization problem with real-valued variables such that the objective function is linear and all constraints are linear inequalities. Such problems can be solved efficiently [46]. Ref. [33] considered the following LP to solve the decoding problem:

$$\begin{aligned} \min \quad & \sum_{i \in Q} x_i && \\ \text{s.t.} \quad & \sum_{S \in E_j^{s_j}} w_{j,S} = 1 && \forall j \in C_X\,, \\ & \sum_{\substack{S \in E_j^{s_j} \\ S \ni i}} w_{j,S} = x_i && \forall i \in Q,\ j \in C_X,\ \{i, j\} \in E\,, \\ & x_i \geq 0 && \forall i \in Q\,, \\ & w_{j,S} \geq 0 && \forall j \in C_X,\ S \in E_j^{s_j}\,. \end{aligned} \tag{2}$$

Here, for any $j \in C_X$, we define

$$E_j^{s_j} = \{S \subseteq f_j : |S| = s_j \mod 2\} \tag{3}$$

as the subsets with odd (when $s_j = 1$) or even (when $s_j = 0$) cardinality (including the empty set $\emptyset \in E_j^0$). In addition to x_i, the optimization problem includes auxiliary variables $w_{j,S}$ for $j \in C_X$ and $S \in E_j^{s_j}$. When $w_{j,S} \in \{0, 1\}$, it corresponds to a binary decision variable indicating the subset of f_j that is believed to be corrupted, i.e.,

$$w_{j,S} = \begin{cases} 1 & x|_{f_j} = S\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{4}$$

We can now see why the optimization (2) is a linear relaxation of the original IP (1). Suppose the last two constraints of the LP (2) are replaced by their binary versions $x_i, w_{j,S} \in \{0, 1\}$. Then the second constraint relates the auxiliary variables to the original variables x_i according to Eq. (4), and the first constraint enforces the parity of the error at each check. Hence, the x_i variables of the optimal integral solution are the same as in the IP.

Allowing $x_i, w_{j,S}$ to take real values in $[0, 1]$ yields an LP with number of variables and constraints scaling linearly with n. If the optimal solution (x^*, w^*) obtained from the LP is integral, then x^* is also the optimal solution of the IP. Such a solution would provide a certificate that $\tilde{e} = x^*$ is most likely error causing the syndrome. However, because the set of feasible solutions is larger for the LP, a fractional solution may yield a smaller objective value than for the IP. In this case, we may either declare a failure or output $\tilde{e}$ obtained by independently rounding each bit of x^* [33].

II.4. Ordered statistics decoding

Ordered statistics decoding (OSD), first introduced in the context of classical codes [47, 48], is a post-processing technique that leverages soft information from decoders like BP (or for us, LP) to enhance decoding accuracy. In quantum error correction, OSD post-processing on a potentially non-covergent BP output ensures a recovery operator that always maps the corrupted state back to the codespace [22, 23]. This approach has been shown to improve the performance of standard BP on quantum LDPC codes by several orders of magnitude [22].

Let $x \in [0,1]^n$ be a vector representing probabilities of Z errors, which may be obtained from another decoder. The idea behind OSD is to commit the bits that are likely not erroneous, erase the rest, and use Gaussian elimination to find a solution consistent with the syndrome, which hopefully has low weight.

More precisely, we sort the qubits by values of x_i in decreasing order and let $\mathcal{S}$ be the first $m_X = \operatorname{rank}(H_X)$ qubits that correspond to linearly independent columns of H_X. (By removing rows from H_X, we may assume without loss of generality that H_X is full rank.) Let $\mathcal{T}$ be the rest of the qubits. In zeroth-order OSD (OSD-0), the correction $\tilde{e}$ is defined by $\tilde{e}_{\mathcal{T}} = 0$ and $\tilde{e}_{\mathcal{S}} = H_{\mathcal{S}}^{-1} s$. In higher-order OSD methods, we try solutions where $\tilde{e}_{\mathcal{T}}$ has a few bits set to one and output the solution where $(\tilde{e}_{\mathcal{S}}, \tilde{e}_{\mathcal{T}})$ has the lowest overall Hamming weight. For example, in the combination sweep strategy (OSD-CS) [23], all weight-one configurations of $\tilde{e}_{\mathcal{T}}$ as well as weight-two configurations limited to the first λ bits are considered, where λ is a parameter to be chosen. The remaining bits $\tilde{e}_{\mathcal{S}}$ are chosen so that the syndrome is satisfied, i.e., $\tilde{e}_{\mathcal{S}} = H_{\mathcal{S}}^{-1} s + H_{\mathcal{S}}^{-1} H_{\mathcal{T}} \tilde{e}_{\mathcal{T}}$. Higher-order OSD generally leads to improved accuracy compared to OSD-0 at the cost of increased runtime. In OSD-CS, the parameter λ is often set to a fixed value, such as 60, to balance the two considerations.

III. CONSTANT-WEIGHT UNCORRECTABLE ERRORS USING LP DECODING

We describe error patterns of constant weight in various quantum LDPC codes that result in fractional solutions to the LP relaxation (2). Such error patterns cannot be corrected by the LP decoder without rounding, and we show that independent rounding of the fractional points often does not lead to a valid correction either. This means that the LP decoder would not give a threshold for these codes, as there would be a constant probability of an uncorrectable error regardless of code size. To explain these error patterns, we use an equivalent "error-based" formulation of the LP [34] and consider its dual optimization problem. The error-based LP is the one typically studied in the classical setting [28], as we have access to the individual corrupted bits. We consider this formulation to adapt techniques that have been developed for analyzing classical codes.

III.1. Error-based LP

To use the error-based method of formulating the LP, we first find any error e' satisfying the syndrome s, i.e., $H_X e' = s$. Then we solve the following LP:

$$\begin{aligned}
\min \quad & \sum_{i \notin e'} x_i - \sum_{i \in e'} x_i && (5)\\
\text{s.t.} \quad & \sum_{S \in E_j^0} w_{j,S} = 1 && \forall j \in C_X\,,\\
& \sum_{\substack{S \in E_j^0 \\ S \ni i}} w_{j,S} = x_i && \forall i \in Q,\ j \in C_X,\ \{i,j\} \in E\,,\\
& x_i \geq 0 && \forall i \in Q\,,\\
& w_{j,S} \geq 0 && \forall j \in C_X,\ S \in E_j^0\,.
\end{aligned}$$

In the objective function, we identified e' as the subset of indices i with $e'_i = 1$. When restricting to integral variables x_i, $w_{j,S}$, the feasible solutions for x are the ones satisfying all the parity checks of the code. Therefore, the optimal integral solution is the closest codeword to e', and the lowest-weight error for the syndrome s is $e' - x^* \pmod 2$. A decoder using the error-based LP would output the solution $\tilde{e}$ with $\tilde{e}_i = x_i^*$ when $e'_i = 0$ and $\tilde{e}_i = 1 - x_i^*$ when $e'_i = 1$. These formulas are also valid when x^* is a fractional solution, but $\tilde{e}$ would need to be rounded.

The error-based LP relaxation is equivalent to the syndrome-based one [33, 35]. This equivalence is formally stated in the following lemma, with a proof provided in Appendix A for completeness.

Lemma III.1. *Let $e \in \mathbb{F}_2^n$ be a Z error with syndrome $s = H_X e$ and e' be any error with the same syndrome. Then the LP (5), defined with e', is equivalent to the LP (2). More precisely, there is a bijection between feasible solutions given by the reflections*

$$x_i = \begin{cases} x'_i, & i \notin e' \\ 1 - x'_i, & i \in e' \end{cases} \quad \forall i \in Q\,, \qquad (6)$$

$$w_{j,S} = w'_{j,S+U_j} \quad \forall j \in C_X,\ S \in E_j^{s_j}\,, \qquad (7)$$

where x_i,$w_{j,S}$ are variables of the LP (2), x'_i,$w'_{j,S}$ are variables of the LP (5), and $U_j = e'|_{f_j}$. Two solutions related by the bijection have the same objective values up to an additive constant $|e'|$. Furthermore, the LP decoders based on these solutions give the same output.

Associated with any LP is a dual LP. For the error-based LP (5), the dual LP has variables σ_j for $j \in C_X$ and τ_{ij} for $i \in Q$ and $j \in C_X$ such that $\{i,j\} \in E$. It is given as follows:

$$\begin{aligned} \max \quad & \sum_{j\in C_X} \sigma_j && && (8)\\ \text{s.t.} \quad & \sum_{j\in N_X(i)} \tau_{ij} \le \gamma_i && \forall i \in Q\,, &&\\ & \sum_{i\in S} \tau_{ij} \ge \sigma_j && \forall j \in C_X,\ S \in E_j^0\,, && \end{aligned}$$

where $\gamma_i = (-1)^{e'_i}$.

A standard result from optimization states that the primal and dual LPs have the same objective value [46].

