WEIGHT ISOMETRIES OF GRAPH STATES: THE WEIGHT FUNCTION UP TO $\mathrm{GL}(n,2)$ IS A COMPLETE INVARIANT FOR $n \leq 7$, AND A CORRECTION TO AN EXAMPLE OF PLLAHA

KOREA SUPERINTELLIGENCE LABS

ABSTRACT. A self-dual additive code over $\mathbb{F}_4$ of length n is an n-qubit stabilizer state, and every such code is equivalent to the graph code $C_G = \{(x, \Gamma_G x) : x \in \mathbb{F}_2^n\}$ of a graph G on n vertices; two graph codes are monomially equivalent — equivalently, the two states are related by local Clifford operations and a permutation of the qubits — exactly when the graphs are related by local complementation and vertex permutation. Pllaha's Problem 6.3 asks how far the symplectic isometries between two self-dual stabilizer codes can exceed the monomial ones, and Gluesing-Luerssen and Pllaha call their single-code form of the question "wide open". We settle the emptiness half of Problem 6.3 at lengths six and seven. If G and H are graphs on six vertices and some $\mathbb{F}_2$-linear $g\colon \mathbb{F}_2^6 \to \mathbb{F}_2^6$ satisfies $w_H(gx) = w_G(x)$ for all x — no invertibility assumed — then G and H are related by local complementations and vertex permutations; so the number of classes surviving the coarser weight equivalence is $W(6) = t(6) = 26$. At length seven the same holds for the 59 class representatives, which is $W(7) = 59$ once Danielsen and Parker's $t(7) = 59$ is granted. The weight *enumerator* is strictly weaker at both lengths: only $E(6) = 23$ enumerators occur among the 26 classes, and we exhibit an inequivalent pair sharing one. We also correct Example 5.10 of Pllaha's e-print arXiv:1807.09107v1: the two length-four codes it displays *are* monomially equivalent — exactly 32 of the $6^4 \cdot 4! = 31\,104$ monomial maps carry one onto the other — although the displayed isometry between them is realised by none. Outside the formal development, the same search run in C against Danielsen's orbit database extends the conclusion to $n \leq 11$, where $4\,501\,923$ inequivalent pairs share a weight enumerator and none is weight-isometric. Everything not explicitly labelled as computed outside the formal development is machine-checked in Lean 4.

1. Introduction

The objects. Fix $n \geq 1$ and a graph G on the vertex set $\{0, 1, \ldots, n-1\}$, with adjacency matrix $\Gamma_G \in \mathbb{F}_2^{n\times n}$ (symmetric, zero diagonal). Its *graph code* is the additive subgroup

$$C_G \;=\; \left\{ (x, \Gamma_G x) \;:\; x \in \mathbb{F}_2^n \right\} \;\subseteq\; (\mathbb{F}_2^2)^n,$$

where coordinate i of the codeword indexed by x is the pair $\big(x_i, (\Gamma_G x)_i\big) \in \mathbb{F}_2^2$. Identifying $\mathbb{F}_2^2$ with the Pauli letters I, X, Z, Y modulo phase, C_G is the label set of the stabilizer group of the n-qubit graph state of G, and it is a self-dual additive code over $\mathbb{F}_4$ of length n: the trace-symplectic form vanishes on it, and it has $\mathbb{F}_2$-dimension n. Conversely every self-dual additive $GF(4)$ code of length n is equivalent to a graph code [2, Thm. 6]. The (symplectic, equivalently Hamming) weight of the codeword indexed by x is the number of coordinates on which it is not I, namely

$$w_G(x) \;=\; \big|\mathrm{supp}(x) \cup \mathrm{supp}(\Gamma_G x)\big|.$$

Two operations act on graphs. *Local complementation* at a vertex u complements the subgraph induced on the neighbourhood of u; writing $a = \Gamma_G e_u$ for that neighbourhood,

$$\Gamma_{G\star u} \;=\; \Gamma_G + aa^{\mathsf{T}} + \mathrm{diag}(a).$$

Vertex permutations act by simultaneous row and column permutation, and are generated by the transpositions of adjacent vertices. Write $G \approx H$ when H arises from G by a finite sequence of these two kinds of move. By Bouchet's theorem [1], rediscovered in the quantum setting by Van den Nest, Dehaene and De Moor [9], $G \approx H$ holds exactly when C_G and C_H are monomially equivalent [2, Thm. 12];

and by [7, Thm. 5.8] monomial equivalence of the codes is exactly equivalence of the two quantum states under local Clifford operations composed with a qubit permutation. Let

$$t(n) \;=\; \#\{\text{graphs on } n \text{ vertices}\}/{\approx}$$

be the resulting class count; it is Danielsen and Parker's census of self-dual additive $GF(4)$ codes [2, Table 2], the sequence $1, 2, 3, 6, 11, 26, 59, 182, 675, 3990, 45144, 1323363$ for $n = 1, \ldots, 12$ [6, A094927].

