

Ultra-High-Rate Quantum LDPC Codes

Kenta Kasai Institute of Science Tokyo

Joint work



Jong Yeon Lee



Koki Okada



Nishad Maskara



Kenta Kasai



Hengyun Zhou

Design Principles for Ultra-High-Rate Quantum Codes: [arXiv:2609.30069](#)
Okada–Kasai, Low-Weight Canonical Logical Bases: [arXiv:2609.35601](#) (posted today)

Pair-partition codes guarantee rate with fewer checks

Pair-partition codes based on cyclic permutation matrices (CPM-PP)

$[[n, k, d]]$: n physical qubits, k logical qubits, distance d ; rate $R = k/n$.

$r = m_X + m_Z$: total check rows. $\delta = r - \text{rank } H_X - \text{rank } H_Z$: row dependencies.

$$\frac{k}{n} = \underbrace{1 - \frac{r}{n}}_{\text{design rate: row count}} + \underbrace{\frac{\delta}{n}}_{\text{extra rate: dependent rows}}.$$

Construction	Where the logical dimension comes from
Standard BB	$r = n$: zero design rate, so $k = \delta$.
Planar surface code	Standard independent checks: $r = n - 1$, $k = 1$, $R = 1/n$.
HGP / LP	Can have positive design rate. At short lengths, very high-rate examples often have small distance.
CPM-PP	$R \geq 1 - 2J/L$: the matrix shape sets a rate floor. Optimize the pairings for distance.

Few checks and large exact distance at short block length.

Each of H_X, H_Z contains $J \times L$ CPM blocks of size $P \times P$

Block layout shown: $J = 3, L = 8$

$$H_X = \begin{pmatrix} E_{00} & E_{01} & E_{02} & E_{03} & E_{04} & E_{05} & E_{06} & E_{07} \\ E_{10} & E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} & E_{17} \\ E_{20} & E_{21} & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} & E_{27} \end{pmatrix}$$

$$H_Z = \begin{pmatrix} D_{00} & D_{01} & D_{02} & D_{03} & D_{04} & D_{05} & D_{06} & D_{07} \\ D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} & D_{17} \\ D_{20} & D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} & D_{27} \end{pmatrix}$$

J block rows, L block columns: JL **CPMs in each matrix**.
Each full binary matrix has JP **rows** and LP **columns**.

$$n = LP, \quad r = 2JP, \quad \boxed{R \geq 1 - 2J/L}$$

Degree control: In classical LDPC codes, column-weight distribution is a sum-product BP design knob.
HGP/LP inherit degrees from product seeds; **CPM-PP directly prescribes uniform column weight J and row weight L** .

Inside every block E_{il}, D_{jl}

CPM = cyclic permutation matrix

$C(a)$: shift each row of I_P

cyclically **right by** a positions.

$a \in \mathbb{Z}_P$, size $P \times P$.

$$C(1) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (P = 4)$$

One 1 in every row and column.

In H_X, H_Z :

column weight J , row weight L .

PP = pair partition.

Equal product pairs cancel and guarantee CSS orthogonality

$$H_X = \begin{pmatrix} E_{00} & E_{01} & E_{02} & E_{03} & E_{04} & E_{05} & E_{06} & E_{07} \\ E_{10} & E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} & E_{17} \\ E_{20} & E_{21} & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} & E_{27} \end{pmatrix} \quad H_Z = \begin{pmatrix} D_{00} & D_{01} & D_{02} & D_{03} & D_{04} & D_{05} & D_{06} & D_{07} \\ D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} & D_{17} \\ D_{20} & D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} & D_{27} \end{pmatrix}$$

Colors indicate equal **products** for row pair (0,0). Individual E and D blocks may differ.

$$\text{CSS condition: } H_X H_Z^T = 0, \quad (H_X H_Z^T)_{ij} = \sum_{\ell=0}^{L-1} E_{i\ell} D_{j\ell}^T.$$

$$\begin{aligned} & \underbrace{E_{00} D_{00}^T + E_{03} D_{03}^T}_0 + \underbrace{E_{01} D_{01}^T + E_{02} D_{02}^T}_0 \\ & + \underbrace{E_{04} D_{04}^T + E_{06} D_{06}^T}_0 + \underbrace{E_{05} D_{05}^T + E_{07} D_{07}^T}_0 = 0. \end{aligned}$$

Why the pairs vanish: equal products have 1s in the same positions.
Over $\mathbb{F}_2$, $1 + 1 = 0$ and $0 + 0 = 0$, so each pair sums to the **zero matrix**.

