Upper Bounds on the Minimum Distance of Quantum APM-LDPC Codes

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

Abstract

This paper studies a previously proposed Calderbank–Shor–Steane code construction method based on affine permutation matrices from the viewpoint of minimum-distance upper bounds. One motivation is that the previously proposed family could plausibly have a minimum distance that increases with blocklength, perhaps linearly, and we ask whether explicit upper-bound evidence can already refute that possibility. For general regular constructions in this framework, we introduce several upper-bound methods. Evaluating them within a common framework, we update the minimum-distance upper bounds reported in the previous paper. We then search, from short to long blocklengths, for codes that make the minimum of the proposed upper bounds as large as possible, and construct a best-code sequence across blocklengths. Over the reported range, the structural upper bounds still grow roughly linearly with blocklength, but decoder-failure witnesses can remain much smaller and in particular produce a weight-18 logical operator at a larger blocklength. Thus the present evidence gives concrete contrary evidence against a naive linear-growth picture, while still not settling the asymptotic behavior of this family.

1 Introduction

The history of low-density parity-check (LDPC) codes is also a history of balancing sparse Tanner graphs, large girth, good minimum distance, and good decoding performance [1, 2]. In the classical setting, this program developed through random regular ensembles, quasi-cyclic constructions, protograph liftings, and algebraic constructions, all of which offered different ways to preserve sparse local structure without sacrificing global distance too severely [3, 4]. In quantum coding theory, this classical line becomes directly relevant through Calderbank–Shor–Steane (CSS) codes, which connect quantum codes to pairs of classical linear codes satisfying an orthogonality relation [5, 6].

For quantum LDPC codes, however, the classical design principles cannot simply be transplanted. One needs two sparse parity-check matrices that are not only individually useful but also mutually compatible. This tension was already visible in early sparse-graph quantum code constructions [7, 8], remained central for hypergraph-product codes with positive rate and square-root distance [9], and still drives the modern literature on asymptotically good quantum LDPC codes and quantum Tanner codes [10, 11]. Much of that line of work is naturally organized around lower bounds and asymptotic existence.

From the viewpoint of minimum-distance evaluation, what one would ideally like is a lower bound for the full code distance, because a lower bound certifies guaranteed error-correction capability rather than merely excluding larger values. In the present family, however, the known lower-bound techniques do not transfer directly to the full distance, and the classical literature on exact minimum-weight search is primarily effective at short and medium blocklengths. The present paper therefore takes the complementary route of making upper bounds as sharp as possible inside one explicit and reproducible construction family; more detailed comments on

the lower-bound literature and on the classical short-code literature are deferred to later sections. As a coarse baseline, if the Tanner graphs of the active matrices themselves both have column weight 3 and girth 8, then Tanner's tree bound implies $d_X \geq 6$ and $d_Z \geq 6$, hence $d \geq 6$ [2].

Affine-permutation-matrix constructions offer a different compromise [12, 13, 14, 15]. Instead of imposing commutativity on the full parent matrix, one enforces orthogonality only on the active part that is actually used as the CSS stabilizer, while allowing controlled non-commutativity in the latent part. This viewpoint makes it possible to search systematically for high-girth binary CSS LDPC codes while keeping track of how the latent structure creates logical obstructions. The closest antecedent is [15], which developed the same $(3, 12)$ family primarily from the construction and decoding side. The present paper is a follow-up on that same construction method, focused on how tightly one can upper-bound and update the minimum distance within the family. Following that line, we refer to this family simply as quantum affine-permutation-matrix-based low-density parity-check (APM-LDPC) codes.

The choice of lift sizes in the present search is not arbitrary. The companion manuscript on affine commutation patterns [16] shows that the cross-commuting nonabelian affine pattern needed on the construction side is obstructed for prime-power moduli and becomes available naturally on Chinese-remainder-theorem-split moduli. For that reason, the computational search here is organized around non-prime-power lift sizes.

This leads to a concrete question. The family proposed in [15] could plausibly have minimum distance that grows with blocklength, perhaps even linearly. The present paper asks whether one can already find contrary evidence, in the form of explicit low-weight logical operators, by running several independent upper-bound methods in parallel. Our current answer is mixed: the previously reported bounds can be improved, and a decoder-failure witness of weight 18 already appears at a larger lift size, but the overall asymptotic behavior of the family remains unsettled.

The goal of this paper is therefore twofold: to develop several minimum-distance upper bounds for the construction method of [15], and to search for codes whose smallest currently known upper bound is as large as possible. First, for general regular constructions in this APM-LDPC framework, we introduce several upper-bound methods. Second, we apply them to the published code of [15] and update the previously reported minimum-distance bound. Third, we search from short to long blocklengths for codes that enlarge the minimum of the proposed bounds and thereby obtain one current best code for each reported lift size. A compact map of the upper-bound methods is given in Section 4, while exact parameter sets, implementation details, and verification or experiment logs are collected in the supplementary materials [17].

2 Parent, Active, and Latent Matrices

Let

$$\hat{H}_X \in \mathbb{F}_2^{r_X \times n}, \qquad \hat{H}_Z \in \mathbb{F}_2^{r_Z \times n}$$

be binary parent matrices with the same number of columns. Choose subsets of rows and call them the active rows. The complementary rows are called latent. After permuting rows if necessary, we may write

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

where H_X, H_Z are the active matrices and $\tilde{H}_X, \tilde{H}_Z$ are the latent matrices. This active–latent viewpoint is inherited from the construction-side analysis of [15]; the present section keeps only the minimum linear-algebraic structure needed for the upper-bound theory. The purpose of this section is to define the distance quantities attached to an arbitrary active–latent decomposition before any specialization to affine permutation matrices, block sizes, or regularity parameters is imposed. The following definition makes that separation explicit at the level of the distance quantities themselves.

Definition 2.1 (Overall, latent, and non-latent distances)**.** From this point on, assume that $H_X H_Z^{\mathsf{T}} = 0$ and define the CSS constituent codes by

$$C_X := \mathrm{Ker}(H_X), \qquad C_Z := \mathrm{Ker}(H_Z).$$

The usual CSS distances are

$$d_X = \min\{\mathrm{wt}(\boldsymbol{x}) : \boldsymbol{x} \in C_Z \setminus C_X^{\perp}\}, \qquad d_Z = \min\{\mathrm{wt}(\boldsymbol{z}) : \boldsymbol{z} \in C_X \setminus C_Z^{\perp}\}.$$

To isolate the contribution coming directly from the latent row spaces, define

$$d_X^{(\mathrm{lat})} := \min\{\mathrm{wt}(\boldsymbol{x}) : \boldsymbol{x} \in (C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}\},$$

$$d_Z^{(\mathrm{lat})} := \min\{\mathrm{wt}(\boldsymbol{z}) : \boldsymbol{z} \in (C_X \cap \mathrm{Row}(\tilde{H}_Z)) \setminus C_Z^{\perp}\}.$$

To measure the complementary contribution outside the latent row spaces, define

$$d_X^{(\mathrm{nlat})} := \min\{\mathrm{wt}(\boldsymbol{x}) : \boldsymbol{x} \in C_Z \setminus C_X^{\perp},\ \boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)\},$$

$$d_Z^{(\mathrm{nlat})} := \min\{\mathrm{wt}(\boldsymbol{z}) : \boldsymbol{z} \in C_X \setminus C_Z^{\perp},\ \boldsymbol{z} \notin \mathrm{Row}(\tilde{H}_Z)\}.$$

Then

$$d_X = \min\{d_X^{(\mathrm{lat})}, d_X^{(\mathrm{nlat})}\}, \qquad d_Z = \min\{d_Z^{(\mathrm{lat})}, d_Z^{(\mathrm{nlat})}\}.$$

□

The point of this definition is to separate the part of the minimum-distance bottleneck that is already explained by the latent row spaces from the part that is not. In that sense, the latent distances measure the latent contribution, whereas the non-latent distances measure the contribution outside the latent side. Later sections evaluate the former structurally and track the latter through explicit upper-bound witnesses.

Example 2.2. Even before specifying a concrete construction, the definition already singles out two different mechanisms for a small CSS distance. One code may have a small latent distance because a low-weight logical representative lies directly in $\mathrm{Row}(\tilde{H}_X)$ or $\mathrm{Row}(\tilde{H}_Z)$. Another code may have a large latent distance but still a small overall distance because a lighter representative exists outside the latent row spaces. The role of the later upper-bound methods is precisely to detect which of these two mechanisms is responsible for the current bottleneck. □

3 APM Specialization

We now specialize the general parent/active/latent framework to the APM family used throughout the computational part. Let L be even and let $F_0, \ldots, F_{L/2-1}$ and $G_0, \ldots, G_{L/2-1}$ be $P \times P$ permutation matrices. We define the parent matrices of the APM-LDPC family by

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

$$(\hat{H}_Z)_{i,j} = G_{i-j}^{\mathsf{T}}, \qquad (\hat{H}_Z)_{i,L/2+j} = F_{i-j}^{\mathsf{T}},$$

with all indices interpreted modulo $L/2$. For the active choice used in this paper, the first J block rows are taken as active and the remaining block rows are latent, in the sense of Section 2.

For each $r \in \mathbb{Z}/(L/2)\mathbb{Z}$, define

$$\Psi_r := \sum_{u=0}^{L/2-1} \left(F_u G_{r-u} + G_{r-u} F_u\right)$$

over $\mathbb{F}_2$. The blocks of $\hat{H}_X \hat{H}_Z^{\mathsf{T}}$ depend only on the row difference and are given by the corresponding Ψ_r.

For the standard active choice, let

$$\Delta := \{(k-i) \bmod (L/2) : 0 \le i, k \le J-1\}.$$

This sufficient criterion is already present in the $(3, 12)$ setting of [15]; here we simply rewrite it in the general (J, L) form. The next theorem states that criterion directly in terms of the mixed residuals Ψ_r.

Theorem 3.1. If $\Psi_r = 0$ for every $r \in \Delta$, then $H_X H_Z^{\mathsf{T}} = 0$. □

Proof. The (i, k) block of $H_X H_Z^{\mathsf{T}}$ is Ψ_{k-i} with indices taken modulo $L/2$. Since $0 \le i, k \le J-1$, the difference $k - i$ always belongs to Δ. Hence $\Psi_r = 0$ for all $r \in \Delta$ implies that every active block vanishes. □

The theorem says that CSS orthogonality can be checked by monitoring only a small difference set. This is what makes the later "single residual interaction" design possible.

Example 3.2. In the concrete family treated computationally in this paper we fix

$$J = 3, \qquad L = 12, \qquad L/2 = 6.$$

Then

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

so the only unconstrained residue is $r = 3$. The search problem is therefore to realize

$$\Psi_r = 0 \quad (r \neq 3), \qquad \Psi_3 \neq 0,$$

which concentrates the latent interaction into one residue class and is intended to make the latent distances introduced in Section 2 the first structural bottleneck. □

Each block is an affine permutation

$$x \mapsto ax + b \pmod{P}, \qquad \gcd(a, P) = 1.$$

This representation is convenient for two reasons. First, commutativity reduces to a linear congruence. Second, if $P = mQ$, the affine rule descends naturally modulo Q, which is exactly what later supports block compression.

Definition 3.3 (Admissible CRT twists)**.** Let $\omega_0 = (F_i^0, G_i^0)_{i=0}^{L/2-1}$ be an affine seed assignment over $\mathbb{Z}/P_0\mathbb{Z}$ satisfying the active commutativity constraints for Δ, and let Q be coprime to P_0. A CRT twist of ω_0 by modulus Q is an affine assignment

$$\eta = (F_i^\eta, G_i^\eta)_{i=0}^{L/2-1}$$

over $\mathbb{Z}/Q\mathbb{Z}$, written as

$$F_i^\eta : x \mapsto \alpha_i x + \beta_i, \qquad G_j^\eta : x \mapsto \gamma_j x + \delta_j \pmod{Q},$$

with $\alpha_i, \gamma_j \in (\mathbb{Z}/Q\mathbb{Z})^\times$ and $\beta_i, \delta_j \in \mathbb{Z}/Q\mathbb{Z}$. The CRT-twisted assignment is

$$\omega_0 \oplus_{\mathrm{CRT}} \eta$$

over $\mathbb{Z}/(P_0 Q)\mathbb{Z}$, obtained by gluing the affine coefficients componentwise through the Chinese remainder theorem.

