A Best-by-Lift Ladder of Girth-8 $(3, 12)$-Regular Binary CSS LDPC Codes with Certified Latent Distance and m-Block Non-Latent Screening

Kenta Kasai

Abstract

We present a table-first study of binary CSS LDPC codes obtained from affine permutation matrices (APMs). Instead of emphasizing a single showcase code, we optimize over the lift size P and report, for each P in our search set, the code with the largest structural minimum-distance upper bound. The resulting family forms a length ladder of girth-8, $(3, 12)$-regular binary CSS LDPC codes ranging from $[\![2592, 1300, \leq 24]\!]$ to $[\![46080, 23044, \leq 128]\!]$. The central artifact is a best-by-lift table: one code per P, each accompanied by structural distance certificates. The main technical point is a three-part certificate pipeline. First, latent logical operators are upper-bounded by explicit vectors obtained from Ψ_3-kernels. Second, when the same kernels are block-constant, we compress the latent problem and certify latent lower bounds exactly by satisfiability checks on the compressed image. Third, non-latent logical operators are upper-bounded by a generalized m-block-constant compression-and-lift argument over all divisors $m \mid P$. These ingredients convert the search problem into a reproducible best-by-lift table with rigorous latent-distance control and explicit non-latent screening. Our strongest current upper-bound example is a girth-8 binary CSS LDPC code with parameters $[\![46080, 23044, \leq 128]\!]$.

1 Introduction

Classical LDPC design benefits strongly from sparse regular Tanner graphs with large girth. In the quantum CSS setting, however, one must also enforce

$$H_X H_Z^{\mathsf{T}} = 0,$$

and this orthogonality constraint often destroys the very structures that make classical LDPC codes attractive. In particular, when one starts from a fully orthogonal parent pair and deletes rows to obtain a positive-rate CSS code, the deleted rows may become low-weight logical operators. This phenomenon imposes structural upper bounds on the minimum distance even before decoding enters the discussion.

Our recent work attacked this "orthogonality barrier" by enforcing orthogonality only on the active block rows and by allowing controlled non-commutativity in the latent part. That point of view produced individual girth-8 binary CSS LDPC examples. The present paper shifts the emphasis from a single example to a reproducible *best-by-lift ladder*: for each lift size P, we search for the code with the largest structural upper bound on the minimum distance and collect these winners in one table.

The paper is organized around three technical components.

- We derive latent upper bounds from the kernel of the interaction matrix Ψ_3.
- We prove latent lower bounds by compressing block-constant kernels and solving exact low-weight exclusion problems on the compressed image.

- We derive non-latent upper bounds from a generalized m-block compression, where the compression factor m is optimized over all divisors of P.

Together, these ingredients yield a practical search-and-certification loop: construct an APM code, verify CSS orthogonality and girth 8, compute latent and non-latent bounds, and keep the best code for each P.

The resulting table gives a more durable message than a one-off example. It exposes how the best achievable structural bound changes with P and clarifies when the latent obstruction dominates and when the non-latent obstruction takes over. The updated ladder shows that the dominant bottleneck can switch back to the non-latent side at larger lifts, so high-girth binary CSS LDPC design should track both ladders rather than optimize only one side in isolation.

2 Construction Framework

Let L be even, and let $F_0, \dots, F_{L/2-1}$ and $G_0, \dots, G_{L/2-1}$ be $P \times P$ permutation matrices. We consider the generalized Hagiwara–Imai parent matrices

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

with indices taken modulo $L/2$. The active matrices H_X, H_Z are the top J block rows, and the remaining $L/2 - J$ block rows are denoted by $\tilde{H}_X, \tilde{H}_Z$.

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

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

over $\mathbb{F}_2$. Then every block of $\hat{H}_X \hat{H}_Z^{\mathsf{T}}$ depends only on the row-difference and equals one of the Ψ_r.

For the standard active choice (top J block rows), let

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

The active orthogonality condition is localized as follows.

