TWENTY EXACT MINIMUM DISTANCES FOR GENERALIZED $\mathbb{Z}_p$ TORIC CODES ON TWISTED TORI

KOREA SUPERINTELLIGENCE LABS

Abstract. In arXiv:2602.20158, Liang and Chen build qudit CSS low-density parity-check codes from a pair of weight-three Laurent polynomials over a twisted two-dimensional torus, and tabulate sixty-five of them at $p \in \{3, 5, 7, 11\}$. The distance column of those tables is the output of a randomized information-set algorithm, and its caption states that the printed values are upper bounds on the exact distance which the authors "believe are tight in practice". We prove that twenty of them are exact: for each of the twenty codes of Table 1 the minimum weight of a logical operator is exactly the printed d, including $[\![48, 4, 9]\!]_3$, $[\![40, 4, 9]\!]_5$, $[\![32, 4, 8]\!]_7$ and $[\![40, 4, 10]\!]_{11}$. All twenty agree with the printed number, so the numbers themselves are not new; what is new is the lower bound, which no source establishes. Each proof is two-sided: an explicit weight-d logical operator, certified not to be a product of stabilizers by a single pairing, together with a search over every support of size less than d in each of the two directions, of which there are 2 011 668 932 in all. The soundness of that search rests on a constructive echelon substitution that replaces the usual rank–nullity argument by an explicit coefficient vector. Every statement below is machine-checked in Lean 4.

1. Introduction

The codes. Fix a prime p and a sublattice $\Lambda = \langle (0, \alpha), (\beta, \gamma) \rangle$ of $\mathbb{Z}^2$, so that $\Gamma = \mathbb{Z}^2/\Lambda$ is a finite abelian group of order $N = \alpha\beta$ — a *twisted torus*, twisted in the sense that the second generator (β, γ) need not be orthogonal to the first. A pair of Laurent polynomials $f, g \in \mathbb{F}_p[x^{\pm 1}, y^{\pm 1}]$ acts on the group algebra $\mathbb{F}_p[\Gamma]$ by multiplication, and the two matrices

$$H_X = \big[\, A_f \mid A_g \,\big], \qquad H_Z = \big[\, -A_{\bar{g}} \mid A_{\bar{f}} \,\big] \tag{1}$$

of multiplication operators, with $\bar{\ }$ the antipode $x^a y^b \mapsto x^{-a} y^{-b}$, are the X- and Z-check matrices of a CSS code on $n = 2N$ qudits of local dimension p. This is the construction of Liang and Chen [1], a generalization to odd prime local dimension both of the bivariate bicycle qubit codes of Bravyi, Cross, Gambetta, Maslov, Rall and Yoder [3] and of the generalized toric codes on twisted tori of Liang, Liu, Song and Chen [4]. Taking f and g of weight three keeps every check of weight six, so the codes are low-density parity-check codes, and the point of the twist is the encoding rate it buys: the abstract of [1] names $[\![242, 10, 22]\!]_3$ and $[\![120, 6, 20]\!]_{11}$ as "both achieving $kd^2/n = 20$", against $kd^2/n = 1$ for the untwisted toric code $[\![2L^2, 2, L]\!]$.

The interest of [1] is a table: a search over $(p, \alpha, \beta, \gamma, f, g)$ producing sixty-five codes, nineteen at $p = 3$, sixteen at $p = 5$, sixteen at $p = 7$ and fourteen at $p = 11$, printed as $[\![n, k, d]\!]_p$ with block lengths reaching 242. The dimension k is exact — it is a rank computation, and [1] also gives it in closed form as

Date: August 30, 2026.

$k = 2\dim_{\mathbb{F}_p} \mathbb{F}_p[x^{\pm 1}, y^{\pm 1}]/\langle f, g, y^\alpha - 1, x^\beta y^\gamma - 1\rangle$, a formula the paper attributes to [4, 5]. The distance column is not exact, and [1] says so in the caption of its Table I, verbatim:

> "We emphasize that the code distances are obtained using the probabilistic algorithm of Ref. [2]; accordingly, the reported values of d are upper bounds on the exact distance. For each code, we sample 5,000–10,000 information sets and repeat the procedure 1,000 times to assess consistency, yielding upper bounds that we believe are tight in practice."

The algorithm referred to is `QDistRnd` [2], a randomized information-set search: it samples information sets, reads off a codeword from each, and returns the least weight it ever saw. Every weight it returns is realized, so the output is an upper bound; no run of it, however long, is a lower bound. The same hedge is repeated in the body of [1], and it is also the practice of the other 2026 paper tabulating codes of this shape [6]. So all sixty-five printed distances are, as of this writing, unverified upper bounds, and the question this paper answers is the one the caption raises: are they the exact distances?

Results. For twenty of the sixty-five codes — every one whose exhaustive support search fits our computation budget — the answer is yes.