The question, and its two halves. A *weight isometry* $C_G \to C_H$ is an $\mathbb{F}_2$-linear map $g\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ with

$$w_H(gx) \;=\; w_G(x) \qquad \text{for every } x \in \mathbb{F}_2^n.$$

Local complementation acts on a graph code by a monomial map [2, Thm. 11], and vertex permutation acts by a permutation of coordinates, which is monomial too; so $G \approx H$ forces the existence of a weight isometry $C_G \to C_H$. Whether the converse holds is the question of this paper, and it is Problem 6.3 of Pllaha [7], which we quote verbatim:

> Let $\mathcal{C},\ \mathcal{C}' \subseteq \mathbb{F}_q^{2n}$ be two self-dual stabilizer codes. Establish how different $\mathrm{rMon}(\mathcal{C},\ \mathcal{C}')$ and $\mathrm{Symp}(\mathcal{C},\ \mathcal{C}')$ can be. That is, let $H,\ K \leq \mathrm{GL}_n(\mathbb{F}_q)$ be two groups that satisfy some reasonable necessary conditions. Is it possible to construct two self-dual stabilizer codes $\mathcal{C}$ and $\mathcal{C}'$ such that $H = \mathrm{rMon}(\mathcal{C},\ \mathcal{C}')$ and $K = \mathrm{Symp}(\mathcal{C},\ \mathcal{C}')$?

Here, with G, G' generator matrices of $\mathcal{C}, \mathcal{C}'$,

$$\mathrm{rMon}(\mathcal{C},\mathcal{C}') = \{B \in \mathrm{GL}_k(\mathbb{F}_q) : \mathsf{G}M|_{\mathcal{C}} = B\mathsf{G}' \text{ for an } \mathrm{SL}_2(\mathbb{F}_q)\text{-monomial map } M\},$$
$$\mathrm{Symp}(\mathcal{C},\mathcal{C}') = \{B \in \mathrm{GL}_k(\mathbb{F}_q) : \mathrm{wt}_s(x\mathsf{G}) = \mathrm{wt}_s(xB\mathsf{G}') \text{ for all } x \in \mathbb{F}_q^k\}.$$

(The letters H, K in the quotation denote groups; everywhere else in this paper H is a graph.) For self-dual codes $k = n$; taking $\mathcal{C} = C_G$ and $\mathcal{C}' = C_H$ for graphs G, H on n vertices, an element of $\mathrm{Symp}(C_G, C_H)$ is exactly an invertible weight isometry $C_G \to C_H$ in the sense above, and $\mathrm{rMon}(C_G, C_H)$ is nonempty exactly when $G \approx H$.

The same theme is Question 7.4 of Gluesing-Luerssen and Pllaha [4], again verbatim: *"For a self-orthogonal submodule $C \subseteq R^{2n}$, how different can* $\mathrm{Mon}_{SL}(C)$ *and* $\mathrm{Symp}(C)$ *be?"* They write: *"In our situation, it is difficult to bring into play the self-orthogonality and therefore Question 7.4 is wide open."* Their Question 7.6 asks the same of $\mathrm{Mon}_{SL}(C^\perp, C) = \{f \in \mathrm{Mon}_{SL}(C^\perp) : f(C) = C\}$ and $\mathrm{Symp}(C^\perp, C) = \{f \in \mathrm{Symp}(C^\perp) : f(C) = C\}$, and they close with *"Note that for self-dual codes, Questions 7.6 and 7.4 are identical."* Both of these are questions about the isometries of a *single* code; Problem 6.3 is the two-code form, and it is the one this paper touches.

Problem 6.3 has two halves, and we answer only the first. The *emptiness* half asks whether $\mathrm{Symp}(\mathcal{C}, \mathcal{C}')$ can be nonempty while $\mathrm{rMon}(\mathcal{C}, \mathcal{C}')$ is empty — equivalently, whether a weight isometry can exist between graph codes of inequivalent graphs. The *group-realisation* half asks which pairs (H, K) of subgroups of $\mathrm{GL}_n(\mathbb{F}_q)$ arise as $(\mathrm{rMon}(\mathcal{C}, \mathcal{C}'), \mathrm{Symp}(\mathcal{C}, \mathcal{C}'))$. Everything below concerns the emptiness half, on an explicit range of lengths. Nothing below says anything about the group-realisation half, and nothing below says how much larger $\mathrm{Symp}(\mathcal{C})$ can be than $\mathrm{Mon}_{SL}(\mathcal{C})$ for a *single* code: our statements have content only for *pairs* of inequivalent codes, and the single-code comparison is exactly the setting of [4, Ex. 7.3].

