

Breaking the Orthogonality Barrier in Quantum LDPC Codes

Kenta Kasai
 Institute of Science Tokyo
 kenta@ict.eng.isct.ac.jp

Abstract

Classical low-density parity-check (LDPC) codes are a widely deployed and well-established technology, forming the backbone of modern communication and storage systems. It is well known that, in this classical setting, increasing the girth of the Tanner graph while maintaining regular degree distributions leads simultaneously to good belief-propagation (BP) decoding performance and large minimum distance. In the quantum setting, however, this principle does not directly apply because quantum LDPC codes must satisfy additional orthogonality constraints between their parity-check matrices. When one enforces both orthogonality and regularity in a straightforward manner, the girth is typically reduced and the minimum distance becomes structurally upper bounded. In this work, we overcome this limitation by using permutation matrices with controlled commutativity and by restricting the orthogonality constraints to only the active part of the construction, while preserving regular check-matrix structures. This design circumvents conventional structural distance limitations induced by parent-matrix orthogonality, and enables the construction of quantum LDPC codes with large girth while avoiding latent low-weight logical operators. As a concrete demonstration, we construct a girth-8, (3,12)-regular $[[9216, 4612, \leq 48]]$ quantum LDPC code and show that, under BP decoding combined with a low-complexity post-processing algorithm, it achieves a frame error rate as low as 10^{-8} on the depolarizing channel with error probability 4%.

1 Introduction

Classical low-density parity-check (LDPC) codes are a widely deployed and well-established technology in modern communication and storage systems [1], and BP decoding can achieve channel capacity [2, 3]. Performance depends strongly on the Tanner-graph degree distribution: regular codes can perform well, and carefully designed irregular distributions can further improve BP performance [4], whereas naive regularization can degrade it. Random LDPC codes with variable-node degree at least 3 have minimum distance growing linearly with blocklength [5, 1], so classical LDPC research has not focused on increasing minimum distance itself. Moreover, BP decoding can be trapped by small Tanner-graph structures called trapping sets [6], which cause decoding failures; suppressing harmful trapping sets typically requires large girth [7].

Sparse-graph quantum error-correcting codes were proposed in [8], and the behavior of iterative decoding and the role of degeneracy were discussed in [9]. Families with positive rate and distance $\Theta(\sqrt{n})$ are known [10]. Constructions based on Kronecker sum and product [11], distance bounds for generalized bicycle (GB) codes [12], finite-length evaluations [13], and decoding protocols [14] have been reported, advancing the theoretical understanding of CSS-type QLDPC codes. However, the orthogonality constraint between the CSS parity-check matrices H_X and H_Z , namely $H_X H_Z^\top = 0$, has no classical counterpart, and enforcing both orthogonality and regularity in a straightforward manner typically reduces girth and induces structural distance upper bounds.

Surface codes and hypergraph product (HGP) codes rely on geometric or product structures and thus follow design principles different from those developed for classical LDPC codes. As a result, it is not straightforward to apply classical degree-distribution and girth design techniques directly [15, 10, 5]. The goal of this work is to provide a construction principle that preserves classical LDPC parity-check structures while enabling their direct use as CSS check matrices. Specifically, we aim for regular LDPC codes with large girth and variable degree at least 3, whose minimum distance is not trivially upper bounded.

Prior work has explored cyclic parity-check matrices that allow explicit control of regular Tanner-graph degree distributions. Representative constructions include the Hagiwara–Imai codes [16] and bicycle constructions [8, 11, 12]. However, these constructions have known limitations. Even for classical codes, one-step lifting of a protograph (lifting by circulant permutation matrices (CPMs)) can impose fixed upper bounds on girth due to the protograph structure. Mitchell et al. pointed out that one-step lifting can impose fixed bounds on minimum distance and girth and that two-step lifting can improve them [17]. For CSS codes, the parity-check matrices H_X and H_Z are QC, and the Tanner-graph girth satisfies $g = \min\{g(H_X), g(H_Z)\}$, so QC girth bounds apply to each matrix. For quasi-cyclic CSS designs, generalized CSS (entanglement-assisted) constructions use classical QC constructions with girth ≥ 6 to avoid 4-cycles [18]. Explicit constructions achieving girth 12 for column weight 2 have also been reported [19]. Moreover, for general quantum CPM–LDPC codes, the girth is upper bounded by 12 for column weight 2 and by 6 for column weight at least 3 [20]. Using APM–LDPC codes, these upper bounds can be broken [19, 21].

In CSS-type LDPC constructions, deleting rows from the parity-check matrices H_X or H_Z preserves the commutation condition $H_X H_Z^\top = 0$, but it weakens stabilizer constraints and can enlarge the code space [8, 16]. Row deletion is used for rate adjustment and for controlling row and column weights, yet distance preservation is not guaranteed in general [16, 22]. In particular, deleting rows can allow low-weight vectors to enter $C_X \setminus C_Z^\perp$ or $C_Z \setminus C_X^\perp$, creating new low-weight logical operators and thus reducing the minimum distance. In CSS codes, with $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$, low-weight vectors in $C_X \setminus C_Z^\perp$ or $C_Z \setminus C_X^\perp$ become non-trivial logical operators [23, 24, 25]. Consequently, deleted check rows (which are LDPC and thus low weight) can become logical operators and upper bound the distance by the row weight. In the CSS product-code constructions of Ostrev et al., row removal is explicitly used to control the number of stabilizers, and distance degradation can occur [22]. Lin et al. report that removing redundant stabilizer rows reduces the syndrome distance, which is a structural effect due to fewer constraints, not a decoder artifact [14].

In this work, we first discuss the general mechanism by which row deletion can degrade the minimum distance of CSS codes. Next, for generalized Hagiwara–Imai codes [19], we establish a code-design method that controls commutativity of submatrices to avoid distance degradation due to row deletion. More concretely, using the algebraic structure of APM–LDPC codes, we guarantee orthogonality only on the active part, introduce a set of row pairs for which commutation is required, and localize the commutation condition. We further control the nonzero pattern of the interaction matrix so that deleted rows do not materialize as low-weight logical operators, and present a sequential construction algorithm based on the APM composition rule. We also give a girth upper bound under the required commutativity and explicitly construct a code that attains this bound.

As a concrete demonstration, we construct a (3,12)-regular $[[9216, 4612, \leq 48]]$ quantum LDPC code and show that, under BP decoding combined with a low-complexity post-processing algorithm, it achieves a frame error rate (FER) as low as 10^{-8} on the depolarizing channel with error probability 4%. We also compare constructions with and without commutation control to demonstrate suppression of low-weight logical operators.