Theorem 2.1. *If $\Psi_r = 0$ for every $r \in \Delta$, then $H_X H_Z^{\mathsf{T}} = 0$.*

In this paper we specialize to

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

For the standard active choice, one obtains

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

so $r = 3$ is the unique difference not forced by active orthogonality. Accordingly, we search for commutation patterns satisfying

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

This makes Ψ_3 the sole latent interaction, and it is precisely this residual interaction that controls the latent-distance mechanism below.

We realize each block by an affine permutation matrix (APM)

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

The APM representation is important for two reasons. First, commutativity reduces to a single congruence relation

$$d(a-1) = b(c-1) \pmod{P},$$

which is easy to enforce during search. Second, when $P = mQ$, every APM descends naturally to the quotient modulo Q, which makes the later compression arguments exact.

3 Latent-Distance Theory

3.1 Latent upper bounds

The CSS distances are

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

We isolate the latent contribution by

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

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

These quantities upper-bound d_X, d_Z, and hence upper-bound $d_{\min}$.

When $\Psi_r = 0$ for $r \neq 3$ and $(J, L) = (3, 12)$, the mixed active-latent products simplify to

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

Hence the latent kernel splits blockwise.

Proposition 3.1 (Latent upper bound from Ψ_3)**.** *Assume $(J, L) = (3, 12)$ and $\Psi_r = 0$ for $r \neq 3$. For any nonzero $u \in \mathrm{Ker}(\Psi_3^{\mathsf{T}})$, place u in one latent row-block and zeros in the other two blocks, and define*

$$x = u^{\mathsf{T}} \tilde{H}_X.$$

If $x \notin C_X^{\perp}$, then $x \in C_Z \setminus C_X^{\perp}$ and

$$d_X^{(\mathrm{lat})} \leq \mathrm{wt}(x).$$

Similarly, for any nonzero $v \in \mathrm{Ker}(\Psi_3)$, the lift $z = v^{\mathsf{T}} \tilde{H}_Z$ gives

$$d_Z^{(\mathrm{lat})} \leq \mathrm{wt}(z)$$

provided $z \notin C_Z^{\perp}$.

This is the source of the latent upper bounds written into our code files. Operationally, we enumerate or sample low-weight kernel vectors of Ψ_3 and Ψ_3^{T}, lift them through $\tilde{H}_X, \tilde{H}_Z$, and keep the lightest valid representatives.

3.2 Latent lower bounds by block-constant compression

The latent upper bound becomes exact when the same kernel has a compatible block-constant structure. Suppose that

$$P = mQ$$

for some divisor $m \geq 2$. For each $t \in \mathbb{Z}_Q$, define the m-point coset

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

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

Let $U_m(P)$ denote the m-block-constant subspace of $\mathbb{F}_2^P$, and let

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

be the natural compression and lift maps. Extending blockwise over the $L = 12$ column blocks gives

$$\pi_{m,L} : U_m(12P) \to \mathbb{F}_2^{12Q}, \qquad \iota_{m,L} : \mathbb{F}_2^{12Q} \to U_m(12P),$$

with the exact weight scaling

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

Theorem 3.2 (Exact latent lower bound in the block-constant case)**.** *Assume* $(J, L) = (3, 12)$*,* $\Psi_r = 0$ *for* $r \neq 3$*, and* $P = mQ$*. Assume further that*

$$\mathrm{Ker}(\Psi_3) = U_m(P) \cap \mathrm{Ker}(\Psi_3), \qquad \mathrm{Ker}(\Psi_3^{\mathsf{T}}) = U_m(P) \cap \mathrm{Ker}(\Psi_3^{\mathsf{T}}),$$

that is, every kernel vector is m*-block constant. Let* $\bar{\mathcal{L}}_X \subset \mathbb{F}_2^{12Q}$ *and* $\bar{\mathcal{L}}_Z \subset \mathbb{F}_2^{12Q}$ *be the compressed latent images induced by* $\tilde{H}_X$ *and* $\tilde{H}_Z$*. If every nonzero vector of* $\bar{\mathcal{L}}_X$ *has weight at least* τ_X *and every nonzero vector of* $\bar{\mathcal{L}}_Z$ *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. Every latent X-candidate is the lift of a compressed vector in $\bar{\mathcal{L}}_X$ by the block-constant assumption. The lift multiplies weight by m, so any nonzero latent X-candidate has weight at least $m\tau_X$. The same argument applies to the Z side. □