- **Exact distances** (Theorems 6–9). For each of the twenty codes of Table 1, the minimum weight of a logical operator is exactly the d printed by [1]: five codes at $p = 3$ with d up to 9, five at $p = 5$ with d up to 9, five at $p = 7$ with d up to 8, and five at $p = 11$ with d up to 10. Each of the twenty is proved in both directions, i.e. against X-type and Z-type logical operators separately.
- **The certificates** (Section 3). The lower bounds rest on a reduction that is elementary but, in the form used here, does without any dimension count: a checked reduced row echelon form of H_Z turns "v is orthogonal to a computed spanning set of $(\operatorname{rowsp} H_Z)^\perp$" into "$v \in \operatorname{rowsp} H_Z$" by producing the coefficient vector explicitly (Proposition 3), and a depth-first walk over supports whose echelon invariant costs nothing to maintain (Proposition 4) turns the finite search into that hypothesis. Neither is new mathematics and neither is claimed as such; they are recorded because they are what makes a single finite evaluation carry a distance statement.

What is new, and what is not. All twenty computed distances *agree* with the values printed in [1]. The numbers are therefore not new, and this paper is not a correction: twenty for twenty is evidence that the method of [1] is reliable in this range, not evidence against it. What is new is the lower half. A statement such as "the code $[\![48, 4, 9]\!]_3$ of [1] has a logical operator of weight 9 and none of weight 8" is not made by [1], is not implied by any run of `QDistRnd`, and we did not find it made anywhere else. The construction, the parameters $(\alpha, \beta, \gamma, f, g)$, the block lengths and the dimensions are all from [1] and are quoted, not derived.

No record is claimed either, and that is worth stating, because these parameters lie inside the range of the standard table. Grassl's tables [8] carry bounds for $[\![n, k, d]\!]_q$ at $q \in \{3, 5, 7\}$ up to length 100, hence for fifteen of the twenty (q, n, k) triples below, and in every one of the fifteen the best code they record beats the code proved here: $[\![16, 4]\!]_3$ at $d = 6$ against 4, $[\![48, 4]\!]_3$ at $d \geq 14$ against 9, $[\![40, 4]\!]_5$

at $d \geq 14$ against 9, $[\![32, 4]\!]_7$ at $d \geq 12$ against 8, and so on (consulted 30 August 2026; the five $q = 11$ triples are outside those tables). That is as it should be: [1] searches for the largest kd^2/n inside a family of weight-six codes, not for the largest d among all codes. What is proved below is the distance of one named low-density code, never a bound on what codes of those parameters can achieve.

Before formalizing anything we reproduced the published dimensions as a check on our reading of the tables: building H_X and H_Z from $(p, \alpha, \beta, \gamma, f, g)$ alone and computing $k = n - \operatorname{rank} H_X - \operatorname{rank} H_Z$ over $\mathbb{F}_p$ by Gaussian elimination returns all *sixty-five* printed values of k, with $\operatorname{rank} H_X = \operatorname{rank} H_Z$ and $H_X H_Z^{\mathsf{T}} = 0$ in every one. A misreading of the tables would not reproduce sixty-five dimensions. The dimension is not part of the formal development below; it is quoted from [1] throughout, and the reproduction is a check on published data rather than a result.

On the search for prior claims. The following was run on 30 August 2026. (i) The text of [1] was read in full from the live e-print: it states the hedge twice and makes no exactness claim and no lower-bound claim anywhere. (ii) The arXiv API returns a single version of it, so the copy we read and the current paper agree. (iii) Two citation indexes, OpenAlex and Semantic Scholar, each report no citing works at all for it, the second returning an empty citation list; a positive control run against each in the same call returned a nonzero citation count for an older paper, so neither query is silently failing. (iv) A full-text search of a local corpus of 367 text shards of arXiv sources finds the identifier `2602.20158` in exactly one place, namely the paper's own record. (v) Four arXiv full-text queries — *twisted torus* with *qudit*; *minimum distance* with *bivariate bicycle* and *exact*; *qudit* with *minimum distance* and *LDPC*; *generalized* with *toric code* and *qutrit* — return, respectively, only [6], one 2025 existence paper on qubit codes, nothing at all, and six papers none of which concerns code distances. (vi) A fresh clone of the data repository behind the Error Correction Zoo [7] cites [1] exactly once, in the bivariate bicycle entry, and the whole sentence is "There exist qudit BB codes that achieve $kd^2/n = 20$" — the headline figure of [1] repeated, with no exactness claim and no distance table. None of the twenty codes of Table 1 is tabulated anywhere in that database: of their twenty parameter triples only $[\![16, 4, 4]\!]$ occurs there at all, and it occurs as the name of two qubit codes with no connection to [1].

Two caveats belong with that list, and we state them rather than round them off. First, both indexes of (iii) missed the one live citer that (vi) found, so their silence should be read as "not indexed", not as "uncited"; the cheap clone of a domain database was the better channel here, and it is the one we would repeat. Second, the local corpus of (iv) excludes `quant-ph` wholesale — [1] is in it at all only through its `math-ph` and `math.QA` cross-lists — so a citer posted to `quant-ph` alone would not appear in it, and its silence carries correspondingly less weight than a citation index would.