2 Problem setting: low-weight logical operators induced by check-row removal

We abstract the conventional CSS–LDPC construction used in this paper, based on [8, 16]. We first construct two orthogonal square (block-circulant) parent matrices \hat{H}_X, \hat{H}_Z . The CSS code defined by these parent matrices typically has rate zero, so to adjust the rate we delete internal rows and use the remaining rows as the active matrices H_X and H_Z . Namely,

$$\hat{H}_X = \begin{bmatrix} H_X \\ \tilde{H}_X \end{bmatrix}, \quad \hat{H}_Z = \begin{bmatrix} H_Z \\ \tilde{H}_Z \end{bmatrix}$$

We refer to H_X and H_Z as the active part and \tilde{H}_X and \tilde{H}_Z as the latent part.

However, in this construction, parent orthogonality imposes strong constraints on the latent rows and can degrade the minimum distance. The minimum distance is defined as $d_{\min} = \min\{d_X, d_Z\}$ with

$$d_Z := \min\{\text{wt}(\mathbf{z}) : \mathbf{z} \in C_X \setminus C_Z^\perp\}, \quad d_X := \min\{\text{wt}(\mathbf{x}) : \mathbf{x} \in C_Z \setminus C_X^\perp\}.$$

Parent orthogonality

$$\hat{H}_X(\hat{H}_Z)^\top = 0$$

forces not only the active orthogonality between the active matrices H_X and H_Z , namely $H_X H_Z^\top = 0$, but also

$$H_X(\tilde{H}_Z)^\top = 0, \quad H_Z(\tilde{H}_X)^\top = 0$$

which implies $\text{Row}(\tilde{H}_X) \subset C_Z$ and $\text{Row}(\tilde{H}_Z) \subset C_X$. Thus each row \mathbf{x} of \tilde{H}_X satisfies $\mathbf{x} \in C_Z$, and each row \mathbf{z} of \tilde{H}_Z satisfies $\mathbf{z} \in C_X$. In general \mathbf{x} need not belong to C_X^\perp and \mathbf{z} need not belong to C_Z^\perp , in which case they become logical operators. The minimum row weight (in our construction, L) becomes an upper bound on d_X and d_Z .

In this work we also obtain the active matrices H_X, H_Z by deleting rows from the parent matrices, but we design them so that low-weight rows in the latent matrices are not orthogonal to H_Z or H_X , respectively. That is, for low-weight $\mathbf{x} \in \text{Row}(\tilde{H}_X)$ we enforce $H_Z \mathbf{x}^\top \neq 0$, and for low-weight $\mathbf{z} \in \text{Row}(\tilde{H}_Z)$ we enforce $H_X \mathbf{z}^\top \neq 0$, so that in particular $\text{Row}(\tilde{H}_X) \not\subset C_Z$ and $\text{Row}(\tilde{H}_Z) \not\subset C_X$. The same argument applies not only to individual latent rows but also to low-weight linear combinations.

Accordingly, we define latent-based distance for the X and Z distances as

$$\begin{aligned} d_X^{(\text{lat})} &:= \min\{\text{wt}(\mathbf{x}) : \mathbf{x} \in (C_Z \cap \text{Row}(\tilde{H}_X)) \setminus C_X^\perp\}, \\ d_Z^{(\text{lat})} &:= \min\{\text{wt}(\mathbf{z}) : \mathbf{z} \in (C_X \cap \text{Row}(\tilde{H}_Z)) \setminus C_Z^\perp\}. \end{aligned}$$

These quantities upper bound d_{\min} . The proposed design aims to make these bounds as large as possible. In [26], binary submatrices corresponding to codewords are lifted to a nonbinary field so that they no longer represent codewords; this can be viewed as increasing these latent-based distance via nonbinary lifting.

3 Method of matrix design

In this section, based on the above problem setting, we aim to construct the latent matrices \tilde{H}_X and \tilde{H}_Z such that

$$H_X H_Z^\top = 0, \quad H_X(\tilde{H}_Z)^\top \neq 0, \quad H_Z(\tilde{H}_X)^\top \neq 0,$$

and, more precisely, such that $d_X^{(\text{lat})}$ and $d_Z^{(\text{lat})}$ are large. We define generalized Hagiwara–Imai codes, i.e., parent matrix pairs with block-circulant structure, and develop a general theory for imposing only the active orthogonality $H_X H_Z^\top = 0$ between the active matrices H_X and H_Z .

A protograph LDPC code with column weight J and row weight L is defined by a parity-check matrix consisting of $J \times L$ permutation matrices of size P [27]. For a permutation $f : [P] \rightarrow [P]$, the corresponding $P \times P$ permutation matrix $F = P(f)$ is defined by $F_{x,y} = 1$ if and only if $f(x) = y$. Each row and column has exactly one 1, so the parent matrix is sparse while its block structure is compact.

We introduce generalized Hagiwara–Imai codes [19] as protograph generalizations of the Hagiwara–Imai quasi-cyclic CSS codes [16]. For permutation matrices F_i, G_i ($i \in [L/2]$) of size P , define the parent matrices \hat{H}_X, \hat{H}_Z of size $(LP/2) \times (LP)$ by

$$(\hat{H}_X)_{i,j} = F_{j-i}, \quad (\hat{H}_X)_{i,L/2+j} = G_{j-i},$$

$$(\hat{H}_Z)_{i,j} = (G_{i-j})^\top, \quad (\hat{H}_Z)_{i,L/2+j} = (F_{i-j})^\top.$$

Throughout, indices in $[L/2]$ are taken modulo $L/2$, i.e., we identify $[L/2]$ with $\mathbb{Z}_{L/2}$. A sufficient condition for orthogonality is given below. The active matrices H_X and H_Z consist of the top J block rows. Conventional designs [19, 26] impose commutativity of F_i and G_j so that both H_X, H_Z and \hat{H}_X, \hat{H}_Z are orthogonal. The top J block rows of the parent matrices are used as the active matrices H_X and H_Z . Our interest is to satisfy active orthogonality $H_X H_Z^\top = 0$ while not imposing parent-matrix orthogonality.

This focus on active orthogonality is motivated by the desire to keep the latent part non-orthogonal to the active part, so that low-weight latent combinations do not automatically become logical operators. The block-circulant structure makes $\hat{H}_X(\hat{H}_Z)^\top$ depend only on differences, and the resulting interaction matrices Ψ_r allow us to localize commutativity constraints to a small difference set.

Each block row of \hat{H}_X is a cyclic shift of $(F_0, F_1, \dots, F_{L/2-1})$ in the left half and $(G_0, G_1, \dots, G_{L/2-1})$ in the right half. Hence blocks depend only on the difference $(j - i)$, and the product of parent matrices depends only on the difference. For any $i, k \in [L/2]$, the (i, k) block is

$$(\hat{H}_X(\hat{H}_Z)^\top)_{i,k} = \sum_{u=0}^{L/2-1} (F_u G_{r-u} + G_{r-u} F_u) =: \Psi_r$$

which depends only on $r := (k - i) \bmod L/2$. For $r \in [L/2]$, if F_u and G_{r-u} commute for all $u \in [L/2]$, then $\Psi_r = 0$.