Let

$$\Gamma_\Delta := \{(u, r-u) \mid 0 \le u < L/2,\ r \in \Delta\},$$

where the second index is read modulo $L/2$. The twist η is called active-commutativity-admissible if

$$\delta_j(\alpha_i - 1) - \beta_i(\gamma_j - 1) \equiv 0 \pmod{Q} \qquad ((i,j) \in \Gamma_\Delta).$$

The set of all such twists is denoted by

$$\mathcal{T}_Q(\Delta).$$

□

This definition is exhaustive for APM-valued CRT twisting: once the seed ω_0 already satisfies the active commutativity constraints modulo P_0, the CRT product satisfies the same active commutativity constraints modulo P_0Q if and only if the auxiliary component η lies in $\mathcal{T}_Q(\Delta)$. Thus admissible CRT twisting is not a single construction but a finite solution set of bilinear congruences over $\mathbb{Z}/Q\mathbb{Z}$.

Remark 3.4 (Tractable subfamilies do not exhaust all twists)**.** The translation twists used in some experiments form the subfamily

$$\alpha_i = \gamma_j = 1, \qquad F_i^\eta : x \mapsto x + \beta_i, \quad G_j^\eta : x \mapsto x + \delta_j.$$

They are automatically active-commutativity-admissible, because translations commute with one another. Centralizer mutations give another tractable subfamily: one multiplies selected blocks by affine maps that commute with their required partners. These two subfamilies are useful for controlled searches, but they do not describe all of $\mathcal{T}_Q(\Delta)$ in general. In particular, admissible twists with nontrivial multipliers α_i or γ_j may satisfy the bilinear congruences collectively without arising from a one-block centralizer mutation of a previously chosen twist. □

From this point onward, all vectors are treated as column vectors. For $A \in \mathbb{F}_2^{r\times n}$, we write

$$\mathrm{Ker}(A) := \{\boldsymbol{x} \in \mathbb{F}_2^n : A\boldsymbol{x} = 0\}.$$

We also write $\mathrm{Row}(A) \subseteq \mathbb{F}_2^n$ for the subspace spanned by the rows of A, identified naturally with a subspace of $\mathbb{F}_2^n$ via the standard inner product.

4 Overview of the Upper-Bound Methods

The three upper bounds that play the most visible role in this paper are the block-compression, cycle-8 ETS, and decoder-failure bounds. In the actual reported best-by-P ladder, however, the current best upper bound is obtained by running several methods in parallel, including the latent, CRT-compression, and direct CSS-search bounds. The purpose of this section is to give a compact map of these mechanisms before the paper turns to their detailed theory.

All six methods are based on the same principle: one light nontrivial logical operator is enough to upper-bound the distance. On the X side, if one finds

$$\boldsymbol{x} \in \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X),$$

then

$$d_X \le \mathrm{wt}(\boldsymbol{x}).$$

Likewise, on the Z side, if

$$\boldsymbol{z} \in \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z),$$

then

$$d_Z \le \mathrm{wt}(\boldsymbol{z}).$$

The difference between the methods lies only in how they try to manufacture such witnesses.

latent upper bound The latent upper bound asks whether the latent row spaces themselves already contain a light logical representative. Restricting candidates to the form $\boldsymbol{x} = \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X$ or $\boldsymbol{z} = \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_Z$ reduces the problem from the full ambient space of length LP to a kernel problem for the mixed products. This method therefore measures the part of the bottleneck created directly by the latent side.

block-compression upper bound The block-compression upper bound searches for periodic witnesses through a shorter quotient code. When $P = mQ$, the subspace of m-block-constant vectors is preserved by the affine action, so a light compressed witness can be lifted back to length LP. If the lifted vector satisfies the CSS condition and stays outside the relevant row spaces, it becomes a non-latent logical operator. The improvement of the published $P = 768$ code from $d \leq 48$ to $d \leq 32$ is the representative example of this method at work.

CRT-compression upper bound The CRT-compression upper bound is a looser structural search than block-compression. For $P = q_1 q_2$ with $\gcd(q_1, q_2) = 1$, it restricts the search to the stripe subspace generated by residue classes modulo q_1 and modulo q_2. Intuitively, this method looks for logical representatives that are not fully block-constant but still exhibit a coprime stripe pattern.

direct CSS-search upper bound The direct CSS-search upper bound is the least structured fallback method. It searches directly inside $\mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)$ and $\mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)$ for low-weight vectors. Conceptually this is just the CSS distance definition itself, but in practice it plays the role of a benchmark method that can catch witnesses missed by more structured searches.

cycle-8 ETS upper bound The cycle-8 ETS upper bound turns local Tanner-graph structure into logical witnesses. Because the family has girth 8, the shortest cycles are 8-cycles, and cycle-8-connected ETSs become the natural local objects to enumerate. Their odd-check boundaries equal their induced syndromes, so $(a, 0)$ ETSs and same-boundary differences of $(a, 2)$ ETSs naturally yield classical candidates. When such candidates survive the row-space test, they become CSS logical operators and hence upper bounds.

decoder-failure upper bound The decoder-failure upper bound converts actual fail logs into distance certificates. If $\boldsymbol{e}$ is the true error and $\hat{\boldsymbol{e}}$ is a decoder estimate with the same syndrome, then the residual $\Delta = \boldsymbol{e} + \hat{\boldsymbol{e}}$ is syndrome-zero. If this residual is a pure X-type or pure Z-type vector outside the corresponding stabilizer row space, then it is already a nontrivial logical operator, and its weight upper-bounds the distance. This is the method that can expose witnesses not predicted by the structural theory.

In summary, the six methods search for the same object, namely a light nontrivial logical operator, from six different viewpoints: latent structure, periodic block compression, CRT stripes, direct nullspace search, local Tanner-graph structure, and decoder failures. Accordingly, the current best upper bound in the reported best-by-P ladder is simply the minimum over all of them. Appendix A plays a different role: it is a lower-bound certification tool showing that, in favorable cases, the latent upper bound is in fact exact on the latent side.

5 Latent Upper Bounds

We now return to a general regular construction in the APM-LDPC framework. Let the active row blocks be indexed by $0, \ldots, J-1$ and the latent row blocks by $J, \ldots, L/2-1$. We work throughout under Definition 2.1. In particular, this section focuses on the latent distances $d_X^{(\mathrm{lat})}$

and $d_Z^{(\mathrm{lat})}$, which measure the part of the minimum-distance bottleneck already explained by the latent row spaces themselves.

Lemma 5.1 (General active–latent mixed product)**.** For every active index $0 \le i \le J-1$ and latent index $0 \le \ell \le L/2 - J - 1$,

$$[H_Z \tilde{H}_X^{\mathsf{T}}]_{i,\ell} = \Psi^{\mathsf{T}}_{(J+\ell-i) \bmod (L/2)}, \qquad [H_X \tilde{H}_Z^{\mathsf{T}}]_{i,\ell} = \Psi_{(J+\ell-i) \bmod (L/2)}.$$

□

Proof. The (i,ℓ) block of $H_Z \tilde{H}_X^{\mathsf{T}}$ is the parent mixed product between active row block i and latent row block $J+\ell$, which is exactly

$$\Psi^{\mathsf{T}}_{(J+\ell-i) \bmod (L/2)}.$$

The $H_X \tilde{H}_Z^{\mathsf{T}}$ formula is analogous. □

The lemma reduces the latent upper-bound problem from the full parent matrix to the kernel of an explicit mixed product. The next proposition turns this observation directly into a witness construction, in the same spirit as the latent-distance analysis developed for this APM framework in earlier work [15].

Proposition 5.2 (Latent upper bounds from mixed-product kernels)**.** The latent distances admit the exact parameterizations

$$d_X^{(\mathrm{lat})} = \min\left\{ \mathrm{wt}(\boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X) : \boldsymbol{\lambda} \in (\mathbb{F}_2^P)^{L/2-J},\ H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda} = 0,\ \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X \notin C_X^{\perp} \right\},$$

$$d_Z^{(\mathrm{lat})} = \min\left\{ \mathrm{wt}(\boldsymbol{v}^{\mathsf{T}} \tilde{H}_Z) : \boldsymbol{v} \in (\mathbb{F}_2^P)^{L/2-J},\ H_X \tilde{H}_Z^{\mathsf{T}} \boldsymbol{v} = 0,\ \boldsymbol{v}^{\mathsf{T}} \tilde{H}_Z \notin C_Z^{\perp} \right\},$$

with the convention that the minimum is $+\infty$ when the corresponding set is empty.

In particular, every feasible coefficient vector $\boldsymbol{\lambda} \in (\mathbb{F}_2^P)^{L/2-J}$ satisfying $H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda} = 0$ and $\boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X \notin C_X^{\perp}$ produces $\boldsymbol{x} := \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X \in (C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}$, hence $d_X^{(\mathrm{lat})} \le \mathrm{wt}(\boldsymbol{x})$ and therefore $d_X \le \mathrm{wt}(\boldsymbol{x})$. The Z-side statement is identical. □

Proof. We prove the X-side identity. By Definition 2.1,

$$d_X^{(\mathrm{lat})} = \min\{\mathrm{wt}(\boldsymbol{x}) : \boldsymbol{x} \in (C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}\}.$$

Let

$$\mathcal{L}_X := \left\{ \boldsymbol{\lambda} \in (\mathbb{F}_2^P)^{L/2-J} : H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda} = 0,\ \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X \notin C_X^{\perp} \right\}.$$

If $\boldsymbol{\lambda} \in \mathcal{L}_X$ and $\boldsymbol{x} := \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X$, then $\boldsymbol{x} \in \mathrm{Row}(\tilde{H}_X)$ and $H_Z \boldsymbol{x}^{\mathsf{T}} = H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda} = 0$, so $\boldsymbol{x} \in C_Z$; by definition of $\mathcal{L}_X$, we also have $\boldsymbol{x} \notin C_X^{\perp}$. Hence every feasible coefficient vector produces an element of $(C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}$.

Conversely, if $\boldsymbol{x} \in (C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}$, then $\boldsymbol{x} \in \mathrm{Row}(\tilde{H}_X)$, so there exists some $\boldsymbol{\lambda} \in (\mathbb{F}_2^P)^{L/2-J}$ with $\boldsymbol{x} = \boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X$. Because $\boldsymbol{x} \in C_Z$, $H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda} = H_Z \boldsymbol{x}^{\mathsf{T}} = 0$, and since $\boldsymbol{x} \notin C_X^{\perp}$, this $\boldsymbol{\lambda}$ belongs to $\mathcal{L}_X$.

Therefore the image set

$$\{\boldsymbol{\lambda}^{\mathsf{T}} \tilde{H}_X : \boldsymbol{\lambda} \in \mathcal{L}_X\}$$

coincides exactly with

$$(C_Z \cap \mathrm{Row}(\tilde{H}_X)) \setminus C_X^{\perp}.$$

This proves the displayed formula for $d_X^{(\mathrm{lat})}$. Finally, since $d_X = \min\{d_X^{(\mathrm{lat})}, d_X^{(\mathrm{nlat})}\}$, every latent witness also yields the overall upper bound $d_X \le \mathrm{wt}(\boldsymbol{x})$. The Z-side argument is identical. □

The proposition says that one nonzero kernel vector in the mixed product already produces a concrete latent logical witness. In practice, this is the first structural certificate returned by the search code.

Example 5.3. In the special case central to this paper, namely $(J, L) = (3, 12)$ with $\Psi_r = 0$ for $r \neq 3$, Lemma 5.1 simplifies to

$$H_Z \tilde{H}_X^{\mathsf{T}} = \mathrm{diag}(\Psi_3^{\mathsf{T}}, \Psi_3^{\mathsf{T}}, \Psi_3^{\mathsf{T}}), \qquad H_X \tilde{H}_Z^{\mathsf{T}} = \mathrm{diag}(\Psi_3, \Psi_3, \Psi_3),$$

so a single nonzero kernel vector of Ψ_3 or Ψ_3^{T} already generates a latent witness. □

The exact latent lower-bound theory based on block-constant compression, together with its rank-test and satisfiability (SAT) / satisfiability-modulo-theories (SMT) certification machinery, is moved to Appendix A so that the main text can keep the focus on the upper-bound story. In what follows, we treat that exact-certification theory as background and proceed to the non-latent structural upper bounds.