What is deliberately not claimed. Nothing here bears on the remaining forty-five printed distances of [1]: they are neither confirmed nor doubted, only untouched. In particular the twenty codes below are exactly the twenty whose search we ran to completion *inside Lean*; two further codes that a prototype in C settled are excluded on purpose and are discussed, as a measurement and not as a theorem, in Remark 13.

2. Preliminaries

Throughout, p is a prime and $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$. Vectors are row vectors; for $v \in \mathbb{F}_p^n$ the *weight* $\mathrm{wt}(v)$ is the number of nonzero entries, and for a matrix M over $\mathbb{F}_p$ with n columns

$$\operatorname{rowsp} M = \Big\{ \textstyle\sum_i c_i M_i \ : \ c \in \mathbb{F}_p^m \Big\} \subseteq \mathbb{F}_p^n, \qquad \ker M = \{ v \in \mathbb{F}_p^n : M v^{\mathsf{T}} = 0 \},$$

where $M_1, \dots, M_m$ are the rows of M. We write $Mv = 0$ for $Mv^{\mathsf{T}} = 0$.

2.1. **The construction.** Let $\alpha, \beta \geq 1$ and $\gamma \in \mathbb{Z}$, put $\Lambda = \langle a_1, a_2 \rangle$ with $a_1 = (0, \alpha)$ and $a_2 = (\beta, \gamma)$, and let $\Gamma = \mathbb{Z}^2/\Lambda$, a group of order $N = \alpha\beta$. For a Laurent polynomial $h = \sum_{(a,b)} h_{a,b}\, x^a y^b$ over $\mathbb{F}_p$ write h_u, for $u \in \Gamma$, for the sum of the coefficients $h_{a,b}$ over exponent pairs (a, b) whose class in Γ is u, and let $A_h \in \mathbb{F}_p^{\Gamma \times \Gamma}$ be the matrix of multiplication by h on the group algebra $\mathbb{F}_p[\Gamma]$,

$$(A_h)_{t,s} = h_{s-t}, \qquad t, s \in \Gamma.$$

The antipode is $\bar{h} = \sum_{(a,b)} h_{a,b}\, x^{-a} y^{-b}$, so that $A_{\bar{h}} = A_h^{\mathsf{T}}$. Identify the $n = 2N$ coordinates with $\Gamma \sqcup \Gamma$ and define $H_X, H_Z \in \mathbb{F}_p^{N \times n}$ by (1). Since multiplication operators commute,

$$H_X H_Z^{\mathsf{T}} = -A_f A_g + A_g A_f = 0,$$

so (H_X, H_Z) is a CSS code, of dimension $k = n - \operatorname{rank} H_X - \operatorname{rank} H_Z$. In the twenty cases of Table 1, f and g have three terms each, so every row of H_X and of H_Z has weight at most six.

For the tabulated data to be reproducible one must also fix the identification of Γ with $\{0, 1, \dots, N-1\}$: we take the representative (i, j) of a class with $0 \leq i < \beta$ and $0 \leq j < \alpha$ and store it at index $i\alpha + j$. Of the two blocks of N coordinates, the first is the one carrying the columns of A_f and of $-A_{\bar{g}}$, the second the columns of A_g and of $A_{\bar{f}}$, as in (1).

2.2. **Logical operators and the distance.**

Definition 1. A vector $v \in \mathbb{F}_p^n$ is a *logical operator* of the CSS code (H_X, H_Z) if

$$\big(H_X v = 0 \text{ and } v \notin \operatorname{rowsp} H_Z \big) \quad \text{or} \quad \big(H_Z v = 0 \text{ and } v \notin \operatorname{rowsp} H_X \big),$$

and the *distance* of the code is $d = \min\{ \mathrm{wt}(v) : v \text{ a logical operator} \}$.

The two disjuncts are the Z-type and X-type logical operators; writing d_Z and d_X for the two one-sided minima, $d = \min(d_X, d_Z)$, which is the quantity tabulated in [1]. We emphasise that "$v \notin \operatorname{rowsp} H_Z$" is used literally — there is no $c \in \mathbb{F}_p^N$ with $\sum_i c_i (H_Z)_i = v$ — and not through any surrogate such as a syndrome or a symplectic pairing; Proposition 12(b) below shows that this distinction is doing real work here, since $\ker H_X$ contains vectors of weight well below d.

3. Certificates for an exact distance

This section records the reduction: three finite, checkable conditions on explicit data that together pin the distance of one code. None of it is new mathematics. What matters for the results of Section 4 is which parts are *trusted*: only the displayed equalities and inequalities are ever checked, while the row reduction that produces the auxiliary matrices R and T below, and the order in which the search visits supports, are never assumed to be correct.