3.1 Sufficient condition for active orthogonality

We re-derive a convenient form of a sufficient condition for $H_X H_Z^\top = 0$ [26]. Define

$$\Delta := \{(k - i) \bmod L/2 \mid 0 \leq i, k \leq J - 1\} \subseteq [L/2].$$

The next theorem spells out how commutativity restricted to these differences guarantees active orthogonality.

Theorem 3.1. If F_u and G_{r-u} commute for all $r \in \Delta$ and all $u \in [L/2]$, then $H_X H_Z^\top = 0$.

Proof. For any $i, k \in [J]$, we have $r = (k - i) \bmod L/2 \in \Delta$. By assumption, $\Psi_r = 0$, hence for all $i, k \in [J]$, $(\hat{H}_X(\hat{H}_Z)^\top)_{i,k} = \Psi_{(k-i) \bmod L/2} = 0$. Therefore $H_X H_Z^\top = 0$. \square

3.2 Necessary condition for latent non-orthogonality

Define the index pairs of (F_i, G_j) required to commute by

$$\Gamma := \bigcup_{r \in \Delta} \Gamma_r, \quad \Gamma_r := \{(i, j) \mid (i, j) = (u, r - u), u \in [L/2]\}$$

where subscripts are modulo $L/2$. Here Γ_r is the set of index pairs contributing to Ψ_r , and $\Gamma_r \cap \Gamma_s = \emptyset$ for $r \neq s$.

If $J \geq L/2$ then the rate is zero, so we assume $J < L/2$. We first state a basic feasibility constraint: without enough blocks, latent non-orthogonality cannot be forced.

Theorem 3.2. Assume the active orthogonality $H_X H_Z^\top = 0$. For the standard active choice (top J block rows), to have $H_X(\tilde{H}_Z)^\top \neq 0$ and $H_Z(\tilde{H}_X)^\top \neq 0$, it is necessary that $L \geq 4J$.

Proof. Assume $L < 4J$. Then for this active choice, $\Delta = [L/2]$. Active orthogonality $H_X H_Z^\top = 0$ implies $\Psi_r = 0$ for all $r \in \Delta$, hence $\Psi_r = 0$ for all $r \in [L/2]$. Therefore every block of $\hat{H}_X(\hat{H}_Z)^\top$ vanishes, in particular the mixed blocks between active and latent rows, so $H_X(\tilde{H}_Z)^\top = 0$ and $H_Z(\tilde{H}_X)^\top = 0$, contradicting the premise. \square

Intuitively, when $L < 4J$ the difference set Δ covers all residues, so enforcing active orthogonality forces $\Psi_r = 0$ for every r . This annihilates all mixed active-latent blocks and makes latent rows orthogonal, which is exactly what we aim to avoid.

We next isolate the minimal algebraic obstruction to latent non-orthogonality.

Theorem 3.3. To have $H_X(\tilde{H}_Z)^\top \neq 0$ and $H_Z(\tilde{H}_X)^\top \neq 0$, it is necessary that there exists $r \in [L/2] \setminus \Delta$ with $\Psi_r \neq 0$.

Proof. Assume $\Psi_r = 0$ for all $r \in [L/2] \setminus \Delta$. For any $i \in [L/2] \setminus [J]$ and $k \in [J]$, $r = (k - i) \bmod L/2$ does not belong to Δ . By assumption $\Psi_r = 0$, hence for all such i, k , $(\hat{H}_X(\hat{H}_Z)^\top)_{i,k} = \Psi_{(k-i) \bmod L/2} = 0$. Therefore $H_X(\tilde{H}_Z)^\top = 0$, and similarly $H_Z(\tilde{H}_X)^\top = 0$, a contradiction. If all pairs $(i, j) \in [L/2]^2 \setminus \Delta$ commute, then each term $F_u G_{r-u} + G_{r-u} F_u$ vanishes over \mathbb{F}_2 , so $\Psi_r = 0$ for all $r \notin \Delta$. \square

3.3 Upper bound on girth

Finally, we recall a structural limitation that applies to the active matrices when commutativity holds on all required differences.

Theorem 3.4. Let L be even and assume $3 \leq J \leq L/2$ with the standard active choice (top J block rows). If F_u and G_{r-u} commute for all $r \in \Delta$ and all $u \in [L/2]$, then the Tanner graphs of the active matrices H_X and H_Z each contain an 8-cycle.

Proof. Since $J \geq 3$, the active block rows include 0, 1, 2. Choose $i, j \in [L/2]$ so that $i + j, i + j - 1, i + j - 2 \in \Delta$ (e.g., $i = j = 0$). Consider the block positions on the active rows

$$[(0, i), (0, L/2 + j), (1, L/2 + j), (1, i + 1), (2, i + 1), (2, L/2 + j + 1), (1, L/2 + j + 1), (1, i)],$$

where indices are modulo $L/2$. The corresponding cycle product, i.e., the product of permutation matrices encountered along this closed walk, is

$$W = F_i G_j^{-1} G_{j-1} F_i^{-1} F_{i-1} G_{j-1}^{-1} G_j F_{i-1}^{-1} \tag{1}$$

and the pairs (i, j) , $(i, j - 1)$, $(i - 1, j)$, and $(i - 1, j - 1)$ correspond to the differences $i + j$, $i + j - 1$, and $i + j - 2$ in Δ . By assumption the relevant F and G commute, so $W = I$ and an 8-cycle exists in the active Tanner graphs. \square

4 Method of Matrix Construction

Building on the commutation conditions and active orthogonality derived in the previous section, we now present a concrete sequential procedure to construct permutation blocks that satisfy those constraints while avoiding short cycles. For a given $(L/2, P, J)$, we construct $\{F_i\}$ and $\{G_i\}$ that satisfy the following simultaneously:

- (1) F_i and G_j commute for $(i, j) \in \Gamma$.
- (2) At least one $(i, j) \in [L/2]^2 \setminus \Gamma$ is non-commuting.
- (3) Avoid short cycles in the active matrices H_X and H_Z .

The framework of constructing quasi-cyclic LDPC blocks from circulant permutation matrices (CPMs) was systematized by Fossorier [28]. We adopt affine permutation matrices (APMs): the APM-LDPC framework extends CPMs, and its algebraic form makes commutativity control straightforward via congruence conditions [29]. In the classical setting, Yoshida and Kasai reported that linear permutation polynomial (APM-based) codes achieve performance comparable to protograph LDPC codes [30], while no fixed girth upper bound has been reported for APM-based constructions, unlike one-step CPM lifting. Methods that combine APM-LDPC codes to extend length and girth have also been proposed [31]. From a group-theoretic viewpoint, APMs on \mathbb{Z}_P form the affine group $\text{AGL}(1, \mathbb{Z}_P) = \mathbb{Z}_P \rtimes \mathbb{Z}_P^\times$, a semidirect product of translations by the unit group; commutativity is therefore governed by the interaction between the linear and translation parts.