6 Non-Latent Structural Upper Bounds

6.1 Classical Prototype of the Block-Compression Upper Bound

The previous two sections established the latent upper bound and the matching exact latent lower bound. Before turning to the non-latent block-compression upper bound, we now isolate the purely classical mechanism underneath it: an APM acts on a block-constant subspace through an exact quotient action. The point of doing so is to separate the classical compression/lift argument from the additional CSS row-space conditions that will be imposed afterward. Accordingly, this section should be read as the classical prototype that immediately precedes the quantum non-latent upper bound of the next section.

Assume $P = mQ$ with $m \geq 2$. For $t \in \mathbb{Z}_Q$, define the coset

$$[t]_m := \{t, t + Q, t + 2Q, \ldots, t + (m-1)Q\} \subset \mathbb{Z}_P.$$

A length-P vector is called *m-block constant* if it is constant on every coset $[t]_m$.

Let $U_m(P) \subset \mathbb{F}_2^P$ be the m-block-constant subspace. Define the compression and lift maps

$$\pi_m : U_m(P) \to \mathbb{F}_2^Q, \qquad \iota_m : \mathbb{F}_2^Q \to U_m(P)$$

by

$$[\pi_m(\boldsymbol{x})]_t := \boldsymbol{x}_t, \qquad [\iota_m(\bar{\boldsymbol{x}})]_{t+jQ} := \bar{\boldsymbol{x}}_t.$$

Applying these maps blockwise over N blocks yields

$$\pi_{m,N} : U_m(NP) \to \mathbb{F}_2^{NQ}, \qquad \iota_{m,N} : \mathbb{F}_2^{NQ} \to U_m(NP),$$

with

$$\mathrm{wt}(\iota_{m,N}(\bar{\boldsymbol{x}})) = m\,\mathrm{wt}(\bar{\boldsymbol{x}}).$$

The point of this construction is that a length-P problem may be reinterpreted as a length-Q problem if the relevant matrices preserve block-constant structure. This is the APM analogue of the quotient viewpoint familiar from classical quasi-cyclic (QC) LDPC code constructions [3, 4].

Example 6.1 (Worked example ($P = 12,\ m = 3,\ Q = 4$))**.** It is useful to see one explicit toy example. In this case the cosets are

$$[0]_3 = \{0, 4, 8\}, \quad [1]_3 = \{1, 5, 9\}, \quad [2]_3 = \{2, 6, 10\}, \quad [3]_3 = \{3, 7, 11\}.$$

Take

$$\bar{\boldsymbol{x}} = (1, 0, 1, 0) \in \mathbb{F}_2^4.$$

Its lift is

$$\iota_3(\bar{\boldsymbol{x}}) = (1, 0, 1, 0,\ 1, 0, 1, 0,\ 1, 0, 1, 0) \in \mathbb{F}_2^{12},$$

which is indeed constant on every coset. The weight scales exactly as

$$\mathrm{wt}(\bar{\boldsymbol{x}}) = 2, \qquad \mathrm{wt}(\iota_3(\bar{\boldsymbol{x}})) = 6 = 3 \cdot 2.$$

Hence, if the compressed length-4 code contains a codeword of weight 2, then the original length-12 code automatically contains a lifted witness of weight 6.

Now consider the affine permutation

$$x \mapsto x + 1 \pmod{12}.$$

It permutes the cosets by

$$[0]_3 \mapsto [1]_3, \quad [1]_3 \mapsto [2]_3, \quad [2]_3 \mapsto [3]_3, \quad [3]_3 \mapsto [0]_3.$$

Therefore, on the quotient side, it becomes exactly the length-4 permutation

$$t \mapsto t + 1 \pmod{4}.$$

This is the concrete content of Lemma 6.2: the action of a long APM on the block-constant subspace is identical to the action of a shorter quotient APM. □

Lemma 6.2 (Descent of APMs on block-constant subspaces)**.** Let M be the $P \times P$ APM corresponding to $x \mapsto ax + b \pmod{P}$, and let $\bar{M}$ be the quotient APM obtained by reducing the same affine rule modulo $Q = P/m$. Then M preserves $U_m(P)$. Moreover, for every $\bar{\boldsymbol{u}} \in \mathbb{F}_2^Q$,

$$M\iota_m(\bar{\boldsymbol{u}}) = \iota_m(\bar{M}\bar{\boldsymbol{u}}), \qquad \pi_m(M\iota_m(\bar{\boldsymbol{u}})) = \bar{M}\bar{\boldsymbol{u}}.$$

The same commutativity holds blockwise for any block matrix made of APM blocks. □

Proof. By definition of an APM,

$$M\boldsymbol{e}_t = \boldsymbol{e}_{at+b}.$$

For every $t \in \mathbb{Z}_Q$ and $s \in \{0, \ldots, m-1\}$,

$$a(t + sQ) + b \equiv at + b \pmod{Q},$$

so the affine permutation maps the coset $[t]_m$ to $[at + b]_m$. Hence the image of an m-block-constant vector is again m-block constant, and the action on cosets is exactly the quotient affine map modulo Q. The displayed identities follow directly from the definitions of π_m and ι_m. □

This lemma ensures that block compression is not a heuristic reduction. It is an exact quotient/lift relation compatible with the Tanner structure.

With this preparation, distance upper bounds for classical APM-LDPC codes can be obtained from compressed codewords. The next proposition is the classical prototype of Proposition 6.6 in Section 6.2, namely the CSS block-compression upper bound used later. Its proof follows the standard QC-LDPC strategy of finding a low-weight codeword in the quotient and lifting it back to the long code [3, 4].

Proposition 6.3 (m-block-constant upper bounds for classical APM-LDPC codes)**.** Let $H \in \mathbb{F}_2^{RP\times NP}$ be a classical parity-check matrix made of $R\times N$ blocks of size $P\times P$, each block being an APM. Let $C(H) := \mathrm{Ker}(H)$. For any divisor $m \mid P$, let

$$\bar{H}^{(m)} \in \mathbb{F}_2^{RQ\times NQ}$$

be the compressed matrix obtained by reducing each block modulo $Q = P/m$. Then every nonzero $\bar{\boldsymbol{c}} \in \mathrm{Ker}(\bar{H}^{(m)})$ yields $\boldsymbol{c} := \iota_{m,N}(\bar{\boldsymbol{c}}) \in C(H)$ with $\mathrm{wt}(\boldsymbol{c}) = m\,\mathrm{wt}(\bar{\boldsymbol{c}})$. Therefore $d(C(H)) \le m\,d(\mathrm{Ker}(\bar{H}^{(m)}))$. More generally, every explicit compressed witness $\bar{\boldsymbol{c}} \neq 0$ gives $d(C(H)) \le m\,\mathrm{wt}(\bar{\boldsymbol{c}})$. □

Proof. Let $\bar{\boldsymbol{c}} \in \mathrm{Ker}(\bar{H}^{(m)})$ be nonzero and set $\boldsymbol{c} = \iota_{m,N}(\bar{\boldsymbol{c}})$. Write

$$\boldsymbol{c} = (\boldsymbol{c}^{(0)}, \dots, \boldsymbol{c}^{(N-1)}), \qquad \bar{\boldsymbol{c}} = (\bar{\boldsymbol{c}}^{(0)}, \dots, \bar{\boldsymbol{c}}^{(N-1)})$$

in block form. Applying Lemma 6.2 blockwise gives

$$\sum_{n=0}^{N-1} H_{r,n}\boldsymbol{c}^{(n)} = \iota_m\left(\sum_{n=0}^{N-1} \bar{H}^{(m)}_{r,n}\bar{\boldsymbol{c}}^{(n)}\right)$$

for each block row r. Hence $H\boldsymbol{c}$ is the lift of $\bar{H}^{(m)}\bar{\boldsymbol{c}}$, so $\bar{H}^{(m)}\bar{\boldsymbol{c}} = 0$ implies $H\boldsymbol{c} = 0$. The weight formula follows from the definition of the lift. □

Example 6.4 (A toy instance of Proposition 6.3)**.** Consider a slightly less trivial APM-LDPC example with $R = 1,\ N = 2,\ P = 4$. Let the first block be the identity permutation I_4, and let the second block be the APM corresponding to the affine permutation

$$x \mapsto x + 1 \pmod 4,$$

namely

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Thus

$$H = \begin{bmatrix} I_4 & M \end{bmatrix} \in \mathbb{F}_2^{4\times 8}.$$

Take $m = 2$ and hence $Q = 2$. The compressed matrix is

$$\bar{H}^{(2)} = \begin{bmatrix} I_2 & \bar{M} \end{bmatrix} \in \mathbb{F}_2^{2\times 4}, \qquad \bar{M} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Now consider

$$\bar{\boldsymbol{c}} = (1,0,0,1)^{\mathsf{T}}.$$

Since

$$\bar{H}^{(2)}\bar{\boldsymbol{c}} = \begin{bmatrix} I_2 & \bar{M} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = (1,0)^{\mathsf{T}} + \bar{M}(0,1)^{\mathsf{T}} = (1,0)^{\mathsf{T}} + (1,0)^{\mathsf{T}} = 0,$$

we have $\bar{\boldsymbol{c}} \in \mathrm{Ker}(\bar{H}^{(2)})$, and its weight is 2. Lifting blockwise gives

$$\boldsymbol{c} = \iota_{2,2}(\bar{\boldsymbol{c}}) = (1,0,1,0,\ 0,1,0,1)^{\mathsf{T}} \in \mathbb{F}_2^8.$$

Proposition 6.3 then implies that $\boldsymbol{c} \in C(H)$ and

$$\mathrm{wt}(\boldsymbol{c}) = 4 = 2 \cdot \mathrm{wt}(\bar{\boldsymbol{c}}).$$

Thus the compressed witness is already nontrivial: the quotient APM $\bar{M}$ in the second block cancels the first block at weight 2, and this witness lifts to a block-constant codeword in the long code. Hence this toy code satisfies

$$d(C(H)) \le 4$$

because the compressed quotient already contains a weight-2 witness. □

The importance of this proposition is that it reduces the search for low-weight classical codewords in the long code to a search in the compressed quotient code. The CSS block-compression bounds of this paper are obtained by imposing additional row-space conditions on top of this classical mechanism.

6.2 The Block-Compression Upper Bound Outside the Latent Row Space

Latent witnesses do not exhaust all logical operators. One must also account for representatives outside the latent row spaces $\mathrm{Row}(\tilde{H}_X)$ and $\mathrm{Row}(\tilde{H}_Z)$. This is exactly the contribution measured by $d_X^{(\mathrm{nlat})}$ and $d_Z^{(\mathrm{nlat})}$ in Definition 2.1. In this section we do not try to determine those quantities exactly. Instead, we control them through explicit row-space-external upper-bound witnesses. The quantities $d_X^{(\mathrm{nlat})}$ and $d_Z^{(\mathrm{nlat})}$ introduced in Section 2 are meant to measure exactly this contribution outside the latent side. In this section we do not attempt to evaluate them exactly. Instead, we derive explicit upper bounds on them from block-compressed row-space-external witnesses.

Fix $m \mid P$ and consider the same subspace

$$U_m(LP) \subset \mathbb{F}_2^{LP}.$$

Since each APM descends modulo $Q = P/m$, the active matrices descend to compressed check matrices

$$\bar{H}_X^{(m)}, \bar{H}_Z^{(m)} \in \mathbb{F}_2^{JQ \times LQ}.$$

Before using them, one must verify that the active syndrome condition is preserved by compression.