Pair all terms for every (i, j) : every block of $H_X H_Z^T$ is zero.

Choose the pairs, then solve one linear system

For each block-row pair (i, j) , pair all L block columns in M_{ij} (L even).

$$M_{00} = \{\{0, 3\}, \{1, 2\}, \{4, 6\}, \{5, 7\}\}, \quad M_{01} \text{ may use different pairs.}$$

CPM blocks: $E_{i\ell} = C(e_{i\ell}), \quad D_{j\ell} = C(d_{j\ell}).$

$$\underbrace{E_{iu}D_{ju}^T = E_{iv}D_{jv}^T}_{\text{equal products in a chosen pair}} \iff \underbrace{e_{iu} - d_{ju} = e_{iv} - d_{jv}}_{\text{one linear equation in the shifts}} \pmod{P}.$$

Collect the pair equations modulo P

$$A_{\mathcal{M}}x = 0 \pmod{P}, \quad x = ((e_{i\ell})_{i,\ell}, (d_{j\ell})_{j,\ell})^T.$$

If P is prime, this is linear algebra over $\mathbb{F}_P$. Composite P is also allowed.

Example $J = 3, L = 8$: $2JL = 48$ shift variables and $J^2L/2 = 36$ equations before removing dependencies.

The pairing array $\mathcal{M}$ is the design choice. Solutions give commuting CPM checks.

Pairing guarantees orthogonality, not code quality

- 1 **Choose pair partitions.** Fill the $J \times J$ array $\mathcal{M} = (M_{ij})$.
- 2 **Screen pairing graphs.** Remove patterns that force short cycles before choosing shift labels.
- 3 **Solve for the shifts.** Find solutions of $A_{\mathcal{M}}x = 0 \pmod{P}$.
- 4 **Test completed lifts.** Check Tanner girth, rank, and distance on the actual binary matrices.

The zero-shift solution shows why testing is essential

All shifts zero always satisfy the pair equations, but repeated blocks create short cycles. A valid CSS code need not be a useful code.

Build in commutation. Verify girth, rank, and distance on each completed code.

Pair-partition codes give large exact distance at short lengths

(J, L)	$[[n, k, d]]$	Girth	Lift P	Rate k/n
(3, 8)	[[472, 122, 16]]	8	59	0.258
(3, 8)	[[776, 198, 20]]	8	97	0.255
(3, 10)	[[970, 392, 16]]	6	97	0.404
(3, 10)	[[1630, 656, 20]]	8	163	0.402
(3, 12)	[[2004, 1006, 16]]	6	167	0.502
(3, 12)	[[2676, 1342, 18]]	6	223	0.501
(3, 14)	[[1414, 812, 12]]	6	101	0.574
(3, 14)	[[3122, 1788, 16]]	6	223	0.573
(4, 10)	[[220, 50, 16]]	6	22	0.227
(4, 10)	[[320, 70, 20]]	6	32	0.219
(4, 12)	[[372, 130, 16]]	6	31	0.349
(4, 12)	[[492, 170, 20]]	6	41	0.346
(4, 14)	[[518, 228, 16]]	6	37	0.440
(4, 14)	[[574, 252, 18]]	6	41	0.439
(4, 16)	[[848, 430, 18]]	6	53	0.507
(4, 16)	[[944, 478, 20]]	6	59	0.506
(5, 10)	[[170, 8, 20]]	6	17	0.047

TABLE I. Representative CPM-based CSS pair-partition codes. A more extended list can be found in Table VIII.

CSS pair-partition codes with cyclic blocks

Different (J, L) choices give different rate–distance tradeoffs.