Lemma III.2. *Let $x^* = (x_1^*, \ldots, x_n^*)$ and $\sigma^* = (\sigma_1^*, \ldots, \sigma_{m_X}^*)$ be the optimal solutions of the LPs (5) and (8), respectively. Then*

$$\sum_{i\notin e'} x_i^* - \sum_{i\in e'} x_i^* = \sum_{j\in C_X} \sigma_j^*\,. \tag{9}$$

III.2. Problematic error configurations

Let e be a reduced error, that is, one with the lowest Hamming weight out of the equivalent errors obtained by adding stabilizers. Because the error-based LP (2) is equivalent to the syndrome-based LP (5) regardless of the representative e' chosen (Lemma III.1), we may assume $e' = e$. The reduced error conditions means that the optimal integral solution of the LP (5) is 0 (for example, with the solution $x = 0$).

For the different error patterns we construct, we argue by contradiction to show that they are not correctable by LP. If the LP decoder succeeds without rounding, the optimal solution must not be fractional. Therefore, the optimal value of the LP (5) is 0. By Lemma III.2, so is the optimal value of the dual LP (8). When this is the case, each σ_j must be 0 because the second constraint with $S = \emptyset$ implies that the variables σ_j are non-positive. When $\sigma_j = 0$, the second constraint can alternatively be written as $\tau_{ij} + \tau_{i'j} \ge 0$ for all $j \in C_X$ and $i, i' \in f_j$ with $i \neq i'$. Therefore, we obtain a simplified set of inequalities.

Proposition III.3. *If the LP (2) outputs an integral solution for a reduced error e, there exist edge weights τ_{ij} that satisfy the inequalities*

$$\sum_{j\in N_X(i)} \tau_{ij} \le \gamma_i \quad \forall i \in Q\,, \tag{10}$$

$$\tau_{ij} + \tau_{i'j} \ge 0 \quad \forall j \in C_X,\ i, i' \in f_j,\ i \neq i'\,. \tag{11}$$

One intuitive way to understand the above inequalities is in terms of hyperflow of "poison" [49]. Each corrupted qubit $i \in e$ has one unit of poison that must be transported away. In order to transport some amount of poison $\beta_j = -\tau_{ij}$ through a neighboring check $j \in C_X$, it has to spread to all other neighbors of j ($\tau_{i'j} = \beta_j$ for $i' \in f_j \setminus \{i\}$). Each uncorrupted qubit $i \in Q \setminus e$ can absorb one unit of poison. Inequalities (10) and (11) have a solution if and only if there is a valid hyperflow. Using this intuition, we prove a fact about errors that are half of a Z stabilizer.

Lemma III.4. *Let g be a Z stabilizer of even weight and e be a reduced error such that $|e \cap g| = |g|/2$. If the LP (5) outputs an integral solution, then any edge weights satisfying Inequalities (10) and (11) must saturate the first inequality for all $i \in g$.*

Proof. Because any $j \in C_X$ commutes with g, the overlap $|f_j \cap g|$ must be even. In particular, it must be at least 2 for any $j \in N_X(g)$. This means whenever we try to move poison away from a qubit in g, it must be distributed somewhere else in g (and possibly to other qubits). Since we need to move $|e \cap g| = |g|/2$ units of poison away from the corrupted vertices in g and there are only $|g \setminus e| = |g|/2$ uncorrupted vertices to absorb the poison in g, the first inequality (10) for all $i \in g$ must be saturated. □

We are now ready to present several configurations of small uncorrectable errors which can occur in codes with stabilizers satisfying certain conditions.

Proposition III.5. *Let g, g' be two Z stabilizers of even weight such that $|g \cap g'| \ge 2$. Furthermore, suppose all X checks $j \in N_X(g \cap g')$ satisfy $f_j \cap (g + g') \neq \emptyset$. Then the LP decoder (2) cannot correct a reduced error e defined with $|\, e|_{g\setminus g'}\,| = \lfloor |g \setminus g'|/2 \rfloor$, $|\, e|_{g'\setminus g}\,| = \lceil |g' \setminus g|/2 \rceil$, and $|\, e|_{g\cap g'}\,| = 1$.*

Proof. Assume for a contradiction that e can be corrected by the LP. Because $|e \cap (g + g')| = |g + g'|/2$, Lemma III.4 implies Inequality (10) is saturated for all $i \in g + g'$. But to remove the one unit of poison in $g \cap g'$, it must go through an X check $j \in N_X(g \cap g')$. This check also has support on $g + g'$ and will therefore spread extra poison to the set, rendering Inequality (10) impossible for some $i \in g + g'$. □

The above proposition states that there will typically be an uncorrectable error pattern if two Z stabilizers overlap on at least two qubits. By commutation with Z stabilizers, f_j must have at least weight-two support on g and g'. The X-check condition expresses that its support within $g \cup g'$ cannot be limited to $g \cap g'$. The weight of the error constructed cannot be reduced by adding g, g', or $g + g'$. Given the many ways to choose e satisfying the conditions, it is likely that one choice will be reduced and thus uncorrectable by LP.

We can explicitly demonstrate a fractional solution of the LP (2) with objective value smaller than $|e|$ that causes the LP decoder to fail. Set

$$x_i = \begin{cases} 1/2, & i \in g + g', \\ 0, & \text{otherwise}\,, \end{cases} \tag{12}$$

which gives objective value $|g + g'|/2 = |e| - 1$. To define the $w_{j,S}$ variables, note that for any check j, the overlap

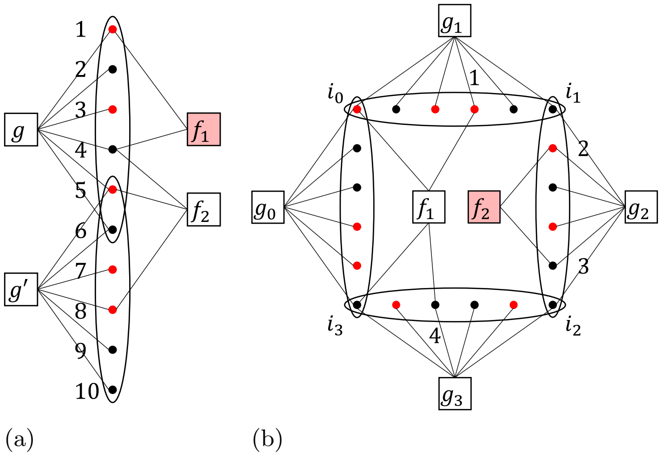


FIG. 1: Error patterns that are uncorrectable by LP (2). In the parts of the Tanner graphs illustrated, dots represent qubits with Z errors affecting those in red, and squares represent checks with the violated X checks highlighted. (a) Proposition III.5. An objective value of 4 is achieved with $x_i = 1/2$ for $i = 1, 2, 3, 4, 7, 8, 9, 10$ and $w_{f_1,\{1\}} = w_{f_1,\{4\}} = w_{f_2,\{4,8\}} = w_{f_2,\emptyset} = 1/2$. (b) Proposition III.6. An objective value of 8 is achieved with $x_i = 1/2$ for $i \neq i_0, i_1, i_2, i_3$ and $w_{f_1,\{1,4\}} = w_{f_1,\emptyset} = w_{f_2,\{2\}} = w_{f_2,\{3\}} = 1/2$.

$S_j := f_j \cap (g + g')$ has even cardinality due to commutativity of the checks. Suppose $|S_j| \geq 2$. If $s_j = 0$, we set $w_{j,S_j} = w_{j,\emptyset} = 1/2$. If $s_j = 1$, let $i \in S_j$, and we set $w_{j,\{i\}} = w_{j,S_j\setminus\{i\}} = 1/2$. When $S_j = \emptyset$, the set f_j cannot have support on $g \cap g'$ either by the conditions of the proposition, so $s_j = 0$, and we define $w_{j,\emptyset} = 1$. All other $w_{j,S}$ are set to 0. Fig. 1(a) illustrates an example of the error pattern in Proposition III.5.

More generally, we have uncorrectable errors associated with cycles in Z Tanner graph G_Z.

Proposition III.6. *Let $g_0, g_1, \ldots, g_{K-1}, g_K = g_0$ be Z stabilizers of even weight with intersections $g_k \cap g_{k+1} = \{i_k\}$ for all k and $g_k \cap g_{k'} = \emptyset$ for $|k-k'| \geq 2$. Suppose all X checks $j \in N_X(I)$ satisfy $f_j \cap (\sum_{k=0}^{K-1} g_k) \neq \emptyset$, where $I = \{i_k\}_{k=0}^{K-1}$. Then the LP decoder (2) cannot correct a reduced error e defined with $e|_I = i_0$ and $|\, e|_{g_k} | = |g_k|/2 - 1$ for all k.*

Proof. Suppose for a contradiction that e can be corrected by the LP. Because $|e \cap \sum_{k=0}^{K-1} g_k| = |\sum_{k=0}^{K-1} g_k|/2$, Lemma III.4 implies that Inequality (10) is saturated for all $i \in \sum_{k=0}^{K-1} g_k$. Now consider the poison on i_0. It must go through an X check $j \in N_X(i_0)$. Because f_j also has support on $\sum_{k=0}^{K-1} g_k$, it will spread too much poison to that set, and Inequality (10) will be violated for some qubit in $\sum_{k=0}^{K-1} g_k$. □

We remark that the even-weight stabilizers condition in the proposition is not restrictive because we can always add two odd-weight stabilizer generators to obtain an even-weight stabilizer. Again, no stabilizer generated by the $\{g_k\}$ can reduce the weight of e as defined, so e will generically be reduced.