The theorem reduces a length-$12P$ lower-bound statement to a length-$12Q$ exact exclusion problem. In our implementation, the exclusion problem is solved by satisfiability: given a target τ, we ask whether the compressed latent image contains a nonzero vector of weight at most $\tau - 1$. UNSAT certifies the lower bound. When Proposition 3.1 produces a matching weight-$m\tau$ witness, we conclude

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

4 Non-Latent Distance Upper Bounds

The latent mechanism does not exhaust all logical operators. We therefore also search for logical operators outside the latent row space. The resulting argument is one-sided: it gives explicit *upper bounds* on non-latent distance.

For any divisor $m \mid P$, define the m-block-constant subspace

$$U_m(12P) \subset \mathbb{F}_2^{12P}$$

as above. Because each APM descends modulo $Q = P/m$, the active matrices compress to

$$\bar{H}_X^{(m)}, \bar{H}_Z^{(m)} \in \mathbb{F}_2^{3Q \times 12Q}.$$

Lemma 4.1 (Compression equivalence for active checks)**.** *For every* $x \in U_m(12P)$ *and* $\bar{x} = \pi_{m,L}(x)$*,*

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

and similarly

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

for every $z \in U_m(12P)$*. Moreover,*

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

This leads to the following explicit upper-bound mechanism.

Proposition 4.2 (m-block non-latent upper bound)**.** *Let* $m \mid P$ *and* $Q = P/m$*. Suppose* $\bar{x} \in \mathrm{Ker}(\bar{H}_Z^{(m)})$ *and let* $x = \iota_{m,L}(\bar{x})$*. If*

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

then x *is a valid* X*-logical representative outside the latent row space, and*

$$d_X \leq \mathrm{wt}(x) = m\,\mathrm{wt}(\bar{x}).$$

Similarly, if $\bar{z} \in \mathrm{Ker}(\bar{H}_X^{(m)})$ *lifts to* $z \notin \mathrm{Row}(H_Z) \cup \mathrm{Row}(\tilde{H}_Z)$*, then*

$$d_Z \leq \mathrm{wt}(z) = m\,\mathrm{wt}(\bar{z}).$$

The proposition justifies a global scan over all divisors $m \mid P$. For each m, we solve a smaller kernel problem on the compressed checks and lift the best witness. Among all divisors, the smallest resulting witness gives our non-latent upper bound. This is exactly the "m-block bound" implemented in the current search code.

5 Search-and-Certification Pipeline

The theory above is embedded into a sequential APM search. Each block is an affine permutation

$$f_i(x) = a_i x + b_i, \qquad g_i(x) = c_i x + d_i \pmod{P},$$

with $\gcd(a_i, P) = \gcd(c_i, P) = 1$. The construction proceeds block by block. Once the multiplicative part a_i or c_j is chosen, the additive part is constrained by the commutation equation, so the search solves a sequence of congruences rather than brute-forcing all affine maps.

The search loop for each P is:

1. Construct $f_0, \ldots, f_5, g_0, \ldots, g_5$ subject to the target non-commutation pattern and the active orthogonality condition $\Psi_r = 0$ for $r \neq 3$.
2. Reject any candidate whose active Tanner graphs contain 4-cycles or 6-cycles, thereby enforcing girth 8.
3. Compute latent upper bounds from Ψ_3-kernel witnesses.
4. Compute exact latent lower bounds from Theorem 3.2 whenever the Ψ_3 kernels are block-constant.
5. Compute non-latent upper bounds by Proposition 4.2 over all divisors $m \mid P$.
6. Score the candidate by

$$\min\{d_X^{(\mathrm{lat}),\mathrm{ub}},\ d_Z^{(\mathrm{lat}),\mathrm{ub}},\ d_X^{(\mathrm{nonlat}),\mathrm{ub}},\ d_Z^{(\mathrm{nonlat}),\mathrm{ub}}\},$$

   and keep the best code for that P.