Distance 20 at short block lengths

$$[[320, 70, 20]], \quad k/n = 0.219.$$

Only 320 physical qubits, with 70 logical qubits.

Distance 20 at rate above one half

$$[[944, 478, 20]], \quad k/n = 0.506.$$

Fewer than 1000 physical qubits.

The shortest distance-20 example shown is $[[170, 8, 20]]$, at a lower rate of 0.047.

Exact distance needs a lower and an upper certificate

Lower-bound certificate

Complete the search up to weight W on **both CSS sides**.

- Reject nonzero syndrome.
- Reject vectors in the opposing stabilizer row space.
- If no logical survives, then $d > W$.

For these regular lifts, all kernel vectors have even weight, so exclusion through $d_0 - 2$ proves $d \geq d_0$.

Upper-bound certificate

Provide explicit vectors

$$v_Z \in \text{Ker } H_X \setminus \text{Row } H_Z,$$

$$v_X \in \text{Ker } H_Z \setminus \text{Row } H_X.$$

Their weights give $d_Z \leq \text{wt}(v_Z)$ and $d_X \leq \text{wt}(v_X)$.

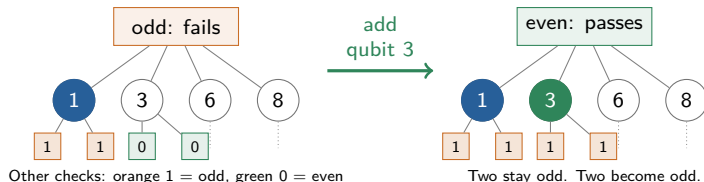
$$[d > d_0 - 2 \text{ and } d \text{ even}] \Rightarrow d \geq d_0, \quad \text{wt}(v) = d_0 \Rightarrow d \leq d_0,$$

Matching certificates determine d_X, d_Z , then $d = \min(d_X, d_Z)$.

Unsatisfied checks guide the search for a logical error

Goal: a small set of Z errors that passes every X check and changes logical information.

Adding a qubit changes all its neighboring checks



Each qubit meets J X checks (column weight $J = 3$ here).

Update every check, then continue at an odd check.

Alternative additions to qubit 1

Add 3

6 and 8 remain available

Add 6

exclude 3

Add 8

exclude 3 and 6

Complete the search up to weight limit W .

No logical on either the X or Z side certifies $d > W$.

The search repeats this step

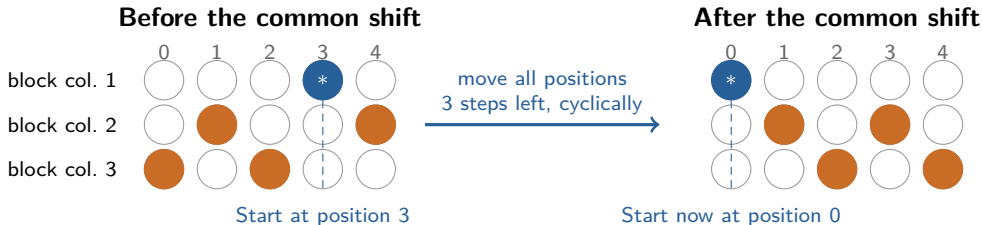
- 1 Start from one qubit.
- 2 Branch at an odd check and update all checks.
- 3 Prune impossible or over-limit completions.
- 4 All checks pass: reject stabilizers, otherwise report a logical.

$$H_X v = 0, \quad v \notin \text{Row } H_Z.$$

Cyclic symmetry leaves one starting point per block column

Shift every block column by the same amount. A logical stays logical, with the same weight.

Illustration: block size $P = 5$, with three block columns shown. Colored circles mark a candidate support.



Every block column needs only its position-zero representative.

$\underbrace{LP}$ $\longrightarrow$ $\underbrace{L}$

all physical starting qubits one start per block column

e.g. $8 \times 59 = 472 \longrightarrow 8$.

Each representative still requires a complete search.

Matrix processing can still grow with block size P .

Kasai codes and CPM-PP share the APM-PP framework

Generalize cyclic shifts to affine permutations $x \mapsto ax + b \pmod{P}$, with $a \in \mathbb{Z}_P^\times$.