A fractional solution of the syndrome-based LP (2) is obtained by setting

$$x_i = \begin{cases} 1/2, & i \in \sum_{k=0}^{K-1} g_k, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

This gives objective value $\frac{1}{2}|\sum_{k=0}^{K-1} g_k| = |e| - 1 < |e|$. Note that for any check j, the overlap $S_j := f_j \cap \sum_{k=0}^{K-1} g_k$ has even cardinality. This allows us to define the auxiliary variables $w_{j,S}$ as follows. When $|S_j| \geq 2$ and $s_j = 0$, we set $w_{j,S_j} = w_{j,\emptyset} = 1/2$. When $|S_j| \geq 2$ and $s_j = 1$, we let $i \in S_j$ and set $w_{j,\{i\}} = w_{j,S_j\setminus\{i\}} = 1/2$. When $S_j = \emptyset$, we must have $s_j = 0$, so we set $w_{j,\emptyset} = 1$. All other $w_{j,S}$ are set to 0. See Fig. 1(b) for an example of the error pattern in Proposition III.6.

Because the fractional solutions in Propositions III.5 and III.6 have variables $x_i = 1/2$, independent rounding will not give a valid solution most of the time, as it is unclear whether to round each coordinate to 0 or 1.

In Proposition III.6, the Z stabilizers give the cycle $0, i_0, 1, i_1, \ldots, K-1, i_{K-1}, K = 0$ in G_Z when the g_k are generators. If the condition $g_k \cap g_{k'} = \emptyset$ for $|k - k'| \geq 2$ is not satisfied, we could obtain a smaller cycle in G_Z that we can apply the proposition to. The case where $|g_k \cap g_{k+1}| \geq 2$ is covered by Proposition III.5 (which also gives the cycle k_0, i_0, k_1, i_1, k_0 in G_Z, where k_0 and k_1 label the Z stabilizer generators g and g', respectively, and $i_0, i_1 \in g \cap g'$). In summary, cycles in G_Z typically give rise to error patterns that are not correctable by LP.

This observation implies that quantum LDPC codes can exhibit small, uncorrectable errors under LP decoding, with the size of such errors determined by the girth (i.e., the length of the shortest cycle) of the Tanner graph G_Z. How large can this girth be? A standard counting argument shows that the girth of G_Z is at most $O(\log n)$. Because each stabilizer generator has constant size, the problematic error patterns arising from the shortest cycle of G_Z will also have weight $O(\log n)$. We also note that for cycles of size greater than the stabilizer weights, the condition $f_j \cap \sum_{k=0}^{K-1} g_k \neq \emptyset$ for $j \in N_X(I)$ is automatically satisfied since otherwise, commutation with each of the g_k would imply $f_j \supseteq I$. Thus, the LP decoder in general cannot correct adversarial errors of weight $\omega(\log n)$. Moreover, if it could correct stochastic errors, the suppression of errors below the threshold would be at most polynomial (instead of exponential) in the blocklength.

Some quantum LDPC codes do have logarithmically scaling girths in their X and Z Tanner graphs [50]. However, many code constructions exhibit *constant* girth and would likely support constant-sized uncorrectable errors. In the next section, we present examples of such codes.

III.3. Uncorrectable errors in different code families

Using Propositions III.5 and III.6, we show that many families of quantum LDPC codes exhibit problematic structures in their Tanner graphs, giving strong evidence that LP by itself does not have a threshold for these codes. To do this, we exhibit constant-sized cycles in G_Z and show that the condition on the X checks is satisfied. Without knowing the specifics of the code, we often cannot certify that error e constructed is reduced, as even though combinations of Z generators from the cycle cannot reduce the error weight, other Z stabilizers might. However, given that many constant-sized cycles are present in the code and e may be defined in many ways from a given cycle, it is unlikely that no such errors would be reduced.

Consider an HGP code $\mathcal{C}$ from two classical LDPC codes C, C'. For simplicity, assume that the generator weights of $\mathcal{C}$ are even. Otherwise, we can make the same argument with Z stabilizers which are sums of two generators. We exhibit a constant-length cycle in the Z Tanner graph by considering two cases.

First, suppose $T(C)$ and $T(C')$ have girths greater than six. Let $P = b_1, a_1, b_2, a_2, b_3$ and $P' = a'_1, b'_1, a'_2, b'_2, a'_3$ be paths in the $T(C)$ and $T(C')$, respectively, where each $a_i \in A$, $a'_i \in A'$, $b_i \in B$, and $b'_i \in B'$. Consider the cycle Γ in G_Z obtained by concatenating $P \times a'_1$, $b_3 \times P'$, $P \times a'_3$, and $b_1 \times P'$, i.e.,

$$(b_1, a'_1), (a_1, a'_1), (b_2, a'_1), (a_2, a'_1), (b_3, a'_1), (b_3, b'_1), (b_3, a'_2), (b_3, b'_2), (b_3, a'_3), (a_2, a'_3), (b_2, a'_3), (a_1, a'_3), (b_1, a'_3), (b_1, b'_2), (b_1, a'_2), (b_1, b'_1), (b_1, a'_1)\,. \tag{14}$$

We show that the intersections of Z generators from Γ are exactly the qubits in the cycle. For two generators to overlap, either the first or the second coordinate must be the same. Without loss of generality, suppose $(b_i, a'), (b_j, a') \in \Gamma$ have a common neighbor $(a, a') \notin \Gamma$. Then the subpath of P from b_i to b_j concatenated with b_j, a, b_i would give a cycle in $T(C)$ of length at most six, contradicting its girth.

Now, suppose one of the Tanner graphs of the classical codes, say $T(C)$, has girth at most six. Let $P = b_1, a_1, \ldots, b_r = b_1$ be a minimal cycle, where each $a_i \in A$ and $b_i \in B$ and $r \leq 3$. Then $P \times a'$ for any $a' \in A$ gives a cycle in G_Z. Intersections of two generators from this cycle cannot overlap on other qubits due to the minimality of the cycle P.

Let I be the qubits in Γ. There is no X check that contains I since the qubits in $A \times A'$ do not all have the same first coordinate. Therefore, any X check $j \in N_X(I)$ intersects the sum of the Z checks in the cycle due to commutation with each Z check. Aside from the reduced error condition, Proposition III.6, or Proposition III.5 if $r = 2$ in the second case, gives a constant-weight error supported on the Z generators of the cycle that cannot be corrected by the LP decoder. See Fig. 2 for an example.

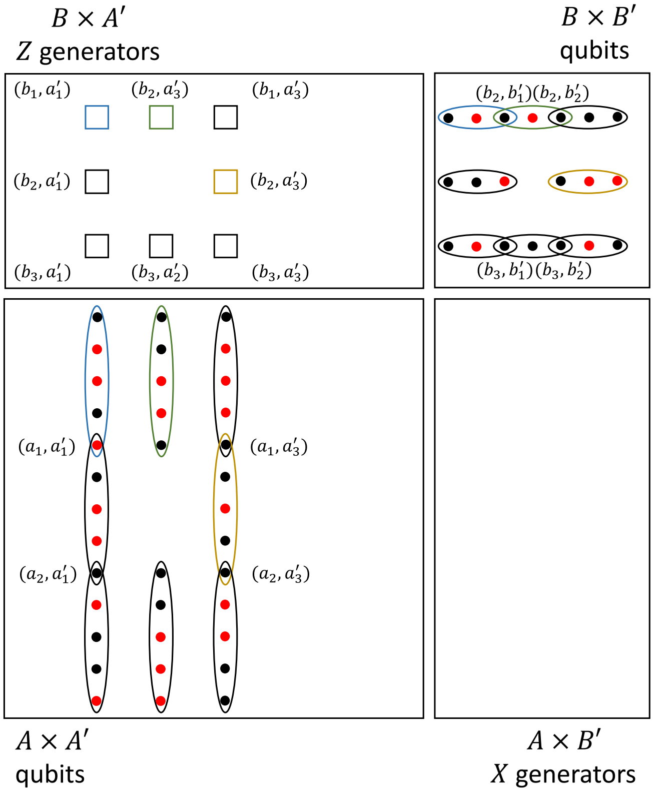


FIG. 2: A constant weight error pattern in an HGP code that cannot be corrected using LP. The cycle (14) is labeled, and some of the Z generators and their qubit supports are shown with different matching colors. Corrupted qubits are colored in red.

Because lifted product codes are quotients of HGP codes, they also typically contain constant-weight uncorrectable errors. Let $\overline{\mathcal{C}}$ be a lifted product code with check weights at most $w = O(1)$, which is obtained as the quotient of an HGP code $\mathcal{C}$ by a group G. Any cycle in the Z Tanner graph of $\mathcal{C}$ projects to a cycle in the Z Tanner graph of $\overline{\mathcal{C}}$. In particular, consider the cycle in $\mathcal{C}$ created from the two paths $P = b_1, a_1, \ldots, b_r$ and $P' = a'_1, b'_1, \ldots, a'_r$ by concatenating $P \times a'_1$, $b_r \times P'$, $P \times a'_r$, and $b_1 \times P'$. If we assume that the projection does not identify two distinct points in the cycle and that the projected cycle in $T(\overline{\mathcal{C}})$ is minimal, we obtain an uncorrectable error pattern. By choosing $r = w/4 + 2$, the X-check condition is automatically satisfied as the cycle has length $4(r-1) > w$.