Lemma 6.5 (Compression equivalence for active checks)**.** For every $\boldsymbol{x} \in U_m(LP)$ and $\bar{\boldsymbol{x}} = \pi_{m,L}(\boldsymbol{x})$,

$$H_Z \boldsymbol{x}^{\mathsf{T}} = 0 \iff \bar{H}_Z^{(m)} \bar{\boldsymbol{x}}^{\mathsf{T}} = 0.$$

Similarly, for every $\boldsymbol{z} \in U_m(LP)$ and $\bar{\boldsymbol{z}} = \pi_{m,L}(\boldsymbol{z})$,

$$H_X \boldsymbol{z}^{\mathsf{T}} = 0 \iff \bar{H}_X^{(m)} \bar{\boldsymbol{z}}^{\mathsf{T}} = 0.$$

Moreover,

$$\mathrm{wt}(\boldsymbol{x}) = m\,\mathrm{wt}(\bar{\boldsymbol{x}}), \qquad \mathrm{wt}(\boldsymbol{z}) = m\,\mathrm{wt}(\bar{\boldsymbol{z}}).$$

□

Proof. Apply Lemma 6.2 blockwise to the quotient blocks of the active matrices. The active syndrome of a lifted block-constant vector is exactly the lifted syndrome computed by the compressed matrix. The weight formula follows from the lift. □

This lemma guarantees that a quotient witness is not merely approximate: if it satisfies the compressed syndrome condition, then its lift satisfies the original active syndrome condition exactly.

Proposition 6.6 (m-block row-space-external upper bounds)**.** Let $m \mid P$ and $Q = P/m$. Take $\bar{\boldsymbol{x}} \in \mathrm{Ker}(\bar{H}_Z^{(m)})$ and let $\boldsymbol{x} = \iota_{m,L}(\bar{\boldsymbol{x}})$. If $\boldsymbol{x} \notin \mathrm{Row}(H_X)$ and $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, then $\boldsymbol{x}$ is a valid X logical representative outside the latent row space, and $d_X \le \mathrm{wt}(\boldsymbol{x}) = m\,\mathrm{wt}(\bar{\boldsymbol{x}})$. Similarly, if $\bar{\boldsymbol{z}} \in \mathrm{Ker}(\bar{H}_X^{(m)})$ lifts to $\boldsymbol{z} = \iota_{m,L}(\bar{\boldsymbol{z}})$ with $\boldsymbol{z} \notin \mathrm{Row}(H_Z) \cup \mathrm{Row}(\tilde{H}_Z)$, then $d_Z \le \mathrm{wt}(\boldsymbol{z}) = m\,\mathrm{wt}(\bar{\boldsymbol{z}})$. □

Proof. By Lemma 6.5, the lift $\boldsymbol{x}$ satisfies $H_Z \boldsymbol{x}^{\mathsf{T}} = 0$, so $\boldsymbol{x} \in C_Z$. If additionally $\boldsymbol{x} \notin \mathrm{Row}(H_X) = C_X^{\perp}$, then $\boldsymbol{x}$ is a valid X logical representative. The extra condition $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$ ensures that it lies outside the latent row space. Hence $d_X \le \mathrm{wt}(\boldsymbol{x}) = m\,\mathrm{wt}(\bar{\boldsymbol{x}})$. The Z case is the same. □

Example 6.7. For the exact code printed in Table 1 of [15], this proposition directly yields the improvement to $d \le 32$. Indeed, the exact transcription and supporting witness archived in the supplementary materials [17] record an explicit X-side witness with block factor $m = 4$. On the compressed side there is a vector $\bar{\boldsymbol{x}} \in \mathrm{Ker}(\bar{H}_Z^{(4)})$ of weight 8. Its 4-block-constant lift $\boldsymbol{x} = \iota_{4,12}(\bar{\boldsymbol{x}})$ therefore satisfies

$$\mathrm{wt}(\boldsymbol{x}) = 4 \cdot 8 = 32,$$

and the recorded row-space checks show that

$$\boldsymbol{x} \notin \mathrm{Row}(H_X), \qquad \boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X).$$

Hence Proposition 6.6 gives $d_X \le 32$, and therefore the overall upper bound for this published code becomes $d \le 32$. By contrast, the previous section shows that the latent side stays exactly at 48, so this is a concrete example where the non-latent block-compression witness is the decisive one. □

The proposition shows that compressed witnesses can remain useful even when the latent row space no longer explains the dominant obstruction. In practice, one scans all divisors $m \mid P$ and records the lightest valid witness among them.

Remark 6.8. If one wants to exclude not only the latent row space itself but also all representatives equivalent to it modulo active stabilizers, the condition should be strengthened to

$$\boldsymbol{x} \notin \mathrm{Row}(H_X) + \mathrm{Row}(\tilde{H}_X),$$

and analogously on the Z side. In this paper we use the weaker condition because it matches the row-space test implemented in the current search logs. □

7 Additional Upper-Bound Methods

At this point the paper has developed two structural upper-bound mechanisms, one tied directly to the latent row spaces and one tied to block-compressed row-space-external witnesses. The remaining implemented methods serve a different role: they look for lighter witnesses that are not predicted a priori by either structural mechanism. The first two are still linear-algebraic in nature, namely the CRT-compression upper bound and the direct CSS-search upper bound. The last two come from local Tanner-subgraph patterns and from actual decoding failures. Together these four auxiliary methods monitor whether the true bottleneck lies in a weaker algebraic search space, in a local graph pattern, or in a witness exposed only through decoding traces.

7.1 The CRT-Compression Upper Bound

The block-compression upper bound restricts the search space to vectors that are constant on residue classes modulo a divisor $m \mid P$. If $P = q_1 q_2$ with $\gcd(q_1, q_2) = 1$, there is another natural coarse quotient: the Chinese remainder decomposition of $\mathbb{Z}_P$ into two coprime directions. The CRT-compression method implemented here does not attempt a full Fourier-mode search. Instead, it restricts attention to the stripe subspace generated by residue classes modulo q_1 and modulo q_2.

Definition 7.1 (CRT stripe subspace)**.** Assume that $P = q_1 q_2$ and $\gcd(q_1, q_2) = 1$. Define the single-block CRT stripe subspace by

$$\mathcal{S}^{(1)}_{q_1,q_2} := \operatorname{span}\Big(\{\mathbf{1}_{\{x\in\mathbb{Z}_P:\, x\equiv r \pmod{q_1}\}} : 0 \le r < q_1\} \cup \{\mathbf{1}_{\{x\in\mathbb{Z}_P:\, x\equiv s \pmod{q_2}\}} : 0 \le s < q_2\}\Big) \subseteq \mathbb{F}_2^P.$$

The full-length CRT stripe subspace is then the blockwise direct sum

$$\mathcal{S}^{\mathrm{crt}}_{q_1,q_2} := \bigoplus_{j=0}^{L-1} \mathcal{S}^{(1)}_{q_1,q_2} \subseteq \mathbb{F}_2^{LP}.$$

□

The point of this definition is that it enlarges the search space beyond block-constant modes while remaining far smaller than the unrestricted ambient space. In that sense it is an intermediate algebraic method between block-compression and a fully structure-free search.

Proposition 7.2 (CRT-compression upper bound)**.** Assume that $P = q_1 q_2$ and $\gcd(q_1, q_2) = 1$. Define

$$u^{\mathrm{crt}}_X(q_1, q_2) := \min\left\{\mathrm{wt}(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{S}^{\mathrm{crt}}_{q_1,q_2} \cap \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)\right\},$$
$$u^{\mathrm{crt}}_Z(q_1, q_2) := \min\left\{\mathrm{wt}(\boldsymbol{z}) : \boldsymbol{z} \in \mathcal{S}^{\mathrm{crt}}_{q_1,q_2} \cap \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)\right\},$$

with the convention that the minimum is $+\infty$ when the corresponding set is empty. Then $d_X \le u^{\mathrm{crt}}_X(q_1, q_2)$ and $d_Z \le u^{\mathrm{crt}}_Z(q_1, q_2)$.

In particular, if $\boldsymbol{x} \in \mathcal{S}^{\mathrm{crt}}_{q_1,q_2} \cap \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)$, then $d_X \le \mathrm{wt}(\boldsymbol{x})$. If in addition $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, then also $d^{(\mathrm{nlat})}_X \le \mathrm{wt}(\boldsymbol{x})$. Likewise, if $\boldsymbol{z} \in \mathcal{S}^{\mathrm{crt}}_{q_1,q_2} \cap \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)$, then $d_Z \le \mathrm{wt}(\boldsymbol{z})$. If moreover $\boldsymbol{z} \notin \mathrm{Row}(\tilde{H}_Z)$, then $d^{(\mathrm{nlat})}_Z \le \mathrm{wt}(\boldsymbol{z})$. □

Proof. We treat the X side. Set

$$\mathcal{W}_X(q_1, q_2) := \mathcal{S}^{\mathrm{crt}}_{q_1,q_2} \cap \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X).$$

Every vector $\boldsymbol{x} \in \mathcal{W}_X(q_1, q_2)$ is, by construction, a valid X-type non-stabilizer logical representative. Therefore the CSS definition implies $d_X \le \mathrm{wt}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathcal{W}_X(q_1, q_2)$. If the set is nonempty, taking the minimum over it yields

$$d_X \le \min_{\boldsymbol{x}\in\mathcal{W}_X(q_1,q_2)} \mathrm{wt}(\boldsymbol{x}) = u^{\mathrm{crt}}_X(q_1, q_2).$$

If the set is empty, the convention $u^{\mathrm{crt}}_X(q_1, q_2) = +\infty$ makes the inequality trivial.

If in addition $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, then $\boldsymbol{x} \in C_Z \setminus C_X^{\perp}$ and $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, so Definition 2.1 gives $d^{(\mathrm{nlat})}_X \le \mathrm{wt}(\boldsymbol{x})$. The Z side is identical. □

Example 7.3 (A toy CRT witness)**.** Take one block of length $P = 6 = 2 \cdot 3$, so $(q_1, q_2) = (2, 3)$. The residue-class stripes modulo 2 are

$$\boldsymbol{a}_0 = (1, 0, 1, 0, 1, 0)^{\mathsf{T}}, \qquad \boldsymbol{a}_1 = (0, 1, 0, 1, 0, 1)^{\mathsf{T}},$$

and the stripes modulo 3 are

$$\boldsymbol{b}_0 = (1, 0, 0, 1, 0, 0)^{\mathsf{T}}, \qquad \boldsymbol{b}_1 = (0, 1, 0, 0, 1, 0)^{\mathsf{T}}, \qquad \boldsymbol{b}_2 = (0, 0, 1, 0, 0, 1)^{\mathsf{T}}.$$

Hence

$$\boldsymbol{x} := \boldsymbol{a}_0 + \boldsymbol{b}_0 = (0, 0, 1, 1, 1, 0)^{\mathsf{T}}$$

lies in $\mathcal{S}^{(1)}_{2,3}$. At the same time, $\boldsymbol{x}$ is neither 2-block constant nor 3-block constant, so the CRT stripe space is strictly larger than the one-period block-compression spaces.

To illustrate Proposition 7.2, consider the toy CSS pair

$$H_X = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \qquad H_Z = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

for which $H_X H_Z^{\mathsf{T}} = 0$ over $\mathbb{F}_2$. One checks directly that $H_Z \boldsymbol{x}^{\mathsf{T}} = 0$ while $\boldsymbol{x} \notin \mathrm{Row}(H_X) = \{0, (1,0,1,0,1,0)\}$. Therefore $\boldsymbol{x} \in \mathcal{S}^{(1)}_{2,3} \cap \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)$, and the proposition yields the explicit upper bound $d_X \le \mathrm{wt}(\boldsymbol{x}) = 3$. □

In the implementation, the method scans all coprime factor pairs (q_1, q_2), forms the restricted nullspace inside $\mathcal{S}^{\mathrm{crt}}_{q_1,q_2}$, and records the lightest vector that survives the row-space test. The CRT-compression upper bound is the minimum over these coprime stripe quotients.

7.2 The Direct CSS-Search Upper Bound

The previous structural methods all search inside prescribed subspaces. The direct CSS-search method removes that structural restriction and searches directly for low-weight logical representatives satisfying the CSS condition itself. Conceptually this is close to the classical literature on low-weight codeword search and information-set decoding [18], although the present implementation is much lighter: it performs a randomized search inside the CSS nullspaces rather than a full ISD attack.