APM-PP: pair-partition codes based on **affine permutation matrices**.

Affine blocks keep the same pair-cancellation condition

For every block-row pair (i, j) and every $\{u, v\} \in M_{ij}$,

$$E_{iu} D_{ju}^\top = E_{iv} D_{jv}^\top.$$

Equal products cancel over $\mathbb{F}_2$, so $H_X H_Z^\top = 0$.

CPM-PP

All affine multipliers equal one.

Pairing conditions reduce to linear equations on shift labels.

Kasai construction

K. Kasai, *Quantum* **10**, 2205 (2026).

$$[[9216, 4612, d \leq 48]]$$

$(3, 12)$ -regular, girth 8. 48 is a distance **upper bound**.

APM-PP unifies the two constructions. Kasai codes need not be CPM-PP codes.

Symplectic halving maps qubit pairs to Pauli coordinates

Required parent structure: after pairing qubits and choosing generators,

$$H_X = (A \mid B), \quad H_Z = (B \mid A).$$

Each half contains n physical qubits. Read the same two halves as Pauli components:

$$H_{\text{half}} = \left(\underbrace{A}_{\text{Pauli } X} \mid \underbrace{B}_{\text{Pauli } Z} \right) \text{ on only } n \text{ qubits.}$$

Paired bits	One Pauli
(0, 0)	I
(1, 0)	X
(0, 1)	Z
(1, 1)	Y

Why the new checks commute

$$H_X H_Z^T = AB^T + BA^T = 0.$$

This is exactly the symplectic commutation condition for H_{half} .

The parent must admit the paired form above. An arbitrary CSS code cannot always be halved.

Halving preserves rate while distance counts occupied pairs

Let $r_{\text{half}} = \text{rank}(A \mid B)$ and $k = n - r_{\text{half}}$.

	CSS parent	Halved non-CSS code
Physical qubits	$2n$	n
Logical qubits	$2k$	k
Encoding rate	$2k/(2n)$	k/n
Distance	d'	$\lceil d'/2 \rceil \leq d \leq d'$

One occupied pair costs one qubit, including a pair that becomes Y .

$$\text{wt}_P(x \mid z) = |\text{supp}(x) \cup \text{supp}(z)|.$$

Example: $(110 \mid 101)$ has binary weight 4 but becomes YXZ , of Pauli weight 3.

Halving never increases the supplied check weights. The new distance must be verified.

The benefit is a shorter block at the same rate, with an instance-dependent distance loss.

Halving gives distance 20 with 200 physical qubits

Check weight 10: $[[90, 21, 11]]$, $[[140, 31, 15]]$, $[[200, 43, 20]]$.

(J, L)	$[[n, k, d]]$	Lift P	Rate k/n
(3, 8)	$[[52, 15, 7]]$	13	0.288
(3, 8)	$[[316, 81, 15]]$	79	0.256
(3, 8)	$[[388, 99, 18]]$	97	0.255
(3, 10)	$[[500, 202, 14]]$	100	0.404
(3, 12)	$[[450, 227, 10]]$	75	0.504
(3, 14)	$[[476, 274, 9]]$	68	0.576
(4, 10)	$[[90, 21, 11]]$	18	0.233
(4, 10)	$[[140, 31, 15]]$	28	0.221
(4, 10)	$[[150, 33, \leq 16]]$	30	0.220
(4, 10)	$[[200, 43, 20]]$	40	0.215
(4, 12)	$[[240, 83, \leq 17]]$	40	0.346
(4, 12)	$[[312, 107, \leq 21]]$	52	0.343
(4, 14)	$[[287, 126, \leq 16]]$	41	0.439
(4, 14)	$[[336, 147, \leq 18]]$	48	0.438
(4, 16)	$[[336, 171, \leq 15]]$	42	0.509

TABLE II. Representative halved non-CSS pair-partition codes. The full frontier is given in Table VII.

Distance entries marked $\leq$ are upper bounds. The comparison uses source-reported distances, including upper bounds.