In particular, because the surface code is an HGP code and BB codes are lifted product codes, they also contain constant-weight error patterns that are not correctable by LP.

We remark that in a recent work, Grouès [35] has independently discovered a different type of constant-weight uncorrectable error pattern in certain HGP codes, which is also related to the presence of Z stabilizers but has no immediate connection to cycles in the Tanner graph.

IV. LP+OSD DECODER

We now describe our proposed decoder based on LP and OSD post-processing to decode quantum LDPC codes. For a CSS code with parity check matrices H_X, H_Z, suppose we are given the syndrome s of a Z error. We first run the syndrome-based LP (2) to obtain a solution x. If x is integral, then it is guaranteed to be the minimum-weight error giving the syndrome s, so we output this solution. Otherwise, x contains fractional components. To obtain an integral solution satisfying the syndrome s, we apply OSD post-processing, where we interpret the fractional components x_i as the probability that qubit i suffered a Z error. Either zeroth- or higher-order OSD may be used, depending on the accuracy desired.

One thing to note is that a fractional solution from the LP often contains many entries which are zero. Thus, there is freedom in ordering the corresponding qubits for OSD post-processing, which may affect the rounded solution. In the examples of LP decoding failure found in Sec. III, the LP assigns zero to one qubit in the error set, and it would be beneficial if this coordinate was erased by OSD. This qubit would typically be connected to a nontrivial syndrome in the Tanner graph. Therefore, we use the heuristic that qubits which are close to the nonzero syndromes are more likely to have an error than those which are farther away. When ordering the qubits according to probability of being flipped for OSD, we break ties by the distance to a nonzero syndrome. Pseudocode for our algorithm is given in Algorithm 1.

Algorithm 1 LP+OSD Decoder

```
Input: Parity-check matrix $H_X$, syndrome $s$ of a $Z$ error
Output: Correction $\tilde{e}$
$x \leftarrow$ solution to the LP (2)
if $x$ is integral then
    return $x$    ▷ minimum-weight error consistent with $s$
else
    for all $i \in Q$ do
        $d_i \leftarrow \mathrm{dist}_{G_X}(i, \{j \in C_X : s_j = 1\})$
    end for
    $Q' \leftarrow$ qubits sorted by descending order of $x_i$, breaking ties by ascending order of $d_i$
    $\tilde{e} \leftarrow$ OSD-0($s$) or OSD-CS($s$)    ▷ $\mathcal{S}, \mathcal{T}$ defined using $Q'$
    return $\tilde{e}$
end if
```

The decoder as described works in the code capacity setting. However, it can easily be adapted to handle phenomenological or circuit-level noise; see Appendix B.

The worst-case complexity of our LP+OSD decoder is $n^{\omega+o(1)}$ in the asymptotic limit, where $\omega \approx 2.37$ is the matrix multiplication exponent [51]. Solving the LP requires $n^{\omega+o(1)}$ time for an n-qubit LDPC code [52]. In the OSD subroutine, the decoder first ranks bits based on probability and, in case of ties, computes shortest distances via Dijkstra's algorithm. Thus, it takes time $O(n \log n)$ to determine $\mathcal{S}$ and $\mathcal{T}$ [53]. We then precompute $H_{\mathcal{S}}^{-1}$, which is done in time $n^{\omega+o(1)}$ [54]. If we run OSD-0, we compute $\tilde{e}_{\mathcal{S}} = H_{\mathcal{S}}^{-1} s$ in time $O(n^2)$. If we run OSD-CS with a constant parameter $\lambda = O(1)$, we need to compute $\tilde{e}_{\mathcal{S}} = H_{\mathcal{S}}^{-1} s + H_{\mathcal{S}}^{-1} H_{\mathcal{T}} \tilde{e}_{\mathcal{T}}$ for all the $O(n)$ candidate values of $\tilde{e}_{\mathcal{T}}$. This can be done in time $n^{\omega+o(1)}$ if we compute $H_{\mathcal{S}}^{-1} H_{\mathcal{T}} \tilde{E}_{\mathcal{T}}$, where $\tilde{E}_{\mathcal{T}}$ is a matrix with columns that are the different values of $\tilde{e}_{\mathcal{T}}$. Picking the solution where $(\tilde{e}_{\mathcal{S}}, \tilde{e}_{\mathcal{T}})$ has the lowest Hamming weight takes time $O(n^2)$. Therefore, the total running time is $n^{\omega+o(1)}$, although the optimized matrix multiplication algorithms have extremely large constant overheads. For practical purposes, conventional $O(n^3)$ matrix multiplication and inversion would be used, and the LP would be solved using the simplex algorithm, which is often fast but has exponential complexity in the worst case.

IV.1. Problematic error patterns under LP+OSD

Now we show how OSD post-processing can help in correcting the problematic error patterns in Sec. III. First, let e be defined as in Proposition III.5 from constant-weight stabilizers g and g'. Consider x from Eq. (12), which is a fractional solution to the syndrome-based LP (2). When OSD from Sec. II.4 is applied to this solution, it will erase a set of qubits $\mathcal{S}$, starting from the elements of $g + g'$ because they are assigned higher probability of being corrupted by the LP. The fact that the columns of H_X corresponding to $\mathcal{S}$ are linearly independent is equivalent to $\mathcal{S}$ not containing any Z stabilizers or Z logical operators. Assuming that the code has macroscopic distance and that there are no other stabilizer generators contained in $g \cup g'$, the only relevant stabilizers are g and g'. Therefore, $\mathcal{S}$ contains all qubits of $g + g'$ except one, which we denote i_1. Without loss of generality, we can assume that $i_1 \notin e$; otherwise, the same analysis would hold for the equivalent error $e+g+g'$ and we would show that OSD returns this error instead.

Consider the case when $\mathcal{S}$ contains the corrupted qubit $i_0 \in g \cap g'$. Then $e \subseteq \mathcal{S}$, so e is a valid solution for the syndrome that is consistent with the unerased qubits set to 0. Because there is a unique solution satisfying the syndrome such that $\tilde{e}_{\mathcal{T}} = 0$, OSD-0 will return the actual error $\tilde{e} = e$.

If $\mathcal{S}$ does not contain the qubit i_0, OSD-0 might not find the solution e because it will set $\tilde{e}_{i_0} = 0$ (although it may find a stabilizer-equivalent solution). However, when we search over weight-one configurations of $\tilde{e}_{\mathcal{T}}$ in OSD-CS, one of the them will have $\tilde{e}_{i_0} = 1$ and $\tilde{e}|_{\mathcal{T}\setminus\{i_0\}} = 0$. In that iteration, the solution e will be found since $e|_{\mathcal{T}} = \tilde{e}|_{\mathcal{T}}$. There are no logically inequivalent errors with lower weight as e has constant weight and the code has growing distance. Therefore, the final output of OSD-CS will be an error equivalent to e.

The error configuration of Proposition III.6 is similar. Let e be the error defined from constant-weight stabilizers $g_0, \ldots, g_{K-1}$, and consider x defined by Eq. (13). Assuming the code has macroscopic distance and no other Z stabilizers are contained in $\cup_{k=0}^{K-1} g_k$, the only relevant stabilizers are the g_k. The erased set $\mathcal{S}$ will contain all qubits in $\sum_{k=0}^{K-1} g_k$ except one, which we can assume is not an error without loss of generality by possibly replacing e with the equivalent error $e + \sum_{k=0}^{K-1} g_k$. Using the same reasoning as before, the solution e will be found by OSD-0 if $i_0 \in S$ or by OSD-CS if $i_0 \notin \mathcal{S}$.

IV.2. Comparison with other LP-based decoders

Before presenting our numerical results, we discuss the similarities and differences between our LP+OSD decoder and other LP-based decoders. For the surface code and certain other topological codes, minimum-weight perfect matching (MWPM) gives the most likely error with a given syndrome. Efficient implementations of MWPM are often based on the blossom algorithm [55–58], which is also formulated in terms of an LP relaxation of the IP that describes the minimum-weight solution of the decoding problem. In that setting, because the algorithm makes use of the special structure of the decoding problem, i.e., each error causing at most two syndromes, the algorithm always outputs the optimal solution in polynomial time.

For general quantum LDPC codes, the hypergraph minimum-weight parity factor (MWPF) decoder has been proposed as a generalization of the MWPM decoders [59]. As such, MWPF is also based on solving an LP. The constraints for the LP in MWPF correspond to larger and larger subsets of the decoding graph that are "invalid clusters" given the observed syndrome. There is an exponential number of constraints, requiring a specialized algorithm that considers a polynomial number of relevant constraints at any given time. MWPF can often find the most likely error (with a certificate of having done so), but there is no guarantee of the runtime for a general code. In contrast, the LP relaxation we consider based on Refs. [33–35], contains more fine-grained information about the local neighborhoods of each check. Because there are only a linear number of total constraints, a generic LP solver may be used for the LP portion of the decoder, and it will always terminate in polynomial time. The downside is that fractional solutions are often obtained, necessitating the need for post-processing techniques such as OSD.