Proposition 7.4 (Direct CSS-search upper bound)**.** Let

$$\mathcal{W}_X \subseteq \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X), \qquad \mathcal{W}_Z \subseteq \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)$$

be any finite candidate sets returned by a direct search routine. If they are nonempty, define

$$u_X^{\mathrm{dir}} := \min_{\boldsymbol{x} \in \mathcal{W}_X} \mathrm{wt}(\boldsymbol{x}), \qquad u_Z^{\mathrm{dir}} := \min_{\boldsymbol{z} \in \mathcal{W}_Z} \mathrm{wt}(\boldsymbol{z}).$$

Then $d_X \le u_X^{\mathrm{dir}}$ and $d_Z \le u_Z^{\mathrm{dir}}$.

In particular, if $\boldsymbol{x} \in \mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)$, then $d_X \le \mathrm{wt}(\boldsymbol{x})$. Likewise, if $\boldsymbol{z} \in \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)$, then $d_Z \le \mathrm{wt}(\boldsymbol{z})$. Moreover, if some $\boldsymbol{x} \in \mathcal{W}_X$ also satisfies $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, then $d_X^{(\mathrm{nlat})} \le \mathrm{wt}(\boldsymbol{x})$, and analogously $\boldsymbol{z} \in \mathcal{W}_Z$ with $\boldsymbol{z} \notin \mathrm{Row}(\tilde{H}_Z)$ implies $d_Z^{(\mathrm{nlat})} \le \mathrm{wt}(\boldsymbol{z})$. □

Proof. We treat the X side. Every vector in $\mathcal{W}_X$ belongs to $\mathrm{Ker}(H_Z) \setminus \mathrm{Row}(H_X)$, so every such vector is already an X-type non-stabilizer logical representative. Therefore $d_X \le \mathrm{wt}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathcal{W}_X$. If $\mathcal{W}_X$ is nonempty, taking the minimum over the sampled candidate set gives

$$d_X \le \min_{\boldsymbol{x} \in \mathcal{W}_X} \mathrm{wt}(\boldsymbol{x}) = u_X^{\mathrm{dir}}.$$

The one-vector statement is the special case $\mathcal{W}_X = \{\boldsymbol{x}\}$.

If moreover $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, then $\boldsymbol{x} \in C_Z \setminus C_X^{\perp}$ and $\boldsymbol{x} \notin \mathrm{Row}(\tilde{H}_X)$, so Definition 2.1 yields $d_X^{(\mathrm{nlat})} \le \mathrm{wt}(\boldsymbol{x})$. The Z side is identical. □

The proposition itself is immediate, but its role is important: it supplies a benchmark method that makes no use of latent structure, block compression, or CRT stripes. On the classical side, the literature of Leon [19], Stern [18], Canteaut–Chabaud [20], Brouwer–Zimmermann-type refinements and implementations [21], and Grassl's computational search and code tables [22, 23] shows that minimum-weight search can be highly effective at short and medium blocklengths. The present method is closest in spirit to that line, but the target here is a logical representative outside the relevant stabilizer row space, so the implementation is kept as a lightweight randomized search inside the CSS nullspaces rather than a direct import of the classical routines. The current implementation computes bases of $\mathrm{Ker}(H_Z)$ and $\mathrm{Ker}(H_X)$, samples many small random linear combinations, and keeps only the survivors that fall outside the active row spaces. Hence the resulting direct CSS-search upper bound is heuristic rather than exhaustive, but it can detect logical witnesses missed by all more structured algebraic scans.

7.3 The Cycle-8 ETS Upper Bound

All codes considered in this paper have girth 8, so the shortest cycles in the active Tanner graph have length 8. This makes cycle-8 structures the natural local generators of low-weight trapping phenomena. In the classical LDPC literature, trapping sets and elementary trapping sets are standard tools for describing such local obstructions [24, 25]. Here we adapt that viewpoint to girth-8 APM-LDPC codes and use it as a distance-upper-bound method.

Definition 7.5 (Cycle-8-connected elementary trapping set)**.** Let $H \in \mathbb{F}_2^{m\times n}$ and let $G(H)$ be the corresponding Tanner graph. For a variable-node set $S \subseteq V(H)$, define

$$\Gamma(S) := \{c \in C(H) : N(c) \cap S \neq \emptyset\}, \qquad d_S(c) := |N(c) \cap S|.$$

Define the odd-check boundary by

$$\partial S := \{c \in \Gamma(S) : d_S(c) \equiv 1 \pmod 2\}.$$

If every $c \in \Gamma(S)$ satisfies

$$d_S(c) \in \{1, 2\}$$

and the induced subgraph on $S \cup \Gamma(S)$ is connected, then S is an elementary trapping set (ETS) of type (a, b), where

$$a := |S|, \qquad b := |\partial S|.$$

If, in addition, S can be written as a union of simple 8-cycles such that each new cycle shares at least one variable node or check node with the previous union, then S is called a *cycle-8-connected ETS*. □

The point of this definition is that for girth 8, the smallest local cyclic building blocks are exactly 8-cycles. Restricting to cycle-8-connected ETSs narrows the search space to a family of local structures that actually appears in the codes studied here.

In the implementation we also monitor, as a separate branch, the three explicit ETS library families used in [15]. They are the $(6, 2)$ ETS obtained by gluing two 8-cycles, the $(12, 2)$ ETS obtained by chaining five 8-cycles, and the $(8, 2)$ path-4 family obtained by augmenting a $(6, 2)$ core with a length-4 Tanner path. Beyond these named patterns, we explicitly enumerate the chain family obtained by connecting $d = 2, \ldots, 10$ 8-cycles in a line. Since the cases $d = 2$ and $d = 5$ reproduce the $(6, 2)$ and $(12, 2)$ families, the additional branch tracks the non-duplicate lengths $d = 3, 4, 6, 7, 8, 9, 10$. We also extend the path-4 construction beyond the $(6, 2)$ core in the right panel by attaching a Tanner path of length 4 to the odd-check pair of these longer chain cores. Figure 1 gives schematic pictures of these three families. Circles denote variable nodes, light gray squares denote check nodes, and dark gray squares denote the odd-check boundary ∂S. The figure is schematic rather than metric; its purpose is to display the local topological pattern that the method searches for. In addition, the generic branch of the method enumerates connected unions made of between two and ten 8-cycles. Thus the present ETS upper bound is not tied to a single hand-crafted family: it combines bounded generic union search with the explicit library patterns of Figure 1, their chain-extension, and their path-4 augmentation.

Lemma 7.6 (The odd-check boundary equals the induced syndrome)**.** For any $S \subseteq V(H)$, let $\mathbf{1}_S \in \mathbb{F}_2^n$ be its indicator vector. Then

$$H\,\mathbf{1}_S^{\mathsf{T}} = \mathbf{1}_{\partial S}^{\mathsf{T}}.$$

In particular, if S is an ETS of type (a, b), then the induced syndrome has weight exactly b. □

Proof. Each syndrome component is the parity of the number of neighbors of the corresponding check node inside S, namely the parity of $d_S(c)$. This is 1 exactly when $c \in \partial S$. □

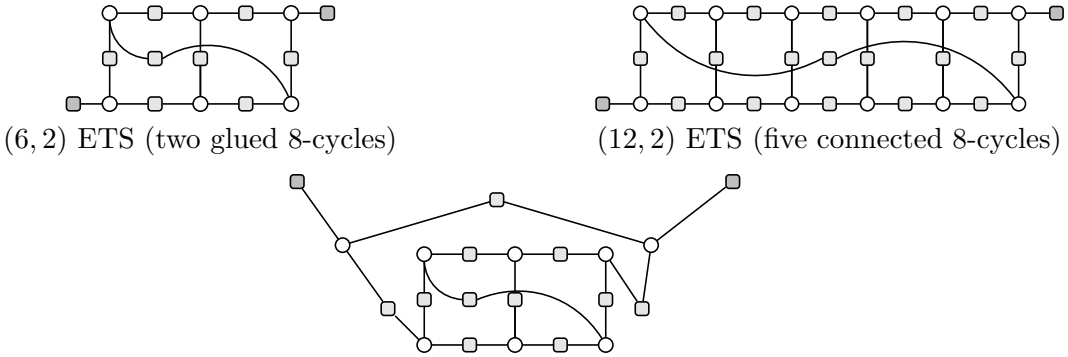

Figure 1: Schematic pictures of the ETS library families explicitly used in [15]. Left: a $(6, 2)$ ETS built from two glued 8-cycles. Middle: a $(12, 2)$ ETS built from a chain of five 8-cycles. Right: the $(8, 2)$ path-4 family obtained by augmenting a $(6, 2)$ core with a length-4 Tanner path. In the present upper-bound method, these families, together with their chain-extension for $d = 2, \ldots, 10$ and the corresponding path-4 augmentations, are counted in parallel with the generic cycle-8-connected ETS search, and are then passed through the same boundary-pairing and row-space tests to produce CSS witness candidates.

This lemma identifies the graph-theoretic boundary with a linear-algebraic syndrome. That is why $(a, 0)$ ETSs and differences of $(a, 2)$ ETSs can become explicit codeword candidates. This is also why the present method focuses on small boundary size b. For $b = 0$, the support is already syndrome-zero, and for $b = 2$, two ETSs with the same boundary cancel by a single symmetric difference. For larger b, one would need to solve a more complicated boundary-cancellation problem among several ETSs, which makes the search space much larger and is left outside the present UB^4 method.

Proposition 7.7 (Classical upper bounds from cycle-8-connected ETSs with $b = 0$)**.** If S is a cycle-8-connected ETS of type $(a, 0)$, then $\mathbf{1}_S \in \mathrm{Ker}(H)$, hence the corresponding classical code $C(H) = \mathrm{Ker}(H)$ satisfies $d(C(H)) \leq a$. □

Proof. By Lemma 7.6, $\partial S = \emptyset$ implies $H\,\mathbf{1}_S^{\mathsf{T}} = 0$. □

Proposition 7.8 (Classical upper bounds from pairs of $(a, 2)$ ETSs with the same boundary)**.** Let S_1, S_2 be cycle-8-connected ETSs of types $(a_1, 2)$ and $(a_2, 2)$, respectively. If $\partial S_1 = \partial S_2$ and $S_1 \triangle S_2 \neq \emptyset$, then $\mathbf{1}_{S_1} + \mathbf{1}_{S_2} = \mathbf{1}_{S_1 \triangle S_2} \in \mathrm{Ker}(H)$, and therefore $d(C(H)) \leq |S_1 \triangle S_2| \leq a_1 + a_2$. □

Proof. Lemma 7.6 gives $H\,\mathbf{1}_{S_i}^{\mathsf{T}} = \mathbf{1}_{\partial S_i}^{\mathsf{T}}$ for $i = 1, 2$. Since $\partial S_1 = \partial S_2$, adding the two equations over $\mathbb{F}_2$ yields $H(\mathbf{1}_{S_1} + \mathbf{1}_{S_2})^{\mathsf{T}} = 0$. The sum of indicator vectors equals the indicator vector of the symmetric difference. The assumption $S_1 \triangle S_2 \neq \emptyset$ guarantees that this codeword is nonzero. □

This proposition shows that a local structure with a nonzero syndrome can still produce a codeword once it is paired with another ETS sharing the same odd-check boundary.

Corollary 7.9 (CSS upper bounds from cycle-8-connected ETS witnesses)**.** Let $H = H_Z$, and let $\boldsymbol{x} \in \mathrm{Ker}(H_Z)$ be a nonzero support vector obtained from Proposition 7.7 or Proposition 7.8. If $\boldsymbol{x} \notin \mathrm{Row}(H_X)$, then $\boldsymbol{x}$ is a valid X logical representative and $d_X \leq \mathrm{wt}(\boldsymbol{x})$. Similarly, if $\boldsymbol{z} \in \mathrm{Ker}(H_X) \setminus \mathrm{Row}(H_Z)$ is a nonzero support vector obtained on the Z side, then $d_Z \leq \mathrm{wt}(\boldsymbol{z})$. □

Proof. This is immediate from the CSS distance definitions together with $\mathrm{Row}(H_X) = C_X^{\perp}$ and $\mathrm{Row}(H_Z) = C_Z^{\perp}$. □

The corollary explains why local Tanner-graph structure can be turned into quantum-distance witnesses by a single row-space test against the opposite stabilizer.