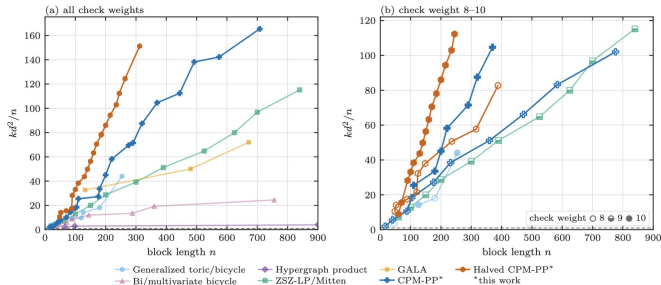


FIG. 8. **Encoding efficiency across construction families.** kd^2/n frontier as a function of blocklength n for different code families [6, 7, 14, 16, 23, 55–58]. (a) All check weights. (b) The same comparison restricted to check weights eight to ten, with the pair-partition templates (3, 8) and (4, 10) drawn as separate series and check weight distinguished by marker fill. The dashed line marks the rotated surface code, $kd^2/n = 1$. Each code contributes the distance its source quotes, which is an upper bound wherever the distance is not yet exact; this applies to our families and to the comparison families alike.

Low-weight CSS logical bases need canonical pairing

Goal: light representatives for all k logical qubits, with canonical pairing.

$$H_Z x_a = 0, \quad H_X z_b = 0, \quad x_a^\top z_b = \delta_{ab}.$$

x_a anticommutes only with its partner z_a .

① **Choose logical classes and partners.** Use cyclic symmetry to organize translated seeds and verify their pairing.

② **Reduce physical weight within each class.** Add stabilizers:

$$z'_b = z_b + H_Z^\top u, \quad \min_u \text{wt}(z_b + H_Z^\top u).$$

The logical action and pairing stay unchanged because $H_Z x_a = 0$.

③ **Verify the resulting basis.** Check syndromes, canonical pairing, and the physical weights.

Our CSS example $[[184, 50, 10]]$: 46 canonical pairs have weights (10, 12) or (12, 10). Four residual pairs are heavier. The mean across all 100 operators is 12.6.

The goal is a full set of light, correctly paired logical operators for computation.

A complete canonical basis can retain low weight

Koki Okada and Kenta Kasai arXiv:2609.35601v1 (28 Sep 2026)

Low-Weight Canonical Logical Bases from Pair-Partition Codes

Explicit complete logical bases

- 1 Assign $\mathbb{F}_4$ coefficients that preserve PP cancellation.
- 2 Disjoint invertible check submatrices give sparse cofactor seeds.
- 3 One invertible pairing polynomial g normalizes every pair. Expand to binary.

Three reported binary codes (exact d)

$[[n, k, d]]$	All k pairs	Weight of each
		X / Z logical
$[[320, 80, 14]]$	80	25 / 25
$[[448, 112, 18]]$	112	27 / 27
$[[2048, 512, 24]]$	512	27 / 27

Weights refer to the constructed logical basis.

Weight-preserving normalization: if $g(z) = z^\nu$ with coefficient 1, normalization is just a cyclic shift.

Binary expansion creates a new code of length $2LP$; check weight rises from 8 to 10 in these examples.

All logical qubits receive explicit, low-weight canonical X/Z partners.

Pair-partition codes achieve large distance at short lengths

- 1 **Orthogonality by paired cancellation.** The two check matrices use CPM blocks. Each chosen pair contributes two equal products, whose sum is zero over $\mathbb{F}_2$.
- 2 **Construction by linear equations.** Choose the cancellation pairs, then solve for their shift labels. Check girth and distance on the completed code.
- 3 **Short codes with large minimum distance.** Examples include $[[320, 70, 20]]$ and $[[944, 478, 20]]$, with different rate–length tradeoffs.
- 4 **Compact codes and light logical operators.** For suitable parents, symplectic halving halves the block length at fixed rate. Quaternary PP extensions give explicit complete canonical bases under the stated conditions.

Takeaway

Pair-partition codes with cyclic blocks give short quantum LDPC codes with certified distance.

Pairing guarantees orthogonality. Distance certification concerns each completed code.