3.1. **The upper bound.**

Proposition 2. *Let A, M be matrices over $\mathbb{F}_p$ with n columns and let $v, u \in \mathbb{F}_p^n$ satisfy*

$$Av = 0, \qquad Mu = 0, \qquad u \cdot v \neq 0.$$

Then $v \in \ker A$ and $v \notin \operatorname{rowsp} M$. In particular, taking $(A, M) = (H_X, H_Z)$, v is a logical operator and $d \leq \operatorname{wt}(v)$.

Proof. If $v = \sum_i c_i M_i$ then $u \cdot v = \sum_i c_i \, (u \cdot M_i) = \sum_i c_i \, (Mu)_i = 0$, contradiction. □

That one pairing replaces every dimension argument on the upper side: a single vector u in $\ker M$ with $u \cdot v \neq 0$ certifies non-membership in a row space of N rows.

3.2. **The span bridge.** The lower side is harder. An exhaustive search can establish that v is annihilated by a list B of vectors spanning (a subspace of) $(\operatorname{rowsp} M)^{\perp}$, and what is wanted is $v \in \operatorname{rowsp} M$. The usual route is a rank–nullity count. The following replaces it by a substitution that outputs the coefficient vector, so that nothing has to be known about dimensions — or about p being prime.

Proposition 3. *Let $M \in \mathbb{F}_p^{m \times n}$, let $R \in \mathbb{F}_p^{r \times n}$, $T \in \mathbb{F}_p^{r \times m}$ and let $j_1, \ldots, j_r \in \{0, \ldots, n-1\}$ be such that*

(a) *$R = TM$;*
(b) *$R_{i,j_l} = \delta_{il}$ for all $1 \leq i, l \leq r$.*

Call $q \in \{0, \ldots, n-1\}$ free if $q \notin \{j_1, \ldots, j_r\}$ and set $u^q = e_q - \sum_{i=1}^{r} R_{i,q} \, e_{j_i}$ for each free q. If $v \in \mathbb{F}_p^n$ satisfies $u^q \cdot v = 0$ for every free q, then

$$v = \Big(\sum_{i=1}^{r} v_{j_i} T_i \Big) M \;\in\; \operatorname{rowsp} M.$$

Proof. Put $w = \sum_i v_{j_i} R_i$. At a pivot column, $w_{j_l} = \sum_i v_{j_i} R_{i,j_l} = v_{j_l}$ by (b). At a free column q, $0 = u^q \cdot v = v_q - \sum_i R_{i,q} v_{j_i} = v_q - w_q$. Every column is of one kind or the other, so $v = w$, and by (a) $w = \sum_i v_{j_i} (T_i M) = (\sum_i v_{j_i} T_i) M$. □

Conditions (a) and (b) are exactly what an honest reduced row echelon form of M with pivot columns $j_1, \ldots, j_r$ satisfies, and they are checked, entry by entry, on the R, T and pivot list that a row reduction produced. The reduction itself is untrusted: if it returns garbage, (a) or (b) fails and nothing is proved. In the formal development a third condition is checked as well — that every row of M is recovered as $\sum_i M_{t,j_i} R_i$ — but it is not used in the proof above and is retained only as a redundant consistency check on the elimination.

3.3. **The exhaustion.** Write B for the matrix whose rows are the u^q of Proposition 3, one per free column, and stack it under A:

$$C = \begin{pmatrix} A \\ B \end{pmatrix} \in \mathbb{F}_p^{(m+r') \times n}, \qquad r' = n - r,$$

with columns $c_1, \ldots, c_n \in \mathbb{F}_p^{m+r'}$. The searched object is a support: a subset $S \subseteq \{1, \ldots, n\}$ of size at most b, standing for all v with $\operatorname{supp}(v) \subseteq S$ at once. The search carries along a *sequential echelon list* $E = ((i_1, w_1), \ldots, (i_l, w_l))$ with every pivot i_t among the first m coordinates and $(w_t)_{i_t} = 1$, and reduces a vector by applying $v \mapsto v - v_{i_t} w_t$ for $t = 1, \ldots, l$ *in order*; call the result $\rho_E(v)$. The

walk is depth-first over supports in increasing index order, and at each column s it computes $\rho_E(c_s)$ and does one of three things:

- if $\rho_E(c_s)$ is nonzero somewhere among the first m coordinates, at the least such index i, it appends $\big(i, \rho_E(c_s)_i^{-1}\rho_E(c_s)\big)$ to the *end* of E and descends;
- if $\rho_E(c_s) = 0$ entirely, it descends with E unchanged;
- if $\rho_E(c_s)$ vanishes on the first m coordinates but not below, it stops and reports failure.