Consider an affine permutation (AP) on \mathbb{Z}_P ,

$$f(x) = ax + b, \quad a \in \mathbb{Z}_P^\times, \quad b \in \mathbb{Z}_P.$$

We set $f_i(x) = a_i x + b_i$ and $g_j(x) = c_j x + d_j$, and denote the corresponding permutation matrices by $F_i := P(f_i)$ and $G_j := P(g_j)$.

Under this representation, commutativity of APs can be checked by a quadratic congruence:

$$f_i g_j = g_j f_i \iff d_j(a_i - 1) - b_i(c_j - 1) \equiv 0 \pmod{P},$$

See [29] for a derivation. The condition involves products of a_i, c_j and b_i, d_j [29]. The commutation table can thus be expressed as a system of congruences; in particular, if a_i, c_j are chosen first, the commutation constraints reduce to linear congruences in b_i, d_j , enabling a consistent search [29, 31].

We sequentially select candidates that satisfy both the commutation table and short-cycle conditions, using backtracking when a candidate fails. The number of trials is adjusted dynamically based on recent success rates. We interpret each candidate generator as an arm in a multi-armed bandit and allocate trials accordingly [32].

Short-cycle detection reduces to checking fixed points of the composite map Σ along a block-cycle pattern [28, 29]. The AP composition is again AP, $\Sigma(x) = Ax + B$, so

$$\Sigma(x) = x \iff \gcd(A - 1, P) \mid B$$

provides a test without enumerating the Tanner graph [29].

5 Method of Decoding

Our experiments use BP decoding [8, 9] as the baseline. We first recall a basic principle: if we knew the support where the BP estimate differs from the true error, decoding would reduce to a linear

system on the corresponding columns, and a unique solution would imply successful recovery. We write $H[A, B]$ for the submatrix with row set A and column set B , and for a vector \mathbf{v} we write $\mathbf{v}[A]$ for its restriction to A . For example, if the X -side mismatch support is E , then with A the full check set, $H_Z[A, E] \mathbf{e}[E] = \mathbf{r}_X[A]$; if $H_Z[A, E]$ has full column rank, the solution is unique and yields the correct correction. The same applies to the Z side with H_X and \mathbf{r}_Z .

Let the BP estimate at each iteration be $\hat{\mathbf{x}}, \hat{\mathbf{z}}$. Define the residual syndromes

$$\mathbf{r}_X := \mathbf{s}_X \oplus H_Z \hat{\mathbf{x}}, \quad \mathbf{r}_Z := \mathbf{s}_Z \oplus H_X \hat{\mathbf{z}}.$$

Empirically, BP stalls or slowly oscillates roughly once per 10^5 trials (frames), and we trigger post-processing in such cases, focusing on instances with a small number of unsatisfied checks; in our experiments we activate it only when the number of unsatisfied checks is at most 20.

We invoke three post-processing methods: ETS-based post-processing, a correction based on flip history, and an OSD-based method. Below we briefly describe the procedure and the criteria. In post-processing, the residual syndrome \mathbf{r} is \mathbf{r}_X (with $H = H_Z$) or \mathbf{r}_Z (with $H = H_X$), depending on the side.

Because the true mismatch support E is not directly observable by the decoder, our approach is to find a candidate support $\hat{E} \supseteq E$ and solve the local system on its neighborhood; when the system defined by $H[N(\hat{E}), \hat{E}]$ and $\mathbf{r}[N(\hat{E})]$ has a unique solution $\delta[\hat{E}]$, we adopt it as the mismatch correction. The correction is applied only when the solution is unique and its weight is below a threshold. Below we summarize the heuristics used to estimate \hat{E} .

5.1 Flip history decoding

Flip history decoding (FHD) uses the flip history to form a candidate support $\hat{E} = F$, where F is the set of variables that flipped during BP, and then applies the local linear solve on $N(F)$. This local correction is a heuristic to resolve stalls caused by trapping structures [6]. Related local-flip decoders include weighted bit-flipping [33].

5.2 Ordered statistics decoding

Ordered statistics decoding (OSD) [34] uses the least reliable variables from BP to form a candidate support $\hat{E} = K$. In the implementation, we perform a binary search on $|K|$ to find the minimum K that is solvable, then take the largest K within the range that preserves uniqueness.

5.3 Trapping set based post-processing

An (a, b) elementary trapping set is a variable-node set V with $|V| = a$ whose induced subgraph has exactly b unsatisfied (odd-degree) check nodes; it is elementary if every check node in the induced subgraph has degree 1 or 2. The post-processing targets elementary trapping sets with $b = 2$ unsatisfied checks, which are among the most harmful structures that cause BP to stall. We precompute a library of harmful trapping sets from the Tanner graph and store each entry as a variable set V (used as \hat{E}) and its odd check pair (c_0, c_1) . During decoding, we run the post-processing only when the residual syndrome has exactly two unsatisfied checks.

6 Results

We instantiate the general theory of Section 3 and the sequential construction of Section 4 for the smallest case we were able to construct, $J = 3, L = 12, P = 768$. We first give explicit Δ, Γ and

Ψ_r , clarify the active orthogonality condition for the active matrices H_X and H_Z , then design the commutation table and APM parameters, and finally evaluate distance bounds and FER.

6.1 Structure and active orthogonality

For $J = 3, L/2 = 6$, each block row is a cyclic shift of the previous row, so \hat{H}_X, \hat{H}_Z have a 6×12 block-circulant structure:

$$\hat{H}_X = \left(\begin{array}{cccccc|cccccc} F_0 & F_1 & F_2 & F_3 & F_4 & F_5 & G_0 & G_1 & G_2 & G_3 & G_4 & G_5 \\ F_5 & F_0 & F_1 & F_2 & F_3 & F_4 & G_5 & G_0 & G_1 & G_2 & G_3 & G_4 \\ F_4 & F_5 & F_0 & F_1 & F_2 & F_3 & G_4 & G_5 & G_0 & G_1 & G_2 & G_3 \\ \hline F_3 & F_4 & F_5 & F_0 & F_1 & F_2 & G_3 & G_4 & G_5 & G_0 & G_1 & G_2 \\ F_2 & F_3 & F_4 & F_5 & F_0 & F_1 & G_2 & G_3 & G_4 & G_5 & G_0 & G_1 \\ F_1 & F_2 & F_3 & F_4 & F_5 & F_0 & G_1 & G_2 & G_3 & G_4 & G_5 & G_0 \end{array} \right).$$

$$\hat{H}_Z = \left(\begin{array}{cccccc|cccccc} G'_0 & G'_5 & G'_4 & G'_3 & G'_2 & G'_1 & F'_0 & F'_5 & F'_4 & F'_3 & F'_2 & F'_1 \\ G'_1 & G'_0 & G'_5 & G'_4 & G'_3 & G'_2 & F'_1 & F'_0 & F'_5 & F'_4 & F'_3 & F'_2 \\ \hline G'_2 & G'_1 & G'_0 & G'_5 & G'_4 & G'_3 & F'_2 & F'_1 & F'_0 & F'_5 & F'_4 & F'_3 \\ G'_3 & G'_2 & G'_1 & G'_0 & G'_5 & G'_4 & F'_3 & F'_2 & F'_1 & F'_0 & F'_5 & F'_4 \\ G'_4 & G'_3 & G'_2 & G'_1 & G'_0 & G'_5 & F'_4 & F'_3 & F'_2 & F'_1 & F'_0 & F'_5 \\ G'_5 & G'_4 & G'_3 & G'_2 & G'_1 & G'_0 & F'_5 & F'_4 & F'_3 & F'_2 & F'_1 & F'_0 \end{array} \right).$$