V. NUMERICAL RESULTS

To evaluate the effectiveness of LP+OSD, we use it to decode the rotated surface code, random HGP codes, and BB codes. We benchmark its performance by comparing it against BP+OSD [22, 23], which is the currently the most common method of decoding general quantum LDPC codes. We consider the code capacity setting with independent Z errors of probability p and estimate the corresponding logical error rate p_L. All codes considered are symmetric under exchanging X and Z stabilizers, so it is valid to consider only decoding Z errors. Further details of the numerical simulations are discussed in Appendix C. The source code that was used to perform the simulations and the data that was collected are available in Ref. [60].

The surface code is one of the most well-studied and practical quantum codes with geometrically local stabilizer checks. However, this means that the Tanner graph of the code is not very expanding, and LP by itself is known to perform poorly on it [33].

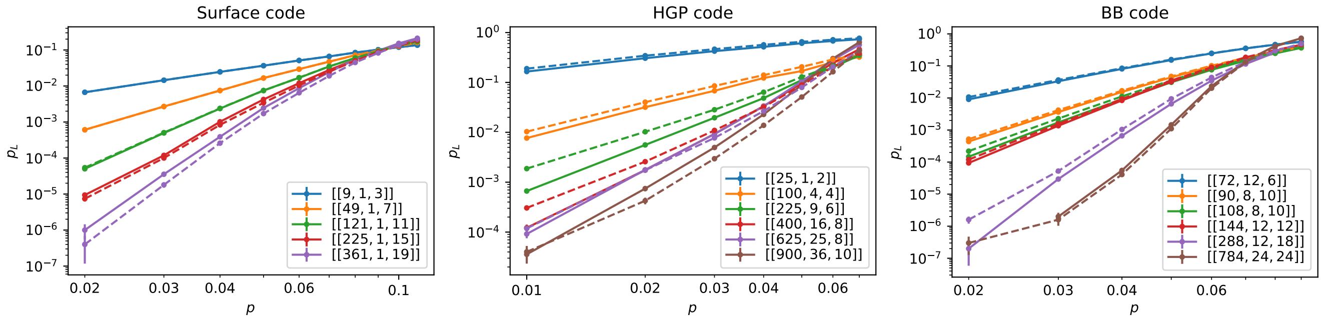
HGP codes are a family of constant-rate quantum LDPC codes which satisfy certain expansion properties when the component classical codes are sufficiently random [17]. LP can perform reasonably well by itself on such codes [33]. The HGP codes we use are randomly sampled to have good distance. In particular, they have parameters $[[n = 25s^2, k \geq s^2, d \geq d(s)]]$ for $s = 1, 2, 3, 4, 5, 6$ and corresponding lower bounds $d(s) = 2, 4, 6, 8, 8, 10$.

Another notable family of LDPC codes is the recently introduced BB codes [6]. These codes have weight-six stabilizers that can be implemented using four local interactions—matching the surface code's connectivity—and two additional nonlocal interactions. BB codes demonstrate promising parameters even at small blocklengths and have shown good performance under BP+OSD decoding. We consider BB codes introduced in Ref. [6], which have parameters $[[72, 12, 6]]$, $[[90, 8, 10]]$, $[[108, 8, 10]]$, $[[144, 12, 12]]$, $[[288, 12, 18]]$, and $[[784, 24, 24]]$.

Fig. 3 compares the LP+OSD-CS decoder with BP+OSD-CS on the three families of codes. For the surface code, LP+OSD performs comparably to BP+OSD at small code sizes (up to the $[[225, 1, 15]]$ code), while BP+OSD does better as the code size increases. In contrast, LP+OSD consistently outperforms BP+OSD for random HGP codes up to ones with parameters $[[400, \geq 16, \geq 8]]$. It also has slightly lower logical error rates than BP+OSD for BB codes up to the $[[288, 12, 18]]$ code. Small code sizes are the regime where more expensive decoders like BP+OSD or LP+OSD, which have cubic complexity, remain practically viable.

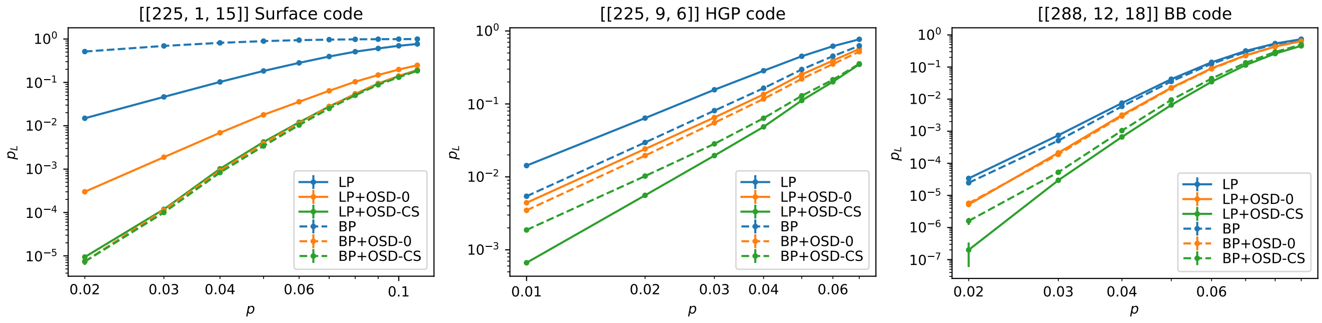
Fig. 4 presents a comparison of LP and BP decoding with independent rounding, OSD-0 post-processing, and OSD-CS post-processing for specific codes of 200-300 qubits. The relative performances vary significantly across the surface code, HGP codes, and BB codes. For both LP and BP, OSD-CS does better than OSD-0, which does better than independent rounding. However, the relative improvements from OSD-0 to OSD-CS for LP is greater than those for BP. This may be due to the fact that sometimes LP assigns a value of 0 to a qubit that does have an error (e.g., the error patterns in Sec. III), which must be searched using a higher-order OSD method.

Recall that in the first step of OSD, the qubits are



(a) (b) (c)

FIG. 3: Comparison of the LP+OSD-CS decoder (solid lines) with the BP+OSD-CS decoder (dashed lines) for (a) the surface code, (b) random HGP codes, and (c) BB codes.



(a) (b) (c)

FIG. 4: Comparison of LP and BP decoders with different OSD post-processing methods for (a) the $[[225, 1, 15]]$ surface code, (b) random HGP codes with parameters $[[225, \geq 9, \geq 6]]$, and (c) the $[[288, 12, 18]]$ BB code.

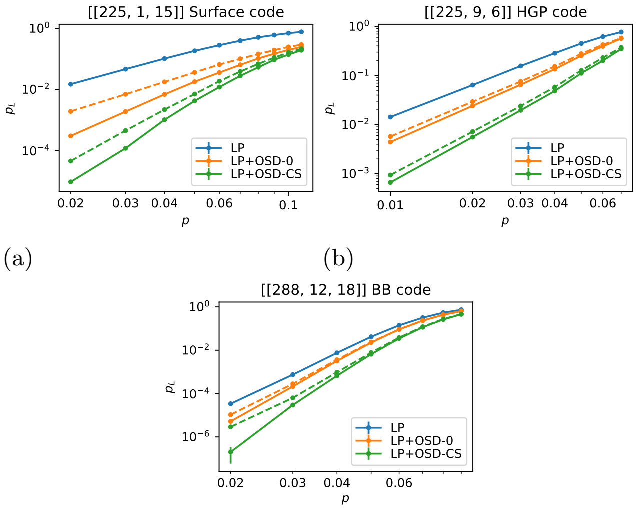
sorted according to their probability of having an error, and there may be many ties when interpreting an LP solution as probabilities. In Fig. 5, we compare two ways of breaking these ties. The plots show that for the $[[225, 1, 15]]$ surface code, random HGP codes with parameters $[[225, \geq 9, \geq 6]]$, and the $[[288, 12, 18]]$ BB code, breaking ties by distance to a nontrivial syndrome, as described in Sec. IV, yields lower logical error rates than breaking ties randomly for both OSD-0 and OSD-CS. The same conclusion holds for the other codes in the three families. Therefore, we break ties by distance to a nontrivial syndrome in our other LP+OSD simulations (Figs. 3 and 4). As expected, LP+OSD-CS outperforms LP+OSD-0, which, in turn, outperforms LP with independent rounding, regardless of the tie-breaking scheme.

Fig. 6 shows the percentage of times, conditioned on LP failure, that LP with independent rounding gives a solution not consistent with the syndrome of the error. The quantity plotted is p_{ws}/p_L, where p_{ws} is the rate of obtaining a rounded solution with the wrong syndrome. The large values of p_{ws}/p_L, especially for bigger code sizes, represent the potential gain from using a more sophisticated rounding technique and motivated the exploration of OSD as a post-processing step.

Finally, we note that our construction of uncorrectable constant-weight error patterns in Sec. III helps explain why LP+OSD-CS performs well at small and intermediate code sizes but tends to underperform as we move to larger codes. The LP solutions to these error patterns have $x_{i_0} = 0$ for a qubit i_0 that was actually corrupted. As mentioned in Section IV.1, when OSD-CS iterates over weight-one configurations of $\tilde{e}_{\mathcal{T}}$, one of the configurations will set qubit i_0 to 1. This leads to a successful recovery since $e \subseteq \mathcal{S} \cup \{i_0\}$. However, as the code size increases, the decoder may encounter multiple such problematic patterns simultaneously. When this happens, it is no longer guaranteed that the weight-one or weight-two search performed by OSD-CS will be sufficient to set $\tilde{e}_{i_0} = 1$ for the corresponding i_0 from all the problematic patterns. Consequently, OSD-CS cannot simultaneously set all the relevant qubits to 1, which limits its effectiveness as the code size grows.