Enumeration algorithm Based on the above theory, the cycle-8 ETS enumerator used in our computations first enumerates all simple 8-cycles in the Tanner graph, builds their adjacency graph, grows connected unions up to a fixed budget, records $(a, 0)$ ETSs and $(a, 2)$ ETSs grouped by boundary, and finally applies Corollary 7.9 to extract classical and CSS upper-bound witnesses from nonzero supports. The algorithm is not used to optimize the decoder; it is used to mine local structural witnesses. Implementation details are included in the supplementary materials [17]. In the computations reported here, the generic branch is run with

$$2 \le t \le 10,$$

so that the method covers local structures ranging from small glued pairs of 8-cycles to visibly more composite cycle unions. The outputs of this generic branch are then merged with the three explicit library families shown in Figure 1.

7.4 Decoder-Failure Experiments

Independent upper bounds can also be extracted from decoder failures of the decoder used in our experiments. Concretely, for an estimated error $\hat{\mathbf{e}}$ whose syndrome agrees with the observed syndrome, we form the residual

$$\boldsymbol{\Delta} = \mathbf{e} + \hat{\mathbf{e}}.$$

If this residual is a pure-X or pure-Z vector that lies outside the opposite stabilizer row space, then its weight gives an upper bound on the corresponding logical distance. Such a witness need not reduce to a latent witness, a block-compression witness, or a cycle-8 ETS witness, so it serves as an independent method that complements the structural screening. The same syndrome-residual viewpoint already appears in the decoder-side analysis of [15]; here we reinterpret it explicitly as a minimum-distance upper-bound method.

Proposition 7.10. Let $\mathbf{e}$ be the true error and $\hat{\mathbf{e}}$ an estimated error with the same observed syndrome, and set the residual to $\boldsymbol{\Delta} = \mathbf{e} + \hat{\mathbf{e}}$. If $\boldsymbol{\Delta}$ is pure X-type and $\boldsymbol{\Delta} \notin \mathrm{Row}(H_X)$, then $d_X \le \mathrm{wt}(\boldsymbol{\Delta})$. Likewise, if $\boldsymbol{\Delta}$ is pure Z-type and $\boldsymbol{\Delta} \notin \mathrm{Row}(H_Z)$, then $d_Z \le \mathrm{wt}(\boldsymbol{\Delta})$. □

Proof. Since $\mathbf{e}$ and $\hat{\mathbf{e}}$ produce the same syndrome, the residual $\boldsymbol{\Delta}$ is syndrome-zero. Hence, if it is pure X-type then $\boldsymbol{\Delta} \in \mathrm{Ker}(H_Z)$, and if it is pure Z-type then $\boldsymbol{\Delta} \in \mathrm{Ker}(H_X)$. By assumption it lies outside the corresponding stabilizer row space, so it is a nontrivial logical representative and its weight upper-bounds the corresponding logical distance. □

Decoder-failure experiments were performed at the fixed channel parameter $p = 0.03$ against the non-daggered current-best row for each block length. The daggered high-P follow-up candidates have only the preliminary screening reported in Table 2. The trial counts quoted below are extracted from completed experimental logs archived in the supplementary materials [17]. For every non-daggered row tracked in the present manuscript, the decoder-failure experiments already amount to at least ten million decoding attempts. On the short-length side this count combines archived scans and later additional runs, while for $P \ge 264$ the completed counts alone already reach 1.28×10^8, 4.48×10^8, and 8.40×10^7 trials for representative rows. Including the April 15, 2026 top-per-P screening update, the large-P $p = 0.03$ campaign ledger contains 82.046×10^6, 84.448×10^6, and 71.698×10^6 completed or accounted trials for $P = 1536, 1920, 2688$, respectively, while the $P = 3072$ and $P = 3840$ reported rows remain at 64.000×10^6 completed trials each. The two new $P = 1920$ fail logs are syndrome-mismatch logs whose coset-completion checks produced no logical representative, so the UB^6 entries in Table 2 are unchanged.

In the present paper, the decoder-failure experiments are treated as a simulation-based method that complements the structural upper-bound methods. Detailed fail logs and individual witnesses are deferred to the supplementary materials [17]; the main text reports only the resulting bounds reflected in Table 2.

Table 1: Correspondence between UB^i and the upper-bound methods

symbol	method	proposition/corollary
UB^1	latent upper bound	Proposition 5.2
UB^2	block-compression upper bound	Proposition 6.6
UB^3	CRT-compression upper bound	Proposition 7.2
UB^4	cycle-8 ETS upper bound	Corollary 7.9
UB^5	direct CSS-search upper bound	Proposition 7.4
UB^6	decoder-failure upper bound	Proposition 7.10

8 Best-by-P Results

In this section, we first re-evaluate the published code of [15] using the new upper-bound theory developed in this paper, and then present the best-by-P ladder obtained by searching for constructions that make these upper bounds as large as possible. Each row listed below is not an intermediate search record, but the current best row adopted after applying the implemented upper-bound methods to the candidates obtained within a fixed search budget. For the exact $P = 768$ code listed in Table 1 of [15], the latent upper bound remains 48, but the block-compression upper bound is 32. Therefore the overall structural upper bound is $d \leq 32$, which is sharper than the earlier organization $d \leq 48$. This re-evaluation shows that the new upper bounds are not only useful for new searches but also materially sharpen the analysis of an already published code.

We then run the search so as to maximize the minimum of the latent, block-compression, cycle-8 ETS, and decoder-failure upper bounds, and retain one current best code for each lift size P. The resulting collection of current best rows, one for each reported lift size, is discussed below. Each row has been checked independently for CSS orthogonality and girth 8. The point is not that one obstruction is made large in isolation, but that the reported rows remain strong after every currently available method has been run against them.

Table 2 is not merely a list of examples. It is a table of current best codes obtained by running the full search-and-certification loop for each reported lift size. The rows should therefore be read as a benchmark summary of where the current construction stands within one fixed search framework. Moreover, each row is optimized against the minimum of multiple independent methods rather than against a single proxy. The only exception is the row 768^*, which is not a current best row but a reference row obtained by re-evaluating the exact code printed in Table 1 of [15]. The daggered high-P rows are not reference rows; they are newly found follow-up candidates retained in the table while their decoder-failure screening is still incomplete.

For convenience, Table 1 summarizes the correspondence between UB^1–UB^6 and the upper-bound methods used in the main table.

The reported lift sizes should therefore be read as a non-prime-power search family rather than as an arbitrary list of moduli. As explained in the companion commutation-pattern note [16], prime-power moduli are excluded by the affine commutation obstruction, whereas CRT-split moduli provide the natural existence side. The rows of Table 2 record the current best codes found in that non-prime-power regime, together with daggered high-P follow-up candidates. The labels $3072^{\dagger 1}$–$3072^{\dagger 4}$ correspond to seeds 3072324102, 3072444078, 3072476086, and 3072508120, respectively. The labels $3840^{\dagger 1}$ and $3840^{\dagger 2}$ correspond to seeds 3840356020 and 3840428002. For these daggered rows, the UB^6 entries remain unset because the current $p = 0.03$ screening has found no decoder-failure witness after several million trials per row.

Table 2 and Fig. 2 show two different tendencies. The structural bounds traced by UB^1–UB^5 largely rise with blocklength, but the overall upper bound can still drop sharply when a decoder-failure witness is found. The reference row illustrates this effect, whereas the updated high-P endpoint of the blue polyline uses the daggered $P = 3840$ follow-up candidate with overall upper bound 256.

Table 2: Current best upper bounds for the reported lift sizes. Shaded cells attain the reported value $d \le \mathrm{ub}$. The row 768^* is the only reference row; it is obtained by re-evaluating the exact printed code of Table 1 in [15] and is not counted among the reported winners. Daggered rows are high-P follow-up candidates; they are not used in Fig. 2. In the UB^4 column, NF means that the budgeted cycle-8 ETS rerun finished on both sides with no CSS witness found, whereas NE means that the row was not evaluated under that protocol or that the rerun remained incomplete. No UB^4 entry should be read as an exhaustive proof of nonexistence.

P	n	k	d	UB^1	UB^2	UB^3	UB^4	UB^5	UB^6
216	2592	1300	≤ 14	24	36	54	NE	54	14
240	2880	1444	≤ 24	24	40	40	NE	40	24
264	3168	1588	≤ 22	24	44	44	NE	44	22
288	3456	1732	≤ 24	24	24	64	NF	32	28
384	4608	2308	≤ 24	24	48	204	NE	128	28
576	6912	3460	≤ 48	48	64	72	NF	72	–
768	9216	4612	≤ 48	48	108	222	NF	74	–
768^*	9216	4612	≤ 32	48	32	96	NE	128	–
1536	18432	9220	≤ 48	48	256	978	NF	512	–
1920	23040	11524	≤ 64	120	64	96	NF	96	–
2688	32256	16132	≤ 128	336	128	224	NE	224	–
3072	36864	18436	≤ 96	96	192	2048	NF	640	–
$3072^{\dagger 1}$	36864	18436	≤ 192	192	192	–	NE	–	–
$3072^{\dagger 2}$	36864	18436	≤ 192	192	192	–	NE	–	–
$3072^{\dagger 3}$	36864	18436	≤ 192	192	192	–	NE	–	–
$3072^{\dagger 4}$	36864	18436	≤ 192	192	192	–	NE	–	–
3840	46080	23044	≤ 128	480	128	240	NF	240	–
$3840^{\dagger 1}$	46080	23044	≤ 256	480	256	–	NE	–	–
$3840^{\dagger 2}$	46080	23044	≤ 192	192	256	–	NE	–	–

Several regimes can be distinguished. At $P = 216$, UB^1 reaches 24, but the overall upper bound is reduced to 14 by a UB^6 witness. At $P = 240$, the reported winner has $\mathsf{UB}^1 = 24$ and a matching screened UB^6 value 24. At $P = 264$, the clean rerun produces the structural upper bound 24, but UB^6 reduces it to 22. At $P = 288$, the reported winner is still bounded by UB^1 and UB^2 at 24; the screened row 288^* with $\mathsf{UB}^6 = 34$ is retained only as a comparison row. For the rows with $P = 384, 576, 768, 1536$, and for the current daggered $P = 3072$ rows, UB^1 determines or ties the overall upper bound. For the rows with $P = 1920$, 2688, and for the current daggered $P = 3840$ winner, the overall upper bound is determined by UB^2. This is why one must keep monitoring UB^1, UB^2, UB^3, UB^4, UB^5, and UB^6 simultaneously.

For UB^4, Table 2 now distinguishes two nonnumeric outcomes. NF means that the budgeted cycle-8 ETS rerun completed on both sides and still produced no CSS witness, whereas NE means that the row was not evaluated under that rerun protocol or that one side remained incomplete. In particular, no NF entry should be interpreted as an exhaustive proof that no cycle-8 ETS witness exists.

In particular, the current $P = 3840$ best row

$$[\![46080, 23044, \le 256]\!]$$

has $\mathsf{UB}^1 = 480$, but the overall upper bound is determined by UB^2 at 256.