Here, for simplicity, A' denotes the transpose of A . For example, the $(0, 1)$ block of $\hat{H}_X(\hat{H}_Z)^\top$ is

$$\begin{aligned} (\hat{H}_X(\hat{H}_Z)^\top)_{0,1} &= F_0G_1 + F_1G_0 + F_2G_5 + F_3G_4 + F_4G_3 + F_5G_2 \\ &\quad + G_0F_1 + G_1F_0 + G_2F_5 + G_3F_4 + G_4F_3 + G_5F_2 \\ &= \Psi_1. \end{aligned}$$

With $\Delta = \{0, 1, 2, 4, 5\}$,

$$\Gamma = \{(i, j) \in [L/2]^2 \mid i + j \not\equiv 3 \pmod{6}\} = [L/2]^2 \setminus \{(0, 3), (1, 2), (2, 1), (3, 0), (4, 5), (5, 4)\}.$$

For the active matrices H_X and H_Z with $J = 3$,

$$H_X H_Z^\top = (\Psi_{(k-i) \bmod 6})_{0 \leq i, k \leq 2} = \begin{pmatrix} \Psi_0 & \Psi_1 & \Psi_2 \\ \Psi_5 & \Psi_0 & \Psi_1 \\ \Psi_4 & \Psi_5 & \Psi_0 \end{pmatrix}.$$

Here we set $r := (k - i) \bmod 6$. For the active part we have $i, k \in \{0, 1, 2\}$, hence $r \in \Delta = \{0, 1, 2, 4, 5\}$. Thus active orthogonality is equivalent to

$$\Psi_r = 0 \quad (r \in \{0, 1, 2, 4, 5\} = \Delta),$$

so the $r = 3$ interaction can be kept free. The parent-matrix product $\hat{H}_X(\hat{H}_Z)^\top$ is

$$\hat{H}_X(\hat{H}_Z)^\top = (\Psi_{(k-i) \bmod 6})_{0 \leq i, k \leq 5} = \left(\begin{array}{ccc|ccc} \Psi_0 & \Psi_1 & \Psi_2 & \Psi_3 & \Psi_4 & \Psi_5 \\ \Psi_5 & \Psi_0 & \Psi_1 & \Psi_2 & \Psi_3 & \Psi_4 \\ \Psi_4 & \Psi_5 & \Psi_0 & \Psi_1 & \Psi_2 & \Psi_3 \\ \hline \Psi_3 & \Psi_4 & \Psi_5 & \Psi_0 & \Psi_1 & \Psi_2 \\ \Psi_2 & \Psi_3 & \Psi_4 & \Psi_5 & \Psi_0 & \Psi_1 \\ \Psi_1 & \Psi_2 & \Psi_3 & \Psi_4 & \Psi_5 & \Psi_0 \end{array} \right).$$

6.2 Design goal and commutation table

With $L/2 = 6$ and $J = 3$, we have $\Delta = \{0, 1, 2, 4, 5\}$, so that $3 \notin \Delta$. Our design guideline is to enlarge $d^{(\text{lat})}$ as a proxy for increasing d_{\min} . In this setting, we target

$$\Psi_r = 0 \quad (r \in \Delta), \quad \Psi_3 \neq 0.$$

This condition guarantees active orthogonality while deliberately relaxing parent orthogonality, thereby creating room to prevent the latent part from becoming orthogonal to the active part.

As an illustrative commutation table, suppose that all pairs (F_i, G_j) commute except $(i, j) = (0, 3)$ and $(1, 2)$. The table below indicates the commuting pairs in Γ by 1:

	G_0	G_1	G_2	G_3	G_4	G_5
F_0	1	1	1	0	1	1
F_1	1	1	0	1	1	1
F_2	1	1	1	1	1	1
F_3	1	1	1	1	1	1
F_4	1	1	1	1	1	1
F_5	1	1	1	1	1	1

Under this commutation pattern, we obtain

$$\Psi_r = 0 \quad (r \neq 3), \quad \Psi_3 = F_0G_3 + G_3F_0 + F_1G_2 + G_2F_1.$$

At first glance, one might expect that it is sufficient to keep Ψ_3 nonzero by making only F_0 and G_3 non-commuting. Indeed, we initially considered this minimal design and constructed codes satisfying $\Psi_3 \neq 0$. However, this choice resulted in many ($a, b = 2$) elementary trapping sets formed by connecting multiple length-8 cycles, which are among the most harmful structures for BP decoding [6]. Moreover, two trapping sets sharing these same check-node pairs gave rise to low-weight logical operators. Representative examples are shown in Fig. 1. In contrast, by allowing both (F_0, G_3) and (F_1, G_2) to be non-commuting, we introduce additional degrees of freedom in Ψ_3 , which substantially reduces the number of harmful trapping sets while still satisfying the active orthogonality constraint. In example codes constructed this way, the only small trapping sets of weight a are those listed in Section 6.5, and none of them produced low-weight logical operators.

Remark. Throughout Section 3 we considered the standard active choice consisting of the top J block rows. More generally, let $S \subseteq \mathbb{Z}_{L/2}$ be an arbitrary active index set with $|S| = J$, and define

$$\Delta_S := \{(k - i) \bmod (L/2) \mid i, k \in S\}.$$

The proof of Theorem 3.1 applies verbatim with Δ replaced by Δ_S .

This generalization allows a substantial reduction of $|\Delta|$. In particular, for $J = 3$ and $L = 12$, choosing $S = \{0, 2, 4\}$ gives $|\Delta_S| = 3$, whereas the standard choice yields $|\Delta| = 5$. Such a reduction increases the number of indices $r \notin \Delta_S$ for which Ψ_r can be kept nonzero, and may be exploited to further control trapping-set structures. Under this relaxed constraint, constructing codes may allow us to achieve the girth and minimum distance studied here at shorter codelengths.