Proposition 4. *If the search above, started with E empty, all n columns and budget b, never reports failure, then every $v \in \mathbb{F}_p^n$ with $Av = 0$ and $\mathrm{wt}(v) \le b$ satisfies $Bv = 0$; hence, by Proposition 3, $v \in \mathrm{rowsp}\, M$.*

Proof sketch. The invariant carried down the walk is that every column visited so far reduces to 0 under the current E. Appending new entries at the *end* of E is what makes it free of cost: reducing an already-zero residue by one more entry leaves it zero, so the invariant is inherited by every descendant without any recomputation. Given that, the soundness needs only three elementary facts. Reduction is linear, so $\rho_E(\sum_s \lambda_s c_s) = \sum_s \lambda_s \rho_E(c_s)$. A vector that vanishes on the first m coordinates is *fixed* by every reduction step, because every pivot i_t lies in those coordinates. And at the branch where a new pivot is added, the residue of the tail is the old combination rescaled by a modular inverse whose existence is not assumed but computed, the identity $r \cdot r^{-1} \equiv 1$ being itself part of what is checked — which is why no primality hypothesis enters. Now let $\lambda = v$ have weight at most b with $Av = 0$. Its support was visited, the top m coordinates of $\rho_\emptyset(\sum_s v_s c_s) = Cv$ are $Av = 0$, and the walk did not report failure on it, so the remaining coordinates Bv vanish as well. □

3.4. **Assembly.**

Proposition 5. *Let (H_X, H_Z) be as in (1) and $d \ge 1$. Suppose*

(i) *the echelon data of Proposition 3 passes its check for $M = H_Z$ and for $M = H_X$, and $v, u \in \mathbb{F}_p^n$ pass the check of Proposition 2 with $\mathrm{wt}(v) = d$;*
(ii) *the search of Proposition 4 succeeds at budget $d-1$ for $(A, M) = (H_X, H_Z)$;*
(iii) *and likewise for $(A, M) = (H_Z, H_X)$.*

Then every logical operator has weight at least d, and there is one of weight exactly d; that is, the code has distance exactly d.

Proof. The second half is Proposition 2. For the first, let x be a logical operator with $\mathrm{wt}(x) \le d-1$. If $H_X x = 0$ then (ii) with Propositions 4 and 3 gives $x \in \mathrm{rowsp}\, H_Z$, and if $H_Z x = 0$ then (iii) gives $x \in \mathrm{rowsp}\, H_X$; either way Definition 1 fails. □

4. Twenty exact distances

Table 1 lists the twenty codes, with the parameters as printed in [1] and the size of the support search each one required. Since the search visits every subset of at most $d-1$ of the n columns, that size is exactly $\sum_{w<d} \binom{n}{w}$, per direction.

Theorem 6. *Each of the five qutrit codes $[\![16, 4, 4]\!]_3$, $[\![26, 6, 5]\!]_3$, $[\![42, 4, 8]\!]_3$, $[\![48, 4, 9]\!]_3$ and $[\![52, 8, 7]\!]_3$ of Table 1 has minimum distance exactly the value d printed there: its logical operators all have weight at least d, and one of weight exactly d exists.*

p	$[\![n,k,d]\!]_p$	(α,β,γ)	f	g	$\sum_{w<d}\binom{n}{w}$
3	$[\![16,4,4]\!]$	$(4,2,1)$	$1+x+y$	$1+y-x^{-1}y$	697
3	$[\![26,6,5]\!]$	$(13,1,5)$	$1+x+x^{-1}y$	$1-y-x^{-1}$	17 902
3	$[\![42,4,8]\!]$	$(3,7,1)$	$1+x+x^{-2}y^{-1}$	$1+y+x$	33 199 096
3	$[\![48,4,9]\!]$	$(8,3,3)$	$1-x+xy^2$	$1-y+x^2$	465 174 935
3	$[\![52,8,7]\!]$	$(13,2,5)$	$1+x+x^{-1}y^{-2}$	$1+y+xy^{-1}$	23 251 684
5	$[\![16,4,4]\!]$	$(4,2,1)$	$1+x+3y$	$1+3y+x^{-1}y$	697
5	$[\![20,4,5]\!]$	$(5,2,1)$	$1+2x+4x^{-1}y$	$1+3y+4x$	6 196
5	$[\![24,4,6]\!]$	$(4,3,2)$	$1+2x+2xy$	$1+3y+x^{-1}y$	55 455
5	$[\![30,4,7]\!]$	$(15,1,6)$	$1+x+3x^2y^{-1}$	$1+2y+2x^2y$	768 212
5	$[\![40,4,9]\!]$	$(4,5,1)$	$1+x+3x^2y^{-1}$	$1+3y+x^2y$	100 146 724
7	$[\![12,4,3]\!]$	$(2,3,0)$	$1+2x+5y$	$1+2y+2x^{-1}$	79
7	$[\![14,4,4]\!]$	$(7,1,3)$	$1+2x+4xy$	$1+4y+2xy$	470
7	$[\![24,4,6]\!]$	$(4,3,2)$	$1+2x+5y$	$1+2y+2x^{-1}$	55 455
7	$[\![28,4,7]\!]$	$(7,2,3)$	$1+5x+x^{-1}y^{-1}$	$1+5y+x$	499 178
7	$[\![32,4,8]\!]$	$(4,4,1)$	$1+x+2x^2y$	$1+2y+3x^{-1}y^2$	4 514 873
11	$[\![16,4,4]\!]$	$(4,2,1)$	$1+7x+4xy$	$1+10y+3x^{-1}y^2$	697
11	$[\![20,4,5]\!]$	$(5,2,1)$	$1+x+2xy$	$1+7y+10xy^{-1}$	6 196
11	$[\![22,4,6]\!]$	$(11,1,3)$	$1+5x+5x^{-1}$	$1+8y+2xy^{-1}$	35 443
11	$[\![32,4,8]\!]$	$(8,2,1)$	$1+3x+10y^3$	$1+10y+3x^{-1}y^2$	4 514 873
11	$[\![40,4,10]\!]$	$(10,2,4)$	$1+x+10x^{-1}y^{-1}$	$1+7y+10xy^{-1}$	373 585 604