VI. DISCUSSION

In this article, we studied LP decoding of quantum LDPC codes. On one hand, we showed that LP with independent rounding runs into problematic error patterns and does not admit a threshold for many fami-

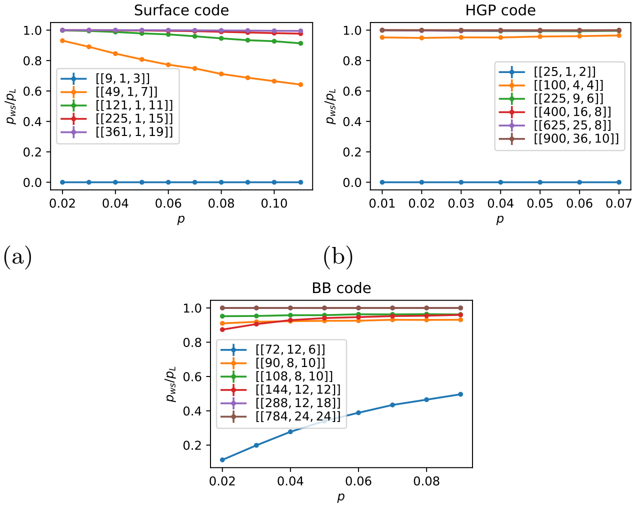


(a)

(b)

(c)

FIG. 5: Comparison of the different LP+OSD decoders. In OSD post-processing, qubits are sorted based on probabilities, with ties broken either based on the distance to a nontrivial syndrome (solid lines) or randomly (dashed lines). We plot the performance for (a) the $[[225, 1, 15]]$ surface code, (b) random HGP codes with parameters $[[225, \geq 9, \geq 6]]$, and (c) the $[[288, 12, 18]]$ BB code.



(a)

(b)

(c)

FIG. 6: Fraction of LP failures that are due to a rounded solution not satisfying the syndrome for (a) the surface code, (b) random HGP codes, and (c) BB codes.

lies of quantum codes. On the other hand, we proposed the LP+OSD decoder, which incorporates an improved rounding technique and shows promising performance for HGP and BB codes of up to a few hundred qubits.

Based on our findings, many open directions remain to be explored. Theoretically, one could try to find modifications of the LP decoder that allow for a provable threshold in certain quantum codes. For example, adding redundant parity checks or considering multiple checks simultaneously could tighten the LP relaxation to remove certain fractional solutions [32, 61], although these methods will lead to increased LP problem sizes. One could also try to find alternative LP relaxations that are aware of the stabilizer generators in a quantum code to obtain provable guarantees.

Practically, heuristic methods to improve the accuracy of LP decoding should also be explored. Taking inspiration from the classical setting, one could consider adaptive LP decoding, which incrementally adds constraints until an integral solution, representing the most likely error, is found [62]. Other possibilities include guessing facets of the LP polytope [63] and branch-and-bound methods [64]. It would be interesting to see what effect these improvements have on LP decoding of quantum codes when combined with OSD post-processing.

In terms of efficiency, LP+OSD has an asymptotic scaling that makes it infeasible for large codes. Several methods have already been developed to reduce the runtime of both LP (in the classical and quantum settings) and OSD (in the context of BP+OSD) without significant degradation in performance. The alternating direction method of multipliers (ADMM) reformulates LP decoding as a message-passing algorithm that has runtime comparable to BP but is guaranteed to converge [65], and it has been successful for decoding HGP codes [35]. BP can be used as a preprocessor that either fully corrects the error or passes a prior to LP that can reduce its runtime [66]. Several techniques for reducing the cost of OSD have also been introduced in the context of BP post-pocessing, such as ordered Tanner forest (OTF) [67] and localized statistics decoding (LSD) [68], which may be adapted to LP post-pocessing as well. Combining these approaches could reduce the runtime of LP+OSD to almost linear complexity, but one would have to evaluate if such techniques can maintain the accuracy of the decoder.

We only benchmarked the performance of LP+OSD in the code capacity setting as a proof-of-concept demonstration. For more accurate comparisons with existing decoders, it would be important to determine if, and for what code sizes, our decoder shows an advantage in the more realistic setting of circuit-level depolarizing noise (see Appendix B). Circuit-level noise may be problematic for LP+OSD since it gives rise to a decoding problem with more error mechanisms, and LP+OSD seems to perform relatively worse at larger code sizes. Nevertheless, one may benefit from applying LP decoding directly to depolarizing noise in such a scenario instead of approximating it as independent X and Z errors, as this approach can potentially capture error correlations more accurately.

In conclusion, our work enables further exploration of decoding quantum codes using LP, helping address the scarcity of accurate and efficient decoders for general quantum codes.

ACKNOWLEDGMENTS

We thank A. Kubica, A. Leverrier, C. Pattison, J. Preskill, and Y. Wu for helpful discussions and S. Puri for providing feedback on an earlier version of this manuscript. S.G. acknowledges funding from the Air Force Office of Scientific Research (FA9550-19-1-0360), the U.S. Department of Energy (DE-SC0018407, DE-AC02-07CH11359), the National Science Foundation (PHY-1733907), and the National Science Foundation's Quantum Leap Challenge Institute in Robust Quantum Simulation (OMA-2120757). M.S. is supported by AWS Quantum Postdoctoral Scholarship and funding from the National Science Foundation PHY-2317110 and NSF Career Award CCF-2048204. The Institute for Quantum Information and Matter is an NSF Physics Frontiers Center.

Appendix A: An equivalent formulation of LP

Lemma III.1. *Let $e \in \mathbb{F}_2^n$ be a Z error with syndrome $s = H_X e$ and e' be an error with the same syndrome. Then the LP (5), defined with e', is equivalent to the LP (2). More precisely, there is a bijection between feasible solutions given by the reflections*

$$x_i = \begin{cases} x'_i, & i \notin e' \\ 1 - x'_i, & i \in e' \end{cases} \quad \forall i \in Q\,, \tag{A1}$$

$$w_{j,S} = w'_{j,S+U_j} \quad \forall j \in C_X,\ S \in E_j^{s_j}\,, \tag{A2}$$

where x_i,$w_{j,S}$ are variables of the LP (2), x'_i,$w'_{j,S}$ are variables of the LP (5), and $U_j = e'|_{f_j}$. Two solutions related by the bijection have the same objective values up to an additive constant $|e'|$. Furthermore, the LP decoders based on these solutions give the same output.

Proof. Let x', w' be a feasible solution to LP (5), and define x, w as in the lemma. Note that $|U_j| = s_j \mod 2$, so if $S \in E_j^{s_j}$, then $S + U_j \in E_j^0$ and $w'_{j,S+U_j}$ is indeed a variable of the LP (5). We verify that x, w is a feasible solution of the LP (2):

$$\sum_{S \in E_j^{s_j}} w_{j,S} = \sum_{S \in E_j^{s_j}} w'_{j,S+U_j} = \sum_{\tilde{S} \in E_j^0} w'_{j,\tilde{S}} = 1 \quad \forall j \in C_X\,, \tag{A3}$$

where we made the substitution $\tilde{S} = S + U_j$. To check the next constraint, let $i \in Q$ and $j \in C_X$ with $\{i,j\} \in E$, and consider two cases. If $i \notin e'$, then for any $S \subseteq f_j$, we have $i \in S$ if and only if $i \in S + U_j$, so

$$\sum_{\substack{S \in E_j^{s_j} \\ S \ni i}} w_{j,S} = \sum_{\substack{S \in E_j^{s_j} \\ S \ni i}} w'_{j,S+U_j} = \sum_{\substack{\tilde{S} \in E_j^0 \\ \tilde{S} \ni i}} w'_{j,\tilde{S}} = x'_i = x_i\,, \tag{A4}$$

where $\tilde{S} = S + U_j$ as before. If $i \in e'$, then for any $S \subseteq f_j$, we have $i \in S$ if and only if $i \notin S + U_j$. In this case,

$$\begin{aligned} \sum_{\substack{S \in E_j^{s_j} \\ S \ni i}} w_{j,S} &= \sum_{\substack{S \in E_j^{s_j} \\ S \ni i}} w'_{j,S+U_j} = \sum_{\substack{\tilde{S} \in E_j^0 \\ \tilde{S} \not\ni i}} w'_{j,S} \\ &= \sum_{\tilde{S} \in E_j^0} w'_{j,S} - \sum_{\substack{\tilde{S} \in E_j^0 \\ \tilde{S} \ni i}} w'_{j,S} = 1 - x'_i = x_i\,, \end{aligned} \tag{A5}$$

where we have also used the first constraint. The nonnegativity of $x_i, w_{j,S}$ follow from the fact that the constraints imply $0 \leq x'_i, w'_{j,S} \leq 1$. Showing that a feasible solution x, w of the LP (2) maps to a feasible solution x', w' of the LP (5) is similar.