Exact latent certification for the reported codes For the ten codes

$$P = 240, 288, 384, 576, 768, 1536, 1920, 2688, 3072, 3840$$

among the reported rows, we reran the latent certification and confirmed that UB^1 coincides with the exact latent lower bound. For each code we construct Ψ_3 and choose the candidate

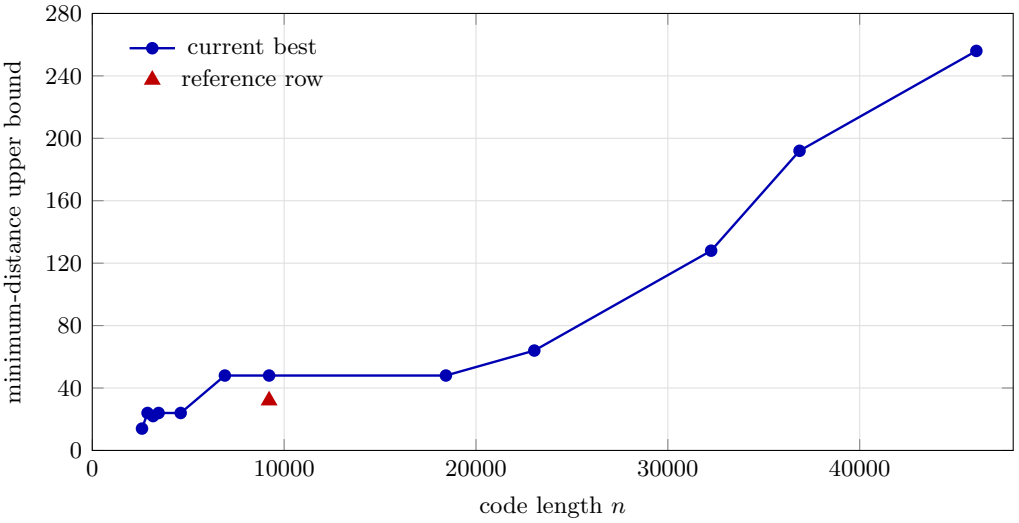

Figure 2: Current best upper bounds as a function of the code length n. The blue polyline traces the best available row found in the present search for each P, including the daggered high-P candidates when they improve the record, while the red triangle marks the non-winner reference row listed in Table 2.

block factor

$$m = P/\dim \mathrm{Ker}(\Psi_3).$$

For these ten codes this gives

$$m = 2, 2, 2, 4, 4, 4, 10, 28, 8, 40,$$

and therefore

$$Q = P/m = 120, 144, 192, 144, 192, 384, 192, 96, 384, 96.$$

We then verify the two rank equalities of Corollary A.4, which shows that the whole kernels of Ψ_3 and Ψ_3^{T} are blockwise m-block constant for every one of these rows.

Next, write the latent witness weights given by Proposition 5.2 as $d_X^{(\mathrm{lat}),\mathrm{ub}}$ and $d_Z^{(\mathrm{lat}),\mathrm{ub}}$. For the ten codes above one has

$$\frac{d_X^{(\mathrm{lat}),\mathrm{ub}}}{m} = \frac{d_Z^{(\mathrm{lat}),\mathrm{ub}}}{m} = 12,$$

so Theorem A.2 reduces the problem to proving that the compressed latent image contains no nonzero word of weight at most 11 on either side. This UNSAT check is automated by a dedicated script and succeeds on both the X and Z sides for all ten non-daggered rows. Implementation details and certification logs are archived in the supplementary materials [17]. Hence, for each of these ten codes,

$$d_X^{(\mathrm{lat})} = d_X^{(\mathrm{lat}),\mathrm{ub}}, \qquad d_Z^{(\mathrm{lat})} = d_Z^{(\mathrm{lat}),\mathrm{ub}}.$$

The certified latent values

$$24,\ 24,\ 24,\ 48,\ 48,\ 48,\ 120,\ 336,\ 96,\ 480$$

correspond to the ten non-daggered rows in the order listed above.

Thus the reported best-by-P ladder is not merely an upper-bound ladder. At least for these ten codes, it also contains a machine-checkable certified latent ladder. The overall upper bound can nevertheless be governed by UB^2 or UB^6. In particular, the non-daggered $P = 3840$ reported row

$$[\![46080, 23044, \leq 128]\!]$$

has

$$d_X^{(\mathrm{lat})} = d_Z^{(\mathrm{lat})} = 480,$$

whereas UB^2 is 128. Likewise, for the reported $P = 2688$ winner, the certified latent value is 336, whereas UB^2 gives the smaller overall upper bound 128.

The exact latent certification is fully automated. The script chooses the candidate block factor, compresses the latent image, runs low-weight exclusion through satisfiability (SAT) and satisfiability modulo theories (SMT) backends, and checks the rank identities of Corollary A.4. Measured runtimes for the ten certified rows are also fixed in the supplementary materials [17].

The complexity profile is also fairly transparent. On the linear-algebra side one computes the $\mathbb{F}_2$ row ranks of Ψ_3 and

$$\begin{bmatrix} \Psi_3 \\ B_m \end{bmatrix}, \qquad \begin{bmatrix} \Psi_3^{\mathsf{T}} \\ B_m \end{bmatrix},$$

which costs $O(P^3)$ field operations under naive Gaussian elimination, or roughly $O(P^3/w)$ bit operations in a bit-packed implementation. On the exact latent exclusion side, the number of source bits is $3Q$ and the number of output bits is $12Q$ on each side, and every output bit is represented as the XOR of three source bits. Hence the constraint size itself is linear in Q, although the SAT/SMT search for excluding weight at most 11 is exponential in the worst case. Because the threshold is fixed at 11 for all reported rows, the practical difficulty is governed mainly by the quotient length Q. In the actual runs, the cases with $Q = 96, 120, 144, 192$ are relatively light, whereas the cases with $Q = 384$ are the heaviest.

9 Discussion and Outlook

From the standpoint of distance evaluation, this paper is a compromise. What one would ultimately like is a rigorous lower bound for the full CSS distance. Appendix A provides such certification only for the latent part. For the overall distance, the best-known lower-bound techniques in the quantum-LDPC literature are tailored to product-type or expansion-type families [9, 10, 11], whereas the present APM-LDPC family is governed by an active/latent decomposition and affine-permutation structure. This mismatch is why the present work settles for the sharpest upper bounds we can currently certify, rather than a full lower-bound theorem.

The same caution applies to the classical minimum-weight algorithms that are often practical for short linear codes [19, 18, 20, 21]. Those algorithms target the minimum weight of one classical code, whereas the present problem asks for a representative outside the relevant stabilizer row space and, in addition, benefits from separating latent and non-latent mechanisms. This is precisely why the present paper organizes several specialized upper-bound methods instead of importing one classical minimum-distance routine wholesale.

The most useful way to read the present results is therefore as follows. The construction method of [15] could plausibly have minimum distance that grows with blocklength, perhaps linearly, and the natural way to challenge that possibility is to search for explicit low-weight logical operators. This paper does so through several independent upper-bound methods. The outcome is genuinely mixed. On one hand, UB^1–UB^5 continue to display a broadly linear structural scale over the reported range. On the other hand, UB^6 can behave very differently: it already dominates the overall bound at several small lift sizes and has produced very low working-log candidates at $P = 2688$, even though the current-best $P = 2688$ row is still governed by UB^2.

The asymptotic question nevertheless remains open. Decoder-failure witnesses are potentially the most direct way to produce unexpectedly small logical operators, but once the relevant residual weights move into the range of a few dozen qubits, the required computational effort rises sharply, and many larger-size searches still return no witness in practical time. The present data therefore provide concrete contrary evidence against a naive linear-growth picture, while still not determining the eventual asymptotic behavior of the family.

A Exact Latent Lower Bounds via Block-Constant Compression

UB^1 becomes exact only when one can prove a matching lower bound. The mechanism used here is block-constant compression.

Example A.1.

$$\pi_{m,L} : U_m(LP) \to \mathbb{F}_2^{LQ}, \qquad \iota_{m,L} : \mathbb{F}_2^{LQ} \to U_m(LP).$$

For the principal family $(J, L) = (3, 12)$ treated in this paper, these maps specialize to compression and lift between $U_m(12P)$ and $\mathbb{F}_2^{12Q}$. The theorem below states that, if the relevant mixed-product kernels are block constant, then every latent candidate descends to the compressed image. This is the step that turns the latent exactness problem into a finite exclusion problem in a shorter quotient space. □

Theorem A.2 (Exact latent lower bounds under block-constant kernel hypotheses)**.** Let $s := L/2 - J$ and assume $P = mQ$. Suppose

$$\mathrm{Ker}(H_Z \tilde{H}_X^{\mathsf{T}}) \subseteq U_m(sP), \qquad \mathrm{Ker}(H_X \tilde{H}_Z^{\mathsf{T}}) \subseteq U_m(sP).$$

In other words, every latent coefficient vector lying in the kernel of either active–latent mixed product is blockwise m-block constant. Define the compressed latent images

$$\bar{\mathcal{L}}_X := \pi_{m,L}(C_Z \cap \mathrm{Row}(\tilde{H}_X)), \qquad \bar{\mathcal{L}}_Z := \pi_{m,L}(C_X \cap \mathrm{Row}(\tilde{H}_Z)).$$

If every nonzero vector in $\bar{\mathcal{L}}_X$ has weight at least τ_X, and every nonzero vector in $\bar{\mathcal{L}}_Z$ has weight at least τ_Z, then

$$d_X^{(\mathrm{lat})} \geq m\tau_X, \qquad d_Z^{(\mathrm{lat})} \geq m\tau_Z.$$

□

Proof. It suffices to prove the X case. Let $\boldsymbol{x} \in C_Z \cap \mathrm{Row}(\tilde{H}_X)$. Then

$$\boldsymbol{x} = \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda}, \qquad \boldsymbol{\lambda} \in \mathbb{F}_2^{sP}.$$

Since $\boldsymbol{x} \in C_Z$,

$$0 = H_Z \boldsymbol{x} = H_Z \tilde{H}_X^{\mathsf{T}} \boldsymbol{\lambda},$$

so

$$\boldsymbol{\lambda} \in \mathrm{Ker}(H_Z \tilde{H}_X^{\mathsf{T}}) \subseteq U_m(sP).$$

Lemma 6.2 therefore implies that $\boldsymbol{x}$ is block constant blockwise, so $\boldsymbol{x} \in U_m(LP)$ and

$$\bar{\boldsymbol{x}} := \pi_{m,L}(\boldsymbol{x}) \in \bar{\mathcal{L}}_X$$

is well defined. Moreover,

$$\boldsymbol{x} = \iota_{m,L}(\bar{\boldsymbol{x}}).$$

If $\boldsymbol{x} \neq 0$, then $\bar{\boldsymbol{x}} \neq 0$ and

$$\mathrm{wt}(\boldsymbol{x}) = m\,\mathrm{wt}(\bar{\boldsymbol{x}}) \geq m\tau_X.$$

Every latent X-logical representative is a nonzero vector in $C_Z \cap \mathrm{Row}(\tilde{H}_X)$, so $d_X^{(\mathrm{lat})} \geq m\tau_X$. The Z case is identical with $\mathrm{Ker}(H_X \tilde{H}_Z^{\mathsf{T}}) \subseteq U_m(sP)$. □

This theorem is the lower-bound half of exact latent certification. Its point is that a low-weight exclusion proof in the quotient space becomes a rigorous lower bound in the original code once the kernel structure is known to be block constant.

To make the kernel hypothesis checkable, one replaces it by a rank computation.

Lemma A.3 (Rank test for block-constant kernels)**.** Let $\boldsymbol{e}_0, \dots, \boldsymbol{e}_{P-1}$ be the standard basis of $\mathbb{F}_2^P$. Define

$$B_m \in \mathbb{F}_2^{(m-1)Q \times P}$$

to be the matrix whose rows are

$$\boldsymbol{e}_t + \boldsymbol{e}_{t+jQ} \qquad (0 \le t \le Q-1,\ 1 \le j \le m-1).$$

Then

$$U_m(P) = \mathrm{Ker}(B_m).$$

Moreover, for every binary matrix $A \in \mathbb{F}_2^{r \times P}$,

$$\mathrm{Ker}(A) \subseteq U_m(P) \iff \mathrm{Row}(B_m) \subseteq \mathrm{Row}(A) \iff \mathrm{rank}\begin{bmatrix} A \\ B_m \end{bmatrix} = \mathrm{rank}(A).$$

□

Proof. The defining equations of $U_m(P)$ are the equalities $x_t = x_{t+jQ}$, equivalently

$$(\boldsymbol{e}_t + \boldsymbol{e}_{t+jQ})\boldsymbol{x}^{\mathsf{T}} = 0.$$

Hence $U_m(P) = \mathrm{Ker}(B_m)$. Over $\mathbb{F}_2$ one has

$$(\mathrm{Ker}(A))^{\perp} = \mathrm{Row}(A),$$

so

$$\mathrm{Ker}(A) \subseteq \mathrm{Ker}(B_m) \iff \mathrm{Row}(B_m) \subseteq \mathrm{Row}(A).$$

The row-space inclusion is equivalent to the displayed rank equality. □

The lemma turns a combinatorial-looking kernel condition into a linear-algebraic condition that can be logged and audited directly. This conversion from kernel structure to rank checks is precisely what makes machine-checkable latent certification possible.