This differs from the earlier one-example workflow in an essential way: we do *not* stop when the first valid code is found. Instead, the search continues and checkpoint files are written whenever the best overall upper bound improves. The final artifact for each P is therefore the best code discovered under a fixed search budget.

6 Main Results

Table 1 is the main result of the paper. For each P we report the code with the largest overall upper bound found so far. Every listed code was checked to satisfy CSS orthogonality and girth 8.

The table should be read as a ladder rather than a loose collection of examples: for each P we rerun the entire search-and-certification loop and keep only the best surviving code. This makes Table 1 a benchmark-style summary of the current design frontier inside the present construction class.

The table reveals three structural regimes. For the smallest two codes, the latent and non-latent obstructions coincide. From $P = 576$ through $P = 3072$, the latent upper bound is the active bottleneck while the non-latent bound stays safely above it. The new $P = 3840$ winner flips the pattern again: the latent witnesses are now much heavier, and the overall upper bound is dictated by the non-latent side. This is the central design lesson of the updated study:

Table 1: Best current code for each lift size P. The code parameter $[\![n, k, \leq u]\!]$ means $[\![n, k, \leq u]\!]$, where u is the overall structural upper bound. The latent upper bound is the minimum of the X- and Z-side latent witnesses. The non-latent upper bound is the best witness found over all divisors $m \mid P$.

P	best code	latent ub	non-latent ub	best m_X, m_Z
216	$[\![2592, 1300, \leq 24]\!]$	24	24	$(12, 6)$
288	$[\![3456, 1732, \leq 24]\!]$	24	24	$(4, 2)$
384	$[\![4608, 2308, \leq 24]\!]$	24	48	$(2, 12)$
576	$[\![6912, 3460, \leq 48]\!]$	48	64	$(32, 32)$
768	$[\![9216, 4612, \leq 48]\!]$	48	96	$(16, 24)$
1536	$[\![18432, 9220, \leq 48]\!]$	48	256	$(16, 64)$
1920	$[\![23040, 11524, \leq 64]\!]$	64	120	$(5, 15)$
2688	$[\![32256, 16132, \leq 84]\!]$	84	128	$(2, 64)$
3072	$[\![36864, 18436, \leq 96]\!]$	96	192	$(32, 192)$
3840	$[\![46080, 23044, \leq 128]\!]$	480	128	$(32, 64)$

both latent and non-latent screening must be tracked throughout the search, because the active obstruction can switch as P grows.

The best current code is the $P = 3840$ entry

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

Its stored bound decomposition is

$$d_X^{(\mathrm{lat}),\mathrm{ub}} = d_Z^{(\mathrm{lat}),\mathrm{ub}} = 480, \qquad d_X^{(\mathrm{nonlat}),\mathrm{ub}} = 128, \qquad d_Z^{(\mathrm{nonlat}),\mathrm{ub}} = 128.$$

Thus the overall upper bound is dictated entirely by the non-latent side.

6.1 Exact latent certification on certified entries

The best upper-bound code and the best exact-certified latent code do not currently coincide. The new $P = 3840$ winner improves the structural upper-bound table to

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

but this improvement comes entirely from the non-latent screening stage. Its latent upper bound is still 480 and has not yet been matched by an exact lower-bound certificate. What is exact today is the earlier $P = 3840$ checkpoint

$$[\![46080, 23044, \leq 96]\!],$$

whose latent upper bound is fully certified:

$$d_X^{(\mathrm{lat})} = d_Z^{(\mathrm{lat})} = 96.$$

The certification proceeds exactly as in Theorem 3.2. Here $\mathrm{rank}(\Psi_3) = 3360$, so $\dim \mathrm{Ker}(\Psi_3) = \dim \mathrm{Ker}(\Psi_3^{\mathsf{T}}) = 480$. Since $3840 = 8 \cdot 480$, the kernels are 8-block constant. The compressed latent images therefore live in length $12 \cdot 480 = 5760$. We then ask whether either compressed latent image contains a nonzero vector of weight at most 11. Both satisfiability queries are UNSAT, which proves

$$d_X^{(\mathrm{lat})} \geq 8 \cdot 12 = 96, \qquad d_Z^{(\mathrm{lat})} \geq 8 \cdot 12 = 96.$$

Combined with the explicit weight-96 latent witnesses, this yields equality.