6.3 Affine permutation construction and choice of P

We construct a girth-8 (3, 12)-regular code; the parameters f_i, g_i are listed in Table 1 (indices start at 0). Each f_i, g_i is an affine permutation $x \mapsto ax + b$ on \mathbb{Z}_P with $\gcd(a, P) = 1$. The resulting

Table 1: Constructed APM parameters ($P = 768, J = 3, L = 12$, girth=8)

i	$f_i(x)$	$g_i(x)$
0	$763x + 435$	$289x + 496$
1	$679x + 69$	$257x + 640$
2	$397x + 330$	$625x + 200$
3	$61x + 18$	$41x + 524$
4	$697x + 612$	$193x + 672$
5	$373x + 246$	$449x + 672$

Tanner graph has girth 8 (no 4- or 6-cycles). If P is a prime power, there is no set of affine permutations $\sigma_0, \sigma_1, \tau_0, \tau_1$ on $[P]$ that simultaneously satisfy: σ_0 commutes with τ_0 but not with τ_1 , and σ_1 does not commute with τ_0 but commutes with τ_1 . Using the standard semidirect-product representation of $\text{AGL}_1(\mathbb{Z}/p^k\mathbb{Z})$, $(a, b) : x \mapsto ax + b$, and the quadratic commutation condition, one can show this by analyzing the resulting zero-product conditions with p -adic valuation, which contradict non-commutativity. Therefore we choose $P = 768 = 3 \times 2^8$.

6.4 Short Cycles

We begin by presenting a concrete example illustrating Theorem 3.4. For the $J = 3, L = 12$, the block positions

$$[(0, 0), (0, 6), (1, 6), (1, 1), (2, 1), (2, 7), (1, 7), (1, 0)]$$

form an 8-cycle when traversed in order. The corresponding cycle word, in the form of Eq. (1), is

$$W = F_0 G_0^{-1} G_5 F_0^{-1} F_5 G_5^{-1} G_0 F_5^{-1}.$$

Counting such active block 8-cycles is also straightforward. For the active rows 0, 1, 2, the unavoidable pattern uses adjacent left columns $i, i+1$ and adjacent right columns $j, j+1$ (indices modulo $L/2$), and it occurs when $i+j, i+j-1, i+j-2 \in \Delta$. Thus the number of block 8-cycles in the active matrices is $(L/2)|\Delta \cap (\Delta+1) \cap (\Delta+2)|$. For $J = 3, L/2 = 6, \Delta = \{0, 1, 2, 4, 5\}$ and $\Delta \cap (\Delta+1) \cap (\Delta+2) = \{0, 1, 2\}$, so there are 18 block 8-cycles, yielding $18P$ lifted 8-cycles.

Enumerating cycles in the active Tanner graphs of this instance gives 60,512 8-cycles on the X side, and 54,656 8-cycles on the Z side. By Theorem 3.4, 8-cycles are unavoidable in the active Tanner graphs; the block pattern above yields the lower bound $18P = 13,824$. The total exceeds this bound because commutativity holds for many additional pairs and the specific APM parameters create extra cycles beyond block lifts, so the total need not be a multiple of P .

6.5 Trapping set library construction

In our initial experiments, trapping sets of the form shown in Fig. 1 were the dominant cause of BP stalls. Since the constructed code has girth 8, we first enumerate 8-cycles and then build an trapping set library by connecting several 8-cycles to generate trapping sets isomorphic to these patterns. The library is not a collection of only the trapping sets encountered in decoding; instead, we enumerate all trapping sets isomorphic to the dominant patterns and add them exhaustively.

Figure 1 shows three elementary trapping-set patterns: (6, 2) from two length-8 cycles, (12, 2) from five length-8 cycles, and (8, 2) formed by attaching a length-4 check-variable path to the two odd checks of (6,2). For $P = 768, J = 3, L = 12$, the counts were: (6, 2) X-side: 48, Z-side: 16 (total 64); (12, 2) X-side: 23, Z-side: 0 (total 23); and (8, 2) path4 X-side: 48, Z-side: 0 (total 48).

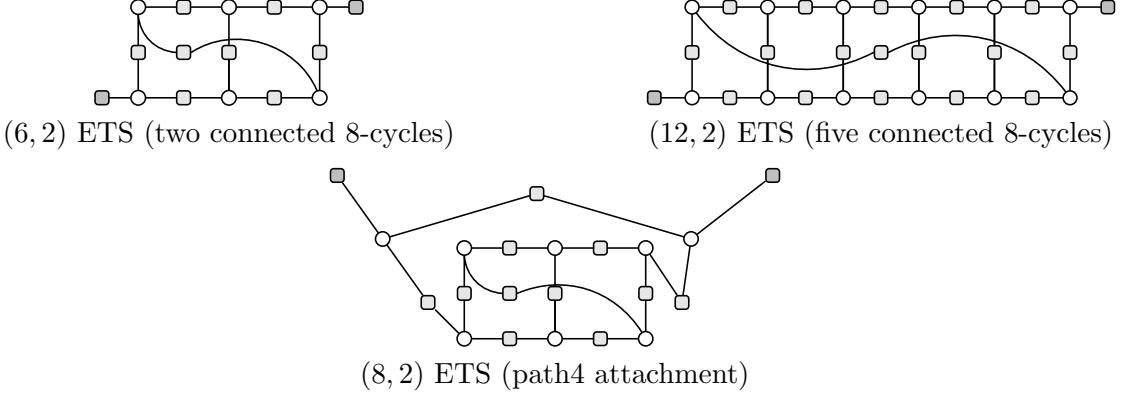


Figure 1: Representative ETSs used in the library.

6.6 Minimum-distance evaluation

Recall the latent-based distances $d_X^{(\text{lat})}$ and $d_Z^{(\text{lat})}$ defined in Section 2. We now construct explicit low-weight vectors in the latent row spaces that satisfy the active parity checks but are not in the duals. Equivalently, any $\mathbf{x} \in \text{Row}(\tilde{H}_X)$ can be written as $\mathbf{x} = (\tilde{H}_X)^\top \mathbf{u}$ for some \mathbf{u} , and likewise for Z . Concretely, we look for coefficient vectors $\mathbf{u} \in \text{Ker}(H_Z(\tilde{H}_X)^\top)$ and $\mathbf{v} \in \text{Ker}(H_X(\tilde{H}_Z)^\top)$, then lift them to $\mathbf{x} = \mathbf{u}^\top \tilde{H}_X$ and $\mathbf{z} = \mathbf{v}^\top \tilde{H}_Z$. Any such \mathbf{x}, \mathbf{z} yields $d_X^{(\text{lat})} \leq \text{wt}(\mathbf{x})$ and $d_Z^{(\text{lat})} \leq \text{wt}(\mathbf{z})$, so below we exploit the block-diagonal form to obtain explicit weight-48 examples.