TABLE 1. The twenty codes. Parameters $[\![n,k,d]\!]_p$, (α,β,γ), f and g are as printed in [1] (Tables I–IV there, one per prime); $n=2\alpha\beta$ in every row. The last column is the number of supports the exhaustive search visits, in each of the two directions.

Theorem 7. *The same holds for the five codes* $[\![16,4,4]\!]_5$, $[\![20,4,5]\!]_5$, $[\![24,4,6]\!]_5$, $[\![30,4,7]\!]_5$ *and* $[\![40,4,9]\!]_5$ *of Table 1.*

Theorem 8. *The same holds for the five codes* $[\![12,4,3]\!]_7$, $[\![14,4,4]\!]_7$, $[\![24,4,6]\!]_7$, $[\![28,4,7]\!]_7$ *and* $[\![32,4,8]\!]_7$ *of Table 1.*

Theorem 9. *The same holds for the five codes* $[\![16,4,4]\!]_{11}$, $[\![20,4,5]\!]_{11}$, $[\![22,4,6]\!]_{11}$, $[\![32,4,8]\!]_{11}$ *and* $[\![40,4,10]\!]_{11}$ *of Table 1.*

Proof of Theorems 6–9. Each of the twenty is an instance of Proposition 5, and the three hypotheses of that proposition are discharged by exhaustive computation. We say precisely what was computed.

(i) The upper bound and the echelon data. For each code, an explicit pair $(v,u)\in\mathbb{F}_p^n\times\mathbb{F}_p^n$ is exhibited with $H_Xv=0$, $\mathrm{wt}(v)=d$, $H_Zu=0$ and $u\cdot v\neq 0$, and the check of Proposition 2 is evaluated on it. Together with it, the conditions (a), (b) of Proposition 3 are evaluated entry by entry on the reduced row echelon data of H_Z and of H_X — N rows of T against N rows of M, and the $r\times r$ identity pattern on the pivot columns — as are the shape conditions (every row of H_X and H_Z has length n). This is a finite evaluation costing seconds per code.

(ii) and (iii) The exhaustion. For each code and each of the two directions, the depth-first walk of Proposition 4 is run to completion at budget $d-1$ over all n columns of the stacked matrix C, and reports no failure. The number of supports visited is $\sum_{w<d}\binom{n}{w}$, the last column of Table 1; summed over both directions of all twenty codes it is 2 011 668 932, of which 1 043 288 628 at $p=3$, 201 954 568 at

$p = 5$, 10 140 110 at $p = 7$ and 756 285 626 at $p = 11$. The single largest is $[\![48, 4, 9]\!]_3$ at 465 174 935 supports per direction. The searches are complete, not sampled: no support of size below d is skipped, and no bound on p-ary coefficients is imposed, because a support is searched once for all $(p-1)^{|S|}$ vectors carried by it.

There is no case analysis and no appeal to symmetry: the twenty codes are treated one at a time and the two directions of each are separate computations. □

Remark 10 (what the two directions give separately). The two exhaustions of a code give $d_Z \geq d$ and $d_X \geq d$ separately, and the weight-d witness certified above is Z-type. So what is proved is $d_Z = d$ exactly, $d_X \geq d$, and hence $d = \min(d_X, d_Z)$ — the quantity printed in [1]. A computation outside the formal development (the same search reimplemented in C, see Section 6) finds an X-type logical operator of weight exactly d in each of the twenty codes as well, so $d_X = d_Z = d$ in fact; that half rests on an unverified program and is stated here only as a remark.

Remark 11 (distance and dimension are independent of each other here). Theorems 6–9 say nothing about k. The value of k enters the statements only through the codes' names; the proofs use H_X and H_Z and never a dimension, which is the point of Proposition 3. A reader who doubted the printed k would still get the distance statements unchanged.

5. Controls

A search that can only answer "yes" proves nothing, and a hypothesis that no vector satisfies is vacuous. The following four facts are computed on the same footing as the theorems above and are what rules those two failure modes out.