What is settled here. Write $W(n)$ for the number of classes of the equivalence relation on n-vertex graphs generated by $\approx$ together with the existence of a weight isometry, and $E(n)$ for the number of distinct weight enumerators occurring among the $t(n)$ classes. Since equivalent graphs are weight-isometric and weight-isometric graph codes have equal weight enumerators,

$$E(n) \;\leq\; W(n) \;\leq\; t(n),$$

and Problem 6.3's emptiness half asks whether the right inequality is ever strict.

- **Length six** (Theorem 4.1, Corollary 4.2). For *all* 2^{15} graphs G and all 2^{15} graphs H on six vertices: if any $\mathbb{F}_2$-linear $g\colon \mathbb{F}_2^6 \to \mathbb{F}_2^6$ satisfies $w_H(gx) = w_G(x)$ for every x, then $G \approx H$. Hence $W(6) = t(6) = 26$. No invertibility of g is assumed anywhere; the search quantifies over all $\mathbb{F}_2$-linear maps, which makes the negative statement strictly stronger than $\mathrm{Symp}(\mathcal{C}, \mathcal{C}') = \emptyset$.

- **Length seven** (Theorem 5.1). No two of the 59 class representatives on seven vertices admit an $\mathbb{F}_2$-linear weight isometry. Granting Danielsen and Parker's $t(7) = 59$ [2, Table 2] and the easy direction, this is $W(7) = 59$ (Corollary 5.2).
- **The enumerator is strictly weaker** (Proposition 6.1). Only $E(6) = 23$ distinct weight enumerators occur among the 26 classes at length six, and the star $K_{1,3}$ and the two disjoint edges, padded to six vertices, are an inequivalent pair sharing one. This is what makes Theorem 4.1 a statement about the weight *function* up to $\mathrm{GL}(6, 2)$ and not a restatement of an enumerator computation.
- **A correction** (Theorem 8.1). Example 5.10 of the e-print [7] displays two length-four self-dual stabilizer codes and asserts that no monomial map carries one onto the other. Exactly 32 of the $6^4 \cdot 4! = 31\,104$ monomial maps do. What survives is the weaker true statement that the *displayed* weight-preserving map extends to no monomial map — 0 of the 31 104.

The last two items are what the first two are worth reading against. Example 5.10 is, so far as we have been able to determine, the only pair in print claimed to be weight-isometric and monomially inequivalent at the self-dual locus; with it withdrawn we are aware of no such pair at any length, and Theorems 4.1 and 5.1 say that none exists at lengths six and seven. The correction also isolates the phenomenon precisely: at length four a weight-preserving map between two self-dual codes may fail to extend to a monomial map even though the codes themselves are monomially equivalent. The map-level failure is real; the pair-level failure, which is what Problem 6.3's emptiness half asks about, does not occur below length twelve (Sections 4, 5 and 9, the last outside the formal development).

What rests on what. All statements below are finite, and all are decided by exhaustive computation over an explicitly bounded space; Section 10 records which trusted step each one uses. Three things are cited rather than proved here, and are flagged where they occur: Danielsen and Parker's class counts $t(n)$ [2, Table 2] and enumerator counts $E(n)$ [2, Table 7], used to *read* our computations as statements about $W(n)$ at $n \geq 7$ but never used inside a proof at $n = 6$, where the classification is reproduced from scratch; and Theorem 11 of [2], the easy direction, which at $n = 6$ is likewise replaced by an explicit machine-checked isometry for each of the 2^{15} graphs, and at $n = 7$ is cited. Section 9 reports the extension of the search to $n \leq 11$; it is a C computation against Danielsen's published orbit database [3] and is labelled throughout as lying outside the formal development.

2. Preliminaries

Graphs as integers. An n-vertex graph is encoded as an integer $c < 2^{\binom{n}{2}}$: bit $\pi(i, j) = i(n-1) - \binom{i}{2} + (j - i - 1)$ of c is the edge $\{i, j\}$ for $i < j$, the pairs taken in lexicographic order. Thus at $n = 6$ the graphs are the integers $c < 32768$; $c = 0$ is the empty graph, $c = 1$ the single edge $\{0, 1\}$, $c = 7$ the star with centre 0 and leaves $1, 2, 3$, and $c = 36$ the pair of disjoint edges $\{0, 3\}$ and $\{1, 2\}$, all three of the last on six vertices with the remaining vertices isolated. The encoding is a bijection between integers below $2^{\binom{n}{2}}$ and labelled graphs, so the moves below act on labelled graphs; this is one of the controls of Proposition 7.1.