Let $\mathbf{u}, \mathbf{v} \in \mathbb{F}_2^{3P}$ be coefficient vectors satisfying $\text{diag}(\Psi_3^\top, \Psi_3^\top, \Psi_3^\top)\mathbf{u} = 0$ and $\text{diag}(\Psi_3, \Psi_3, \Psi_3)\mathbf{v} = 0$. Define $\mathbf{x} := \mathbf{u}^\top \tilde{H}_X$ and $\mathbf{z} := \mathbf{v}^\top \tilde{H}_Z$ using the latent matrices \tilde{H}_X and \tilde{H}_Z . Here H_X and H_Z are the active matrices, so $\mathbf{u} \in \text{Ker}(H_Z(\tilde{H}_X)^\top)$ and $\mathbf{v} \in \text{Ker}(H_X(\tilde{H}_Z)^\top)$. With $L/2 = 6, J = 3$ and $\Psi_r = 0$ ($r \neq 3$),

$$H_Z(\tilde{H}_X)^\top = \text{diag}(\Psi_3^\top, \Psi_3^\top, \Psi_3^\top), \quad H_X(\tilde{H}_Z)^\top = \text{diag}(\Psi_3, \Psi_3, \Psi_3).$$

Thus $\mathbf{u} = (\mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5)$ and $\mathbf{v} = (\mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5)$ with $\mathbf{u}_r \in \text{Ker}(\Psi_3^\top)$ and $\mathbf{v}_r \in \text{Ker}(\Psi_3)$ for $r = 3, 4, 5$. Commutativity makes the terms for $u = 2, 3, 4, 5$ vanish, so

$$\Psi_3 = F_0 G_3 + G_3 F_0 + F_1 G_2 + G_2 F_1,$$

and hence

$$\Psi_3^\top = F_0^\top G_3^\top + G_3^\top F_0^\top + F_1^\top G_2^\top + G_2^\top F_1^\top.$$

In this instance, $\text{rank}(\Psi_3) = 576$, so $\dim \text{Ker}(\Psi_3) = \dim \text{Ker}(\Psi_3^\top) = 192$. The four permutation terms in Ψ_3 have disjoint supports, so both row and column weights are 4 and the kernels are generated by 192 disjoint weight-4 codewords supported on the 4-point blocks

$$[t] := \{t, t + 192, t + 384, t + 576\} \subset \mathbb{Z}_{768}, \quad t \in \mathbb{Z}_{192}.$$

The explicit weight-4 vectors constructed above lift to $\mathbf{x} = \mathbf{u}^\top \tilde{H}_X$ and $\mathbf{z} = \mathbf{v}^\top \tilde{H}_Z$ of weight 48, and they are not in C_X^\perp and C_Z^\perp for this instance. This gives the upper bounds $d_X^{(\text{lat})} \leq 48$ and $d_Z^{(\text{lat})} \leq 48$.

Because each APM block preserves the 4-point blocks, any latent vector $\mathbf{x} = \mathbf{u}^\top \tilde{H}_X$ is block-constant within each column block. Compressing each column block to length 192 gives $\bar{\mathbf{x}}$ with $\text{wt}(\mathbf{x}) = 4 \text{wt}(\bar{\mathbf{x}})$. In the compressed space, each generator has weight 12 and overlaps across row blocks are uniformly bounded, so any nonzero $\bar{\mathbf{x}}$ has $\text{wt}(\bar{\mathbf{x}}) \geq 12$. Hence $\text{wt}(\mathbf{x}) \geq 48$, and the same

argument applies to Z , so $d_X^{(\text{lat})} \geq 48$ and $d_Z^{(\text{lat})} \geq 48$ are provable here. Together with the explicit weight-48 examples we conclude $d_X^{(\text{lat})} = d_Z^{(\text{lat})} = 48$. This shows $d_{\min} \leq 48$.

To conclude $d_{\min} = 48$, we must also bound logical operators outside the latent range; in other words, we need a lower bound on the minimum weight of $\mathbf{x} \in C_Z \setminus C_X^\perp, \mathbf{x} \notin \text{Row}(\tilde{H}_X)$. At present we do not have certified lower bounds on these non-latent minima, so the rigorous statement here is only $d_{\min} \leq 48$. In summary, we prove $d_X^{(\text{lat})} = d_Z^{(\text{lat})} = 48$ for this instance, which yields the bound $d_{\min} \leq 48$, while certified lower bounds for non-latent logical operators remain open.

6.7 Frame error rate

Figure 2 reports the FER of the constructed code under BP with post-processing. For reference, we also include the hashing bound for the depolarizing channel in the plot. At $p = 4\%$, the FER reaches 10^{-8} . For each point we collected at least 50 error events; 95% confidence intervals are plotted but may be visually indistinguishable because they are narrow. For FER above 10^{-7} , all failures left hundreds of remaining errors, and we did not observe cases where the decoder returned low-weight logical errors (including higher-weight ones), which is common in small-distance codes. Around 10^{-8} , an error-floor tendency begins to appear, and the dominant failures are stalls trapped by trapping sets of size on the order of tens.

6.8 Density evolution benchmark

To benchmark BP performance, we analyze a non-orthogonal random $(3, 12)$ -regular pair of classical LDPC codes H_X, H_Z drawn independently from the standard configuration-model ensemble. Although this pair is no longer a CSS code and has no direct physical meaning, it provides a useful classical benchmark for estimating noise from syndromes. Under the cycle-free (infinite-length) approximation, we estimate the density-evolution fixed point using a population-dynamics (Monte Carlo) simulation: following the numerical viewpoint emphasized by MacKay [35], we track a large population of messages when analytic density tracking is intractable, iteratively updating it by randomly sampling tree-like neighborhoods with the regular degrees and applying the BP update rules for the depolarizing channel; see [3, 5] for standard density-evolution background. This yields a depolarizing-channel BP threshold of $p \simeq 0.05702$, which we plot in Fig. 2 as a benchmark reference. The proposed code approaches this DE threshold, suggesting it retains sufficient LDPC randomness and that the decoder effectively exploits it.

Data availability

The data that support the findings of this study are available from the author upon reasonable request.

Code availability

The construction algorithm is available at https://github.com/kasaikenta/construct_apm_css_code. The decoding algorithm is available at https://github.com/kasaikenta/joint_BP_plus_PP.