Proposition 12. (a) The search can report failure. *Run at budget* 8 *rather than* 7 *on the code* $[\![32, 4, 8]\!]_7$*, the very same search returns failure: it finds the weight-*8 *logical operator that is there.*

(b) The hypotheses are not vacuous, and the row-space condition is load-bearing. *For the same code, the first row of* H_Z *has weight* 6*, well below the distance* 8*, and lies in* $\ker H_X$*. It is excluded from Definition 1 only because it is a product of stabilizers — it lies in* $\operatorname{rowsp} H_Z$*, with the coefficient vector* e_0 *exhibited — and it also fails the other disjunct, since it is not in* $\ker H_Z$*.*

(c) The too-small claim is refuted. *The code* $[\![12, 4, 3]\!]_7$ *has no logical operator of weight* 2*, so its distance is not* 2*.*

(d) The too-large claim is refuted. *That code does have one of weight* 3*, so its distance is not* 4*.*

Part (b) is the one worth pausing on. Without it, the twenty theorems would be compatible with the far weaker and quite different statement that H_X has no low-weight kernel vector — which is *false* here, as (b) exhibits. The content of the exhaustion is about the quotient $\ker H_X / \operatorname{rowsp} H_Z$, and it is Proposition 3 that makes the quotient accessible to a finite search.

6. Verification

Every statement of Sections 3–5 has been formalised and machine-checked in Lean 4; the repository is private and a repository reference is to be added at submission. The development defines $\mathbb{F}_p$-vector arithmetic, the construction (1), the

row reduction, the search and all the checkers directly on lists of natural numbers, importing no library at all, so that the sweeps compile to a shared object and run as compiled code; the mathematical layer — Propositions 2, 3, 4 and 5, and the definitions of Section 2 — sits in a separate module that imports Mathlib only for $\mathbb{Z}/p\mathbb{Z}$ and a handful of tactics, and an audit of its axiom set finds nothing beyond propositional extensionality, the axiom of choice and quotient soundness: no incomplete proof anywhere. The finite evaluations are discharged by compiled evaluation (`native_decide`), which contributes one evaluation axiom per Boolean; an audit of the thirty statements involved — the twenty distance statements, the four controls and the six bridging theorems — reports, besides the three logical axioms just named, exactly seventy-three evaluation axioms and nothing else, each of them a finite assertion about explicit integer matrices. There is no subtraction anywhere in the computational layer (-1 is represented as $p-1$ and every operation is built from $+$ and $\cdot$ modulo p), so the proofs never meet a truncated natural-number subtraction. Three further cross-checks were run outside Lean and are reported as such: the check matrices were built independently in Python and compared entry by entry against the Lean construction on $[\![12,4,3]\!]_7$ (all 6×12 entries of both H_X and H_Z agree); the same support search was reimplemented in C and run on all twenty codes; and a third method sharing no idea with either — enumerate *every* vector of weight at most d and decide row-space membership by comparing ranks — reproduces the distance on the six codes where its cost is affordable, namely $[\![16,4,4]\!]_3$, $[\![26,6,5]\!]_3$, $[\![16,4,4]\!]_5$, $[\![20,4,5]\!]_5$, $[\![12,4,3]\!]_7$ and $[\![14,4,4]\!]_7$. None of these three is trusted by the formal development; they check the search *idea* rather than only its implementation.

7. Cost, and the frontier

The twenty codes of Table 1 are exactly the ones we could afford: the whole computation is about 7.9 hours of processor time, of which 3.8 hours is $[\![48,4,9]\!]_3$ alone, and its memory footprint never exceeds 0.8 GiB because the computational modules import nothing. The cost of a code is $2\sum_{w<d}\binom{n}{w}$ residue evaluations and grows very fast along the tables of [1].

Remark 13 (what a faster search would reach, and what it would not). The following are measurements of our own tooling, not theorems, and no statement of this paper depends on them. Of the forty-five codes of [1] that we did not settle, the three nearest are $[\![48,4,10]\!]_5$ and $[\![48,4,10]\!]_7$, at 2.14×10^9 supports per direction, and $[\![48,4,11]\!]_{11}$ at 8.68×10^9; then come $[\![62,6,10]\!]_5$ at 2.42×10^{10} and $[\![78,10,9]\!]_5$ at 2.64×10^{10}, and the most expensive cell in the tables, $[\![234,6,25]\!]_3$, needs about 3.9×10^{32} — out of reach of support enumeration by any constant factor. At our measured throughput the first two would cost about 8.5 hours of processor time per direction and the third about 34, against the roughly 4 hours per code we allowed ourselves; so none of the forty-five fits today. Three constant-factor improvements would change that, none of them implemented here. The first is the translation symmetry: Γ acts regularly on each half of the $n=2N$ coordinates and permutes the rows of H_X and of H_Z, so every logical operator has a translate whose support meets a fixed window, which cuts the walk by a constant factor. The second is a smaller detector: of the $n-\operatorname{rank} H_Z$ rows of B only about k carry independent information modulo $\operatorname{rowsp} H_X$, and replacing B by that many rows plus a certificate

would shorten every reduced vector. The third, and the largest, is a change of data structure in the inner reduction loop.