The objective value of the LP (2) with solution x is

$$\sum_{i \in Q} x_i = \sum_{i \notin e} x'_i + \sum_{i \in e'} (1 - x'_i) = \sum_{i \notin e'} x'_i - \sum_{i \in e} x'_i + |e'|\,, \tag{A6}$$

which is a constant $|e'|$ greater than that of LP (5) with solution x'.

The decoder based on LP (5) outputs (a rounded version of) $\tilde{e}$ with $\tilde{e}_i = x'^*_i$ when $e'_i = 0$ and $\tilde{e}_i = 1 - x'^*_i$ when $e'_i = 1$. This is the same as the output x^* of the decoder based on LP (2). □

Appendix B: LP decoding of circuit-level noise

In this appendix, we describe how LP+OSD can handle more general noise models such as phenomenological or circuit-level noise. Since code capacity or phenomenological noise can be mapped to circuit-level noise with errors only at specific locations, we will discuss decoding a stabilizer circuit with Pauli noise.

Given a circuit, we first need to identify the *error mechanisms* affecting the circuit and define *detectors* [69]. An error mechanism v_i is a Pauli error that occurs at a specific location in the circuit with probability p_i. For example, v_i could be a two-qubit Pauli error after a two-qubit operation, or it could be a measurement error which is a bit-flip error on the classical outcome. We assume that error mechanisms occur independently, so the total distribution of errors in the circuit is a product of all the error mechanisms. In this framework, an r-qubit depolarizing channel should be decomposed as $4^r - 1$ independent error mechanisms corresponding to the nontrivial r-qubit Pauli operators. A detector D_j is a product of measurement outcomes in the circuit that is deterministic in the absence of errors but reveal information about possible errors when flipped. For syndrome extraction circuits of stabilizer codes, the detectors are often chosen to be the product of consecutive stabilizer measurements.

We then define the *detector check matrix* [57] H to be the binary matrix with rows corresponding to the different detectors and columns corresponding to the different error mechanisms such that $H_{ji} = 1$ exactly when v_i occurring causes the value of D_j to flip.

Finding the most likely error that has affected the circuit is equivalent to decoding the classical code with parity-check matrix H. Thus, we can use the LP+OSD decoder. One modification that may be needed is in the case when the error mechanisms do not all occur with the same probability, as we had assumed in the main text. In this case, we would change the coefficients of the objective function in the LP (2) so that we minimize $\sum_i w_i x_i$ instead of $\sum_i x_i$, where $w_i = \log(1 - p_i)/p_i$ is the log probability ratios of the error mechanism v_i. The OSD post-processing step is unchanged.

Note that as with stabilizers in the code capacity setting, there are combinations of error mechanisms that are harmless for circuit-level noise. For example, a qubit may suffer two consecutive Z errors, with all measurements in between that would be flipped also suffering a measurement error. However, since we are only interested finding the most likely error and our decoder does not make use of the degeneracy, these harmless configurations of errors do not need to be explicitly calculated.

Appendix C: Simulation details

We give more details on the simulations in Sec. V. We applied Monte Carlo sampling to study the performance the various decoders on the rotated surface code, random HGP codes, and BB codes. The random HGP codes are sampled as follows. We create a uniformly random connected $(3, 4)$-biregular bipartite graph G with $|A| = 4s$ left vertices and $|B| = 3s$ right vertices for a parameter s. Consider G as the Tanner graph of classical codes C with parity-check matrix $H \in \mathbb{F}_2^{B\times A}$ and $C^\top$ with parity-check matrix $H^\top \in \mathbb{F}_2^{A\times B}$. We postselect on both C and $C^\top$ having distance at least $d(s) = 2, 4, 6, 8, 8, 10$ for $s = 1, 2, 3, 4, 5, 6$, respectively (where the distance of the trivial code $\{0\}$ is defined to be infinity). The HGP code $\mathcal{C}$ is the hypergraph product of C with itself, which has parameters $[[n = 25s^2, k \geq s^2, d \geq d(s)]]$.

To estimate the logical error rate p_L, we apply Z errors independently to each qubit of the code with probability p, giving an error e. The syndrome $s = H_X e$ is passed to the decoder, which outputs a correction $\tilde{e}$. By repeating multiple times, we estimate the logical error rate p_L as the fraction of times $e+\tilde{e}$ is not a Z stabilizer. For Fig. 6, we instead determine the fraction of times $H_X\tilde{e} \neq s$, considering only the trials where LP with independent rounding fails.

For the surface code and BB codes, we took between 50,000 and 10 million samples for each value of p to obtain reasonable error bars. For the random HGP codes, each data point was the result of randomly sampling between 1000 and 60,000 codes, decoding 10 different errors for each code, and taking the total fraction over all codes and decoding trials. The error bars for this procedure were obtained via bootstrapping.

The decoders considered were LP or BP with independent rounding, OSD-0 post-pocessing, and OSD-CS post-pocessing. For these decoders, we used the same parameters as in Ref. [23]: min-sum BP with variable scaling factor $\alpha = 1 - 2^{-t}$, where t is the iteration number, and $\lambda = 60$ in OSD-CS. The BP-based decoders were implemented using the LDPC package [70]. The LPs were solved using Gurobi [30]. See Ref. [60] for the source code for the simulations and the data used to generate the plots.

[1] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. doi:10.1103/PhysRevA.52.R2493. URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[2] John Preskill. Reliable quantum computers. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):385–410, 1998. doi:10.1098/rspa.1998.0167. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0167.

[3] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge university press, Cambridge, 10th anniversary edition edition, 2010. ISBN 9781107002173.

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

[5] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information and Computation*, 14(15–16):1339–1371, November 2014. ISSN 15337146, 15337146. doi:10.26421/QIC14.15-16-5. URL http://www.rintonpress.com/journals/doi/QIC14.15-16-5.html.

[6] 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, March 2024. ISSN 1476-4687. doi:10.1038/s41586-024-07107-7. URL https://www.nature.com/articles/s41586-024-07107-7.

[7] Qian Xu, J. Pablo Bonilla Ataides, Christopher A. Pattison, Nithin Raveendran, Dolev Bluvstein, Jonathan Wurtz, Bane Vasić, Mikhail D. Lukin, Liang Jiang, and Hengyun Zhou. Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *Nature Physics*, 20(7):1084–1090, July 2024. ISSN 1745-2481. doi:10.1038/s41567-024-02479-z. URL https://www.nature.com/articles/s41567-024-02479-z.

[8] Quynh T. Nguyen and Christopher A. Pattison. Quantum fault tolerance with constant-space and logarithmic-time overheads, 2024. URL https://arxiv.org/abs/2411.03632.

[9] Hayata Yamasaki and Masato Koashi. Time-efficient

constant-space-overhead fault-tolerant quantum computation. *Nature Physics*, 20(2):247–253, February 2024. ISSN 1745-2481. doi:10.1038/s41567-023-02325-8. URL https://www.nature.com/articles/s41567-023-02325-8.

[10] 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, 2014. doi:10.1109/TIT.2013.2292061.

[11] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: breaking the n1/2 polylog(n) barrier for quantum ldpc codes. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1276–1288, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380539. doi:10.1145/3406325.3451005. URL https://doi.org/10.1145/3406325.3451005.

[12] Pavel Panteleev and Gleb Kalachev. Quantum ldpc codes with almost linear minimum distance. *IEEE Transactions on Information Theory*, 68(1):213–229, 2022. doi:10.1109/TIT.2021.3119384.

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

[14] 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*, STOC 2022, page 375–388, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. doi:10.1145/3519935.3520017. URL https://doi.org/10.1145/3519935.3520017.

[15] Anthony Leverrier and Gilles Zemor. Quantum Tanner codes . In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 872–883, Los Alamitos, CA, USA, November 2022. IEEE Computer Society. doi:10.1109/FOCS54457.2022.00117. URL https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00117.

[16] Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, and Thomas Vidick. Good quantum ldpc codes with linear time decoders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 905–918, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi:10.1145/3564246.3585101. URL https://doi.org/10.1145/3564246.3585101.

[17] 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, 2015. doi:10.1109/FOCS.2015.55.

[18] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Constant overhead quantum fault-tolerance with quantum expander codes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754, 2018. doi:10.1109/FOCS.2018.00076.

[19] Shouzhen Gu, Christopher A. Pattison, and Eugene Tang. An efficient decoder for a linear distance quantum ldpc code. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, page 919–932, Orlando FL USA, June 2023. ACM. ISBN 9781450399135. doi:10.1145/3564246.3585169. URL https://dl.acm.org/doi/10.1145/3564246.3585169.

[20] Anthony Leverrier and Gilles Zémor. Efficient decoding up to a constant fraction of the code length for asymptotically good quantum codes. *ACM Trans. Algorithms*, May 2024. ISSN 1549-6325. doi:10.1145/3663763. URL https://doi.org/10.1145/3663763. Just Accepted.

[21] Shouzhen Gu, Eugene Tang, Libor Caha, Shin Ho Choe, Zhiyang He, and Aleksander Kubica. Single-shot decoding of good quantum ldpc codes. *Communications in Mathematical Physics*, 405(3):85, March 2024. ISSN 1432-0916. doi:10.1007/s00220-024-04951-6. URL https://doi.org/10.1007/s00220-024-04951-6.