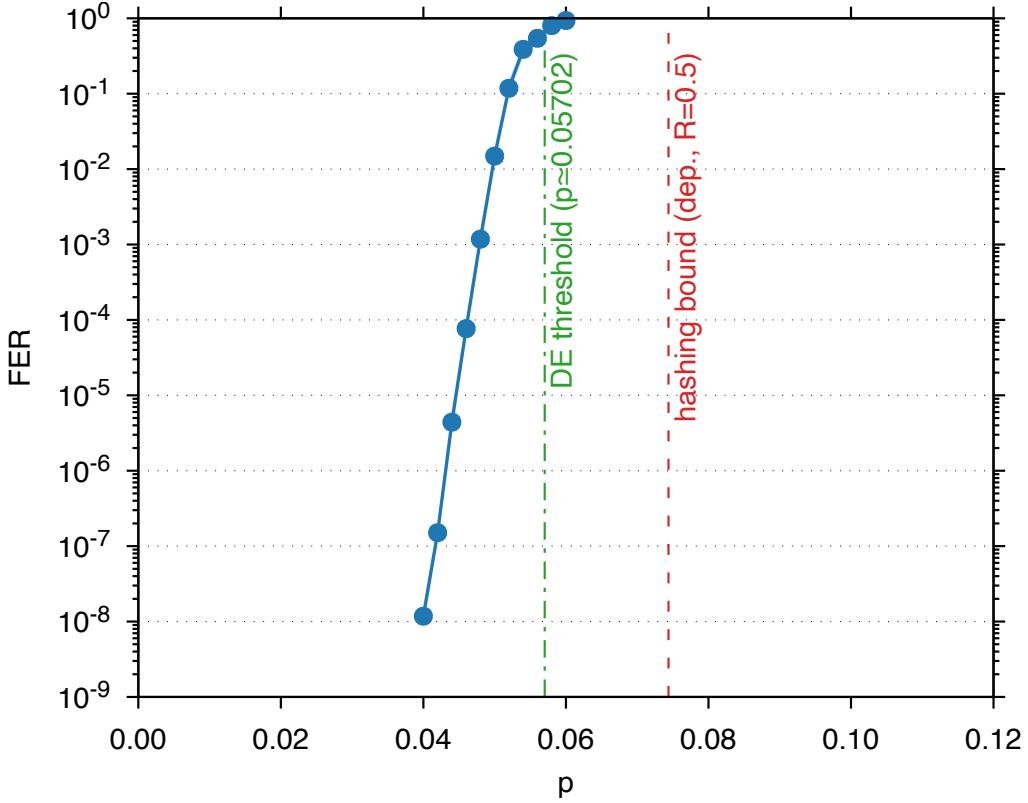


Figure 2: FER curve of the constructed code girth-8, (3,12)-regular $[[9216, 4612, \leq 48]]$, ($P = 768, J = 3, L = 12$) with error bars (95% confidence). For reference, we plot the depolarizing-channel hashing bound and the density-evolution benchmark threshold for a non-orthogonal random (3, 12)-regular H_X, H_Z pair ($p = 0.05702$). Here the number of physical qubits is $n = PL$, the active matrix ranks are $\text{rank}(H_X) = 2302$ and $\text{rank}(H_Z) = 2302$, hence the number of logical qubits is $k = 4612$.

References

- [1] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.
- [2] S. Kudekar, T. J. Richardson, and R. L. Urbanke. Spatially coupled ensembles universally achieve capacity under belief propagation. *IEEE Transactions on Information Theory*, 59(12):7761–7813, 2013.
- [3] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001.
- [4] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.
- [5] T. J. Richardson and R. L. Urbanke. *Modern Coding Theory*. Cambridge University Press, Cambridge, UK, 2008.

- [6] T. J. Richardson. Error floors of LDPC codes. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.
- [7] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.
- [8] D. J. C. MacKay, G. Mitchison, and P. L. McFadden. Sparse-graph codes for quantum error correction. *IEEE Transactions on Information Theory*, 50(10):2315–2330, 2004.
- [9] D. Poulin and Y. Chung. On the iterative decoding of sparse quantum codes. *Physical Review A*, 77:012308, 2008.
- [10] J.-P. Tillich and G. Zémor. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014.
- [11] A. A. Kovalev and L. P. Pryadko. Quantum kronecker sum-product low-density parity-check codes with finite rate. *Physical Review A*, 88:012311, 2013.
- [12] R. Wang and L. P. Pryadko. Distance bounds for generalized bicycle codes. *Symmetry*, 14(7):1348, 2022.
- [13] O. A. Mostad, H.-Y. Lin, E. Rosnes, D.-S. Lee, and C.-Y. Lai. Advancing finite-length quantum error correction using generalized bicycle codes. In *Proc. 13th International Symposium on Topics in Coding (ISTC)*, pages 1–5, Los Angeles, CA, USA, Aug. 2025.
- [14] H.-K. Lin, X. Liu, P. K. Lim, and L. P. Pryadko. Single-shot and two-shot decoding with generalized bicycle codes, 2025. arXiv:2502.19406.
- [15] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86:032324, 2012.
- [16] M. Hagiwara and H. Imai. Quantum quasi-cyclic LDPC codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 806–810, Nice, France, 2007.
- [17] D. G. M. Mitchell, R. Smarandache, and Jr. Costello, D. J. Quasi-cyclic LDPC codes based on pre-lifted protographs. *IEEE Transactions on Information Theory*, 60(10):5856–5874, 2014.
- [18] M.-H. Hsieh, T. A. Brun, and I. Devetak. Entanglement-assisted quantum quasi-cyclic low-density parity-check codes. *Physical Review A*, 79:032340, 2009.
- [19] D. Komoto and K. Kasai. Quantum error correction near the coding theoretical bound. *npj Quantum Information*, 11(1):154, 2025.
- [20] F. Amirzade, D. Panario, and M.-R. Sadeghi. Girth analysis of quantum quasi-cyclic LDPC codes. *Problems of Information Transmission*, 60(2):71–89, 2024.
- [21] K. Kasai. Quantum error correction with girth-16 non-binary LDPC codes via affine permutation construction. In *Proc. 13th International Symposium on Topics in Coding (ISTC)*, pages 1–5, Los Angeles, CA, USA, Aug. 2025.
- [22] D. Ostrev, D. Orsucci, F. Lazaro, and B. Matuz. Classical product code constructions for quantum calderbank–shor–steane codes. *Quantum*, 8:1420, 2024.

- [23] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, 1996.
- [24] A. Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London A*, 452(1954):2551–2577, 1996.
- [25] D. Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997. Ph.D. dissertation; available as arXiv:quant-ph/9705052.
- [26] K. Kasai. Quantum error correction exploiting degeneracy to approach the hashing bound, 2025. arXiv:2506.15636.
- [27] J. Thorpe. Low-density parity-check (LDPC) codes constructed from protographs. Technical Report 42-154, IPN, Aug. 2003.
- [28] M. Fossorier. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. *IEEE Transactions on Information Theory*, 50(8):1788–1793, 2004.
- [29] M. Gholami and M. Alinia. High-performance binary and non-binary low-density parity-check codes based on affine permutation matrices. *IET Communications*, 9(17):2114–2123, 2015.
- [30] R. Yoshida and K. Kasai. Linear permutation polynomial codes. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 66–70. IEEE, July 2019.
- [31] S. Myung, K. Yang, and D. S. Park. A combining method of structured LDPC codes from affine permutation matrices. In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, pages 674–678, 2006.
- [32] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002.
- [33] J. Zhang and M. P. C. Fossorier. A modified weighted bit-flipping decoding of low-density parity-check codes. *IEEE Communications Letters*, 8(3):165–167, 2004.
- [34] I. Dumer. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Transactions on Information Theory*, 44(7):2587–2604, 1998.
- [35] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2003.