Two of those codes are in fact already settled, though not by us in a form we are willing to call a theorem. The C reimplementation mentioned in Section 6 ran $[\![48, 4, 10]\!]_5$ and $[\![48, 4, 10]\!]_7$ to completion, 2.14×10^9 supports per direction: it found no logical operator of weight at most 9 in either direction of either code, and weight-10 operators in both directions of both. So $d = 10$ holds for these two codes as a matter of computation. We record it as a measurement and nothing more — an unverified C program is evidence, not a proof — and the two codes are deliberately *not* among the twenty of Theorems 6–9.

Remark 14 (a note on the cost of verified enumeration). Also outside the formal development, and recorded because it is the reason the frontier sits where it does: on the identical algorithm and the identical inputs, the compiled Lean search runs 20 times slower than the C reimplementation at $n = 28$ and 67 times slower at $n = 48$, the gap widening with the block length because the per-node work is a vector operation whose every unit is a heap cell on one side and a byte of a cache-resident array on the other. Verified enumeration is affordable here, but its price is not a small constant, and it is not the same constant at every problem size.

8. Open questions

(1) *Are the other forty-five printed distances of* [1] *exact too?* Twenty for twenty is suggestive and nothing more. Two more, $[\![48, 4, 10]\!]_5$ and $[\![48, 4, 10]\!]_7$, are within reach of the improvements of Remark 13; most of the rest are not reachable by support enumeration at all. A Brouwer–Zimmermann style information-set search, made complete and then verified, would beat naive support enumeration outright; but the usual qubit intuition about the trade-off does not transfer, because enumerating messages over an information set of size r costs $(p-1)^r$ here while support enumeration is independent of p — which is why the naive method is the right baseline at $p = 11$.

(2) *Is there a structural lower bound?* Every distance we prove is proved by exhaustion. For the untwisted toric code the distance is a lattice quantity; for these twisted quotients we know of no argument giving even the right order of magnitude, and a proof that $d \geq 9$ for $[\![48, 4, 9]\!]_3$ not passing through 4.7×10^8 supports would be of much more interest than the fact itself.

(3) *Is $d_X = d_Z$ always?* It holds in all twenty codes, and one expects it from the shape of (1), but we neither assumed it nor proved it: each direction was searched separately (Remark 10).

(4) *The second table.* The seventy-six further qudit twisted-torus codes of [6] carry the same randomized hedge and would be natural targets. We did not attempt them: applying the construction of [1] to three of that paper's rows returned $k = 2$ against printed values 4, 6 and 4 (with n and d matching), so its conventions must be reconciled against its own table before any of its cells is touched.

References

[1] Z. Liang and Y.-A. Chen, *Generalized $\mathbb{Z}_p$ toric codes as qudit low-density parity-check codes*, arXiv:2602.20158v1 (23 February 2026). All quotations, table numbers and parameter values above are against the typeset v1, still the only version on 30 August 2026.

[2] L. P. Pryadko, V. A. Shabashov and V. K. Kozin, *QDistRnd: A GAP package for computing the distance of quantum error-correcting codes*, Journal of Open Source Software **7** (2022), no. 71, 4120; DOI 10.21105/joss.04120; arXiv:2308.15140.

[3] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall and T. J. Yoder, *High-threshold and low-overhead fault-tolerant quantum memory*, Nature **627** (2024), no. 8005, 778–782; DOI 10.1038/s41586-024-07107-7; arXiv:2308.07915.

[4] Z. Liang, K. Liu, H. Song and Y.-A. Chen, *Generalized toric codes on twisted tori for quantum error correction*, PRX Quantum **6** (2025), no. 2, 020357; DOI 10.1103/rmy6-9n89; arXiv:2503.03827.

[5] K. Chen, Y. Liu, Y. Zhang, Z. Liang, Y.-A. Chen, K. Liu and H. Song, *Anyon theory and topological frustration of high-efficiency quantum low-density parity-check codes*, Phys. Rev. Lett. **135** (2025), no. 7, 076603; DOI 10.1103/86j7-cmsw; arXiv:2503.04699.

[6] M. Halla, *Qudit twisted-torus codes in the bivariate bicycle framework*, arXiv:2602.04443v1 (4 February 2026).

[7] V. V. Albert and P. Faist (eds.), *The Error Correction Zoo* (2025), entry "Bivariate bicycle (BB) code" (code `qcga`), `https://errorcorrectionzoo.org/`. Read on 30 August 2026 in a shallow clone of the project's data repository `eczoo_data` at commit `5385c55c`; the Zoo is a living document and the entry may have changed since.

[8] M. Grassl, *Bounds on the minimum distance of linear codes and quantum codes*, online at `http://www.codetables.de`. Consulted 30 August 2026, when its quantum tables covered $[\![n, k, d]\!]_q$ for $q \in \{2, 3, 4, 5, 7, 8\}$, to length 256 at $q = 2$ and to length 100 otherwise. It records best-known parameters, not the individual codes of [1].

[9] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE-28, Lecture Notes in Computer Science **12699**, Springer, 2021.

[10] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020.