Corollary A.4 (Machine-readable form of the latent kernel hypothesis)**.** Let $s := L/2 - J$ and define

$$B_m^{(s)} := I_s \otimes B_m \in \mathbb{F}_2^{s(m-1)Q \times sP}.$$

Then the kernel hypothesis of Theorem A.2 is equivalent to the two GF(2) rank equalities

$$\mathrm{rank}\begin{bmatrix} H_Z \tilde{H}_X^{\mathsf{T}} \\ B_m^{(s)} \end{bmatrix} = \mathrm{rank}(H_Z \tilde{H}_X^{\mathsf{T}}), \qquad \mathrm{rank}\begin{bmatrix} H_X \tilde{H}_Z^{\mathsf{T}} \\ B_m^{(s)} \end{bmatrix} = \mathrm{rank}(H_X \tilde{H}_Z^{\mathsf{T}}).$$

Therefore exact latent certification at block factor m can be audited by combining these two rank checks with an explicit latent witness and an UNSAT certificate for low-weight vectors in the compressed latent image. □

Example A.5. In the specialized $(J, L) = (3, 12)$ family with $\Psi_r = 0$ for $r \neq 3$, the mixed products reduce to

$$H_Z \tilde{H}_X^{\mathsf{T}} = \mathrm{diag}(\Psi_3^{\mathsf{T}}, \Psi_3^{\mathsf{T}}, \Psi_3^{\mathsf{T}}), \qquad H_X \tilde{H}_Z^{\mathsf{T}} = \mathrm{diag}(\Psi_3, \Psi_3, \Psi_3).$$

Thus Theorem A.2 and Corollary A.4 reduce to checking block-constant structure for the kernels of Ψ_3 and Ψ_3^{T}. This is the concrete form used in the computational exact-latent certificates of this paper. □

Example A.6. The exact $P = 768$ code printed in Table 1 of [15] also satisfies this block-constant kernel hypothesis. Running the current latent certification tool on the exact transcription archived in the supplementary materials [17] gives

$$\operatorname{rank}(\Psi_3) = 576, \qquad \dim \operatorname{Ker}(\Psi_3) = 192,$$

so the relevant block factor is $m = 4$ with $Q = 192$, and the kernels of both Ψ_3 and Ψ_3^{T} are 4-block constant. The SAT/SMT exclusion on the compressed latent image also succeeds, so for this published code one obtains the exact latent values

$$d_X^{(\mathrm{lat})} = d_Z^{(\mathrm{lat})} = 48.$$

Hence the improvement from $d \le 48$ to $d \le 32$ for the code of [15] does not come from sharpening UB^1; it comes from the separate 4-block-constant non-latent witness giving UB^2. □

This completes the latent exactness story. Once block-constant kernel structure is verified, the latent lower-bound problem is reduced to proving nonexistence of compressed latent vectors below a target weight threshold, typically by satisfiability (SAT) or satisfiability modulo theories (SMT).

B APM parameter tuples for the reported rows

Each reported code is specified by six affine permutations

$$f_i(x) = a_i x + b_i, \qquad g_i(x) = c_i x + d_i \qquad (i = 0, \dots, 5).$$

Table 3 records the six affine maps $f_i(x)$ and the six affine maps $g_i(x)$ for all rows reported in the main text, including the daggered candidate rows, in explicit linear form. The row 768^* is the exact transcription of the printed code of [15].

Table 3: APM coefficients for the reported rows, daggered candidate rows, and the $P = 768$ paper-code reference row. Each code is displayed in two rows: the first row lists $f_i(x)$, and the second row lists $g_i(x)$.

P	i	0	1	2	3	4	5
216	f_i	$181x + 105$	$91x + 57$	$37x + 192$	$181x + 186$	$91x + 57$	$37x + 156$
	g_i	$193x + 112$	$97x + 116$	$193x + 64$	$181x + 54$	$97x + 44$	$193x + 100$
240	f_i	$1x + 54$	$31x + 141$	$1x + 0$	$31x + 153$	$151x + 201$	$1x + 96$
	g_i	$161x + 112$	$121x + 148$	$121x + 180$	$221x + 162$	$121x + 164$	$41x + 44$
264	f_i	$67x + 126$	$1x + 177$	$1x + 126$	$133x + 198$	$1x + 261$	$1x + 144$
	g_i	$89x + 136$	$1x + 248$	$89x + 48$	$89x + 86$	$1x + 32$	$1x + 124$
288	f_i	$73x + 98$	$241x + 244$	$241x + 36$	$1x + 192$	$1x + 64$	$97x + 72$
	g_i	$109x + 75$	$217x + 186$	$1x + 120$	$217x + 0$	$217x + 126$	$37x + 141$
384	f_i	$181x + 282$	$145x + 8$	$233x + 180$	$169x + 84$	$337x + 296$	$89x + 108$
	g_i	$127x + 63$	$157x + 366$	$277x + 138$	$163x + 129$	$13x + 294$	$229x + 114$
576	f_i	$1x + 90$	$289x + 408$	$325x + 441$	$1x + 168$	$397x + 195$	$433x + 540$
	g_i	$1x + 48$	$289x + 392$	$97x + 72$	$409x + 134$	$97x + 328$	$385x + 272$
768	f_i	$557x + 626$	$161x + 624$	$385x + 704$	$737x + 592$	$721x + 760$	$49x + 264$
	g_i	$571x + 639$	$55x + 681$	$229x + 294$	$307x + 579$	$637x + 234$	$121x + 660$
768^*	f_i	$763x + 435$	$679x + 69$	$397x + 330$	$61x + 18$	$697x + 612$	$373x + 246$
	g_i	$289x + 496$	$257x + 640$	$625x + 200$	$41x + 524$	$193x + 672$	$449x + 672$
1536	f_i	$1069x + 1446$	$401x + 1160$	$1361x + 1128$	$433x + 24$	$641x + 1088$	$209x + 40$
	g_i	$1039x + 111$	$1501x + 318$	$319x + 711$	$97x + 1488$	$97x + 816$	$985x + 12$
1920	f_i	$1261x + 141$	$1357x + 1101$	$1417x + 1806$	$1753x + 1482$	$1789x + 1473$	$1717x + 771$

P	i	0	1	2	3	4	5
	g_i	$161x + 440$	$481x + 840$	$161x + 1080$	$1817x + 826$	$721x + 460$	$961x + 80$
2688	f_i	$1681x + 759$	$1345x + 1050$	$1x + 2268$	$1x + 2541$	$1345x + 2625$	$1x + 336$
	g_i	$1793x + 1944$	$1x + 2040$	$1793x + 2408$	$1921x + 610$	$1x + 2640$	$1x + 2456$
3072	f_i	$2341x + 3034$	$3041x + 2992$	$929x + 272$	$2881x + 2080$	$1121x + 1776$	$2273x + 560$
	g_i	$2239x + 2907$	$319x + 411$	$1069x + 2670$	$697x + 108$	$1249x + 816$	$2353x + 2808$
$3072^{\dagger 1}$	f_i	$181x + 926$	$833x + 1504$	$641x + 3008$	$1x + 512$	$1025x + 1536$	$2305x + 384$
	g_i	$2167x + 1449$	$1861x + 1206$	$2779x + 1167$	$1699x + 1035$	$2041x + 2388$	$847x + 717$
$3072^{\dagger 2}$	f_i	$209x + 1096$	$1153x + 1088$	$2305x + 2176$	$1025x + 1024$	$385x + 2240$	$257x + 1664$
	g_i	$205x + 2862$	$1417x + 1524$	$337x + 2376$	$1675x + 957$	$193x + 2592$	$2053x + 1626$
$3072^{\dagger 3}$	f_i	$2689x + 1752$	$2561x + 3008$	$1x + 2048$	$1537x + 2880$	$2561x + 1856$	$1x + 2048$
	g_i	$1249x + 738$	$1729x + 1260$	$865x + 2514$	$2137x + 1245$	$1537x + 1512$	$1729x + 1116$
$3072^{\dagger 4}$	f_i	$1481x + 1842$	$769x + 1216$	$2689x + 2720$	$2177x + 2592$	$385x + 1888$	$1025x + 512$
	g_i	$1357x + 2547$	$2065x + 1284$	$1381x + 1977$	$2773x + 2829$	$301x + 1131$	$2161x + 156$
3840	f_i	$2431x + 393$	$3001x + 2100$	$3001x + 1620$	$1981x + 1290$	$841x + 780$	$661x + 3150$
	g_i	$1601x + 96$	$481x + 144$	$2081x + 112$	$313x + 3508$	$1921x + 3264$	$1x + 1280$
$3840^{\dagger 1}$	f_i	$3113x + 354$	$2881x + 272$	$1601x + 3408$	$3521x + 2480$	$1x + 2816$	$2881x + 2256$
	g_i	$2461x + 1515$	$661x + 465$	$961x + 2160$	$3241x + 1686$	$241x + 1260$	$3481x + 270$
$3840^{\dagger 2}$	f_i	$3649x + 549$	$1537x + 1872$	$3073x + 2400$	$769x + 2976$	$1x + 1872$	$3073x + 3456$
	g_i	$1921x + 1870$	$3521x + 1175$	$1921x + 2450$	$1361x + 1135$	$1601x + 2285$	$641x + 1650$

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963.

[2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[3] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, 2004.

[4] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, "Quasi-cyclic LDPC codes based on pre-lifted protographs," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5856–5874, 2014.

[5] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, pp. 1098–1105, 1996.

[6] A. M. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[7] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, 2004.

[8] M. Hagiwara and H. Imai, "Quantum quasi-cyclic LDPC codes," in *IEEE International Symposium on Information Theory*, 2007, pp. 806–810.

[9] J.-P. Tillich and G. Zemor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2014.

[10] P. Panteleev and G. Kalachev, "Asymptotically good quantum and locally testable classical LDPC codes," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 375–388.

[11] A. Leverrier and G. Zemor, "Quantum tanner codes," in *Proceedings of the 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science*, 2022, pp. 872–883.

[12] N. Myung, H. Yang, and J. Park, "A combining method of structured LDPC codes from affine permutation matrices," in *IEEE International Symposium on Information Theory*, 2006, pp. 747–751.

[13] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Quantum error correction beyond the bounded distance decoding limit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1223–1230, 2012.

[14] Y. Komoto and K. Kasai, "Quantum error correction near the coding theoretical bound," *npj Quantum Information*, vol. 11, p. 154, 2025.

[15] K. Kasai, "Breaking the orthogonality barrier in quantum LDPC codes," 2026, arXiv:2601.08824.

[16] ——, "Cross-commuting nonabelian squares in affine groups over finite commutative principal ideal rings," 2026, arXiv preprint.

[17] https://kasaikenta.github.io/.

[18] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, ser. Lecture Notes in Computer Science, vol. 388. Springer, 1989, pp. 106–113.

[19] J. S. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1354–1359, 1988.

[20] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 367–378, 1998.

[21] P. Lisoněk and L. Trummer, "Algorithms for the minimum weight of linear codes," *Advances in Mathematics of Communications*, vol. 10, no. 1, pp. 195–207, 2016.

[22] M. Grassl, "Searching for linear codes with large minimum distance," in *Discovering Mathematics with Magma*, ser. Algorithms and Computation in Mathematics, W. Bosma and J. Cannon, Eds. Springer, 2006, vol. 19, pp. 287–313.

[23] ——, "Bounds on the minimum distance of linear codes and quantum codes," 2007, [Online]. Available: https://www.codetables.de. Accessed: Apr. 2, 2026.

[24] M. Karimi and A. H. Banihashemi, "An efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Transactions on Information Theory*, vol. 58, no. 11, pp. 6942–6958, 2012.

[25] S. H. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6713–6736, 2016.