[22] Pavel Panteleev and Gleb Kalachev. Degenerate Quantum LDPC Codes With Good Finite Length Performance. *Quantum*, 5:585, November 2021. ISSN 2521-327X. doi:10.22331/q-2021-11-22-585. URL https://doi.org/10.22331/q-2021-11-22-585.

[23] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell. Decoding across the quantum low-density parity-check code landscape. *Phys. Rev. Res.*, 2:043423, Dec 2020. doi:10.1103/PhysRevResearch.2.043423. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.043423.

[24] Nicolas Delfosse, Vivien Londe, and Michael Beverland. Toward a union-find decoder for quantum ldpc codes, 2021. URL https://arxiv.org/abs/2103.08049.

[25] M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. doi:10.1109/18.556667.

[26] Judea Pearl. Reverend bayes on inference engines: a distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82, page 133–136. AAAI Press, 1982.

[27] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi:10.1109/18.910572.

[28] Jon Feldman, Martin J. Wainwright, and David R. Karger. Using linear programming to decode binary linear codes. *IEEE Trans. Inform. Theory*, 51(3):954–972, 2005. ISSN 0018-9448. doi:10.1109/TIT.2004.842696. URL https://doi.org/10.1109/TIT.2004.842696.

[29] Jon Feldman. *Decoding error-correcting codes via linear programming*. Thesis, Massachusetts Institute of Technology, 2003. URL https://dspace.mit.edu/handle/1721.1/42831.

[30] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

[31] Jon Feldman, Tal Malkin, Rocco A. Servedio, Cliff Stein, and Martin J. Wainwright. LP decoding corrects a constant fraction of errors. *IEEE Transactions on Information Theory*, 53(1):82–89, 2007. doi:10.1109/TIT.2006.887523.

[32] Jon Feldman and Cliff Stein. Lp decoding achieves capacity. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 460–469, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857.

[33] Omar Fawzi, Lucien Grouès, and Anthony Leverrier. Linear programming decoder for hypergraph product quantum codes. In *2020 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2021. doi:10.1109/ITW46852.2021.9457611.

[34] July X. Li and Pascal O. Vontobel. LP decoding of quantum stabilizer codes. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1306–1310, 2018. doi:10.1109/ISIT.2018.8437905.

[35] Lucien Grouès. *Decoding of LDPC quantum codes*. Theses, Sorbonne Université, December 2022. URL https://theses.hal.science/tel-04018968.

[36] David J. C. MacKay and Radford M. Neal. Good codes based on very sparse matrices. In Colin Boyd, editor, *Cryptography and Coding*, page 100–111, Berlin, Heidelberg, 1995. Springer. ISBN 9783540492801. doi:10.1007/3-540-60693-9˙13.

[37] David J.C. MacKay and Radford M Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32:1645–1646, 1996. doi:10.1049/el:19961141. URL https://digital-library.theiet.org/doi/abs/10.1049/el%3A19961141.

[38] D.J.C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999. doi:10.1109/18.748992.

[39] D. Poulin and Y. Chung. On the iterative decoding of sparse quantum codes. *Quantum Information and Computation*, 8(10):986–1000, November 2008. ISSN 15337146, 15337146. doi:10.26421/QIC8.10-8. URL http://www.rintonpress.com/journals/doi/QIC8.10-8.html.

[40] Zunaira Babar, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. Fifteen years of quantum ldpc coding and improved decoding strategies. *IEEE Access*, 3:2492–2519, 2015. doi:10.1109/ACCESS.2015.2503267.

[41] Nithin Raveendran and Bane Vasić. Trapping Sets of Quantum LDPC Codes. *Quantum*, 5:562, October 2021. ISSN 2521-327X. doi:10.22331/q-2021-10-14-562. URL https://doi.org/10.22331/q-2021-10-14-562.

[42] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996. doi:10.1103/PhysRevA.54.1098. URL https://link.aps.org/doi/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, Nov 1996. doi:10.1098/rspa.1996.0136. URL https://doi.org/10.1098%2Frspa.1996.0136.

[44] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003. ISSN 0003-4916. doi:10.1016/S0003-4916(02)00018-0. URL https://www.sciencedirect.com/science/article/pii/S0003491602000180.

[45] Richard M. Karp. *Reducibility among Combinatorial Problems*, page 85–103. Springer US, Boston, MA, 1972. ISBN 9781468420012. doi:10.1007/978-1-4684-2001-2˙9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

[46] Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, Chichester Weinheim, nachdr. edition, 2011. ISBN 9780471982326.

[47] M.P.C. Fossorier and Shu Lin. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Transactions on Information Theory*, 41(5):1379–1396, 1995. doi:10.1109/18.412683.

[48] M.P.C. Fossorier. Iterative reliability-based decoding of low-density parity check codes. *IEEE Journal on Selected Areas in Communications*, 19(5):908–917, 2001. doi:10.1109/49.924874.

[49] Constantinos Daskalakis, Alexandros G. Dimakis, Richard M. Karp, and Martin J. Wainwright. Probabilistic analysis of linear programming decoding. *IEEE Transactions on Information Theory*, 54(8):3565–3578, 2008. doi:10.1109/TIT.2008.926452.

[50] Michele Pacenti and Bane Vasic. Quantum margulis codes, 2024. URL https://arxiv.org/abs/2409.09830.

[51] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 2025. ISBN 9781611978322. doi:10.1137/1.9781611978322. URL https://epubs.siam.org/doi/book/10.1137/1.9781611978322.

[52] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 938–942, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi:10.1145/3313276.3316303. URL https://doi.org/10.1145/3313276.3316303.

[53] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. In *25th Annual Symposium onFoundations of Computer Science, 1984.*, pages 338–346, 1984. doi:10.1109/SFCS.1984.715934.

[54] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001. ISBN 0262032937.

[55] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.

[56] Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. Towards practical classical processing for the surface code. *Phys. Rev. Lett.*, 108:180501, May 2012. doi:10.1103/PhysRevLett.108.180501. URL https://link.aps.org/doi/10.1103/PhysRevLett.108.180501.

[57] Oscar Higgott and Craig Gidney. Sparse Blossom: correcting a million errors per core second with minimum-weight matching. *Quantum*, 9:1600, January 2025. ISSN 2521-327X. doi:10.22331/q-2025-01-20-1600. URL https://doi.org/10.22331/q-2025-01-20-1600.

[58] Yue Wu and Lin Zhong. Fusion blossom: Fast mwpm decoders for qec. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 928–938, 2023. doi:10.1109/QCE57702.2023.00107.

[59] Yue Wu. Hypergraph minimum-weight parity factor decoder for qec, 2025. URL https://github.com/yuewuo/mwpf.

[60] Shouzhen Gu and Mehdi Soleimanifar. Data for "power and limitations of linear programming decoder for quantum ldpc codes", 2025. URL https://doi.org/10.5281/zenodo.16748401.

[61] David Burshtein and Idan Goldenberg. Improved linear programming decoding and bounds on the minimum distance of ldpc codes. In *2010 IEEE Information Theory Workshop*, pages 1–5, 2010. doi:10.1109/CIG.2010.5592887.

[62] Mohammad H. Taghavi N. and Paul H. Siegel. Adaptive methods for linear programming decoding. *IEEE Transactions on Information Theory*, 54(12):5396–5410, 2008. doi:10.1109/TIT.2008.2006384.

[63] Alexandros G. Dimakis and Martin J. Wainwright. Guessing facets: Polytope structure and improved lp decoder. In *2006 IEEE International Symposium on Information Theory*, pages 1369–1373, 2006. doi:10.1109/ISIT.2006.262070.

[64] Kai Yang, J. Feldman, and Xiaodong Wang. Nonlinear programming approaches to decoding low-density parity-check codes. *IEEE Journal on Selected Areas in Communications*, 24(8):1603–1613, 2006. doi:10.1109/JSAC.2006.879405.

[65] Siddharth Barman, Xishuo Liu, Stark Draper, and Benjamin Recht. Decomposition methods for large scale lp decoding. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 253–260, 2011. doi:10.1109/Allerton.2011.6120176.

[66] Yige Wang, Jonathan S. Yedidia, and Stark C. Draper. Multi-stage decoding of ldpc codes. In *2009 IEEE International Symposium on Information Theory*, pages 2151–2155, 2009. doi:10.1109/ISIT.2009.5205786.

[67] Antonio deMarti iOlius, Imanol Etxezarreta Martinez, Joschka Roffe, and Josu Etxezarreta Martinez. An almost-linear time decoding algorithm for quantum ldpc codes under circuit-level noise, 2024. URL https://arxiv.org/abs/2409.01440.

[68] Timo Hillmann, Lucas Berent, Armanda O. Quintavalle, Jens Eisert, Robert Wille, and Joschka Roffe. Localized statistics decoding: A parallel decoding algorithm for quantum low-density parity-check codes, 2024. URL https://arxiv.org/abs/2406.18655.

[69] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. ISSN 2521-327X. doi:10.22331/q-2021-07-06-497. URL https://doi.org/10.22331/q-2021-07-06-497.

[70] Joschka Roffe. LDPC: Python tools for low density parity check codes, 2022. URL https://pypi.org/project/ldpc/.