At the short-length end, the same method also certifies the $P = 216$ entry exactly: there $216 = 2 \cdot 108$, the compressed latent images have length $12 \cdot 108 = 1296$, and both sides are UNSAT below compressed weight 12, giving

$$d_X^{(\mathrm{lat})} = d_Z^{(\mathrm{lat})} = 24.$$

The same certificate generation has been automated by a dedicated certification script, `certify_latent_bounds.py`. The script detects the block factor from Ψ_3, compresses the latent image, and performs exact low-weight exclusion with a SAT or SMT backend. This automation is important for scale: once the construction is fixed, the lower-bound side becomes a reproducible computational certificate rather than a hand-crafted appendix. In other words, the latent lower-bound theory is not just existential; it is implemented as a reusable certification pipeline attached to the code table.

6.2 Verification and decoding sanity checks

Each code in Table 1 was checked by an independent decoder-side tool to satisfy

$$H_X H_Z^{\mathsf{T}} = 0, \qquad g(H_X) = g(H_Z) = 8.$$

We also ran a low-noise sanity test using joint BP with post-processing at

$$p = \frac{2}{n} \qquad \text{and} \qquad p = \frac{4}{n},$$

with 2000 trials per point. Across all nine codes, these tests produced zero frame failures. This does not prove a distance lower bound, but it does provide evidence that the table is not being driven by obvious hidden low-weight defects.

7 Discussion and Outlook

The present paper suggests a stronger narrative for high-girth binary CSS LDPC design. The contribution is no longer "we found one code with girth 8." Rather, it is:

> we developed a structural certificate pipeline that turns controlled non-commutativity into a reproducible best-by-lift table of binary CSS LDPC codes.

That framing is more durable and more scalable.

Several directions can make the story even stronger.

1. **Turn the best-by-lift ladder into a benchmark suite.** The files of Table 1 already form a useful public benchmark family. A natural next step is to publish them together with decoding traces, ETS statistics, and certificate logs.

2. **Release machine-checkable certificates.** The latent SAT/SMT proofs can be exported as proof logs or independently checkable witness bundles. That would turn the distance claims into auditable artifacts rather than narrative arguments in the paper.

3. **Track two ladders rather than one.** The new $P = 3840$ winner improves the upper-bound ladder, while the earlier $P = 3840$ checkpoint remains the strongest large-length code with a complete exact latent certificate. Reporting both ladders may better reflect the current state of the search.

4. **Upgrade the non-latent side from upper bounds to certificates.** At present the non-latent theory is strongest on the upper-bound side. A matching lower-bound theory for selected m-block spaces would significantly strengthen the distance claims.

5. **Optimize for a Pareto front rather than a single scalar.** The current search maximizes the minimum of latent and non-latent upper bounds. A more ambitious objective would jointly track rate, girth, latent exact distance, non-latent upper bound, and low-noise decoding.

6. **Exploit alternative active index sets.** For $(J, L) = (3, 12)$ the standard active choice leaves only $r = 3$ free. Non-consecutive active sets can shrink the active difference set and may allow larger latent certificates at shorter lengths.

In short, the main impact of the present work is methodological. The table is the visible output, but the deeper contribution is the latent/non-latent certificate pipeline that makes such a table possible.

References

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

[2] 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.

[3] M. Hagiwara and H. Imai. Quantum quasi-cyclic LDPC codes. In *IEEE International Symposium on Information Theory*, pages 806–810, 2007.

[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*, 60(10):5856–5874, 2014.

[5] Y. Komoto and K. Kasai. High-girth quantum LDPC constructions based on affine permutation matrices. 2025. preprint.

[6] T. J. Richardson. Error floors of LDPC codes. In *Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.