The moves. For n fixed there are $2n - 1$ generators: local complementation at each of the n vertices, and the $n - 1$ transpositions of adjacent vertices. The relation $G \approx H$ is the existence of a finite word in these generators carrying G to H. Adjacent transpositions generate the symmetric group, so this is the same relation as local complementation together with arbitrary vertex permutations.

Linear maps and weight isometries. An $\mathbb{F}_2$-linear map $g \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is presented by the list $(v_0, \ldots, v_{n-1})$ of the images of the standard basis vectors, each $v_i < 2^n$ read as an element of $\mathbb{F}_2^n$ through its binary digits; then $g(x) = \bigoplus_{i \in \mathrm{supp}(x)} v_i$. We write $C_G \sim_w C_H$ when some such g satisfies $w_H(gx) = w_G(x)$ for all $x \in \mathbb{F}_2^n$. Three elementary facts are used freely.

Remark 2.1. (i) A weight isometry is injective, because $w_G(x) = 0$ only for $x = 0$; a finite self-map that is injective is bijective, so a weight isometry lies in $\mathrm{GL}(n, 2)$ and $\sim_w$ is symmetric. Quantifying over all $\mathbb{F}_2$-linear maps rather than over $\mathrm{GL}(n, 2)$ therefore costs nothing and makes every negative result stronger. (ii) Weight isometries compose, so $\sim_w$ is an equivalence relation, coarser than $\approx$. (iii) $C_G \sim_w C_H$ implies that C_G and C_H have the same weight enumerator. Only (ii) is used inside the formal proofs below — in the reduction of an arbitrary pair to a pair of class representatives. (i) and (iii) are used only to read the statements and to organise the computation of Section 9; in particular the searches at $n = 6$ and $n = 7$ run on *every* pair of representatives, with no enumerator prefilter, so nothing in Sections 4–5 depends on (iii).

The three counts. Recall $t(n)$, $E(n)$ and $W(n)$ from Section 1. The values relevant below are

n	4	5	6	7	8	9	10
$t(n)$ [2, Table 2]	6	11	26	59	182	675	3990
$E(n)$ [2, Table 7]	5	10	23	46	116	320	909

with $t(11) = 45144$, $t(12) = 1323363$, $E(11) = 3312$ and $E(12) = 15537$. So inequivalent classes begin to share a weight enumerator at $n = 4$; the question is whether any of those coincidences is ever upgraded to a weight isometry.

3. The classification at length six

The first ingredient is a machine-checked classification of the 2^{15} labelled graphs on six vertices. The count is not new — it is $t(6) = 26$ of [2, Table 2] — but the proof of Theorem 4.1 uses the classification itself, not the number, so it is reproduced rather than cited.

Theorem 3.1 (the classification at $n = 6$; the count is [2, Table 2])**.** *Let*

$$R_6 = \{\ 0,\ 1,\ 3,\ 7,\ 15,\ 31,\ 36,\ 37,\ 44,\ 45,\ 60,\ 61,\ 106,\\ 120,\ 121,\ 122,\ 246,\ 656,\ 657,\ 659,\ 684,\ 691,\ 692,\ 700,\ 1880,\ 7071\ \}$$

in the encoding of Section 2. Then $|R_6| = 26$*; every one of the* 32768 *labelled graphs on six vertices is carried to a member of* R_6 *by a word of local complementations and adjacent transpositions; and no two distinct members of* R_6 *are so related. Consequently there are exactly* 26 *classes, which is* $t(6)$ *of* [2, Table 2] *and* [6, A094927].

Proof (exhaustive computation). The certificate consists of three arrays indexed by the 32768 graphs: rep, the proposed class representative; dst, a proposed distance to it; and mv, a generator realising one step of a descent. The following is checked for every one of the 32768 graphs c: that rep$[c]$ is in range and is its own representative; that for each of the 11 generators m, applying m to c stays in range and does not change rep; and that either dst$[c] = 0$ and rep$[c] = c$, or $m =$ mv$[c]$ is a generator with dst$[m(c)] + 1 =$ dst$[c]$ *and* $m(m(c)) = c$. That is $32768 \cdot 11$ generator applications plus 32768 descent steps.

The second condition gives, by induction on the length of a word, that rep is constant on classes, hence that distinct members of R_6 are inequivalent once one knows they are their own representatives (checked, 26 tests) and that every rep value lies in R_6 (checked, 32768 tests). The third condition gives, by induction on dst$[c]$, a word carrying c to rep$[c]$ *and* a word carrying rep$[c]$ back to c; the back-step clause $m(m(c)) = c$ is what supplies the second word, so that no involutivity of the generators is needed in general and symmetry of $\approx$ is never invoked. That the 26 listed integers are pairwise distinct is a separate finite check. The certificate arrays were produced outside the proof, by a C program; nothing about how they were found is trusted, since all three conditions are re-verified from scratch. ☐

4. Length six: the weight function is a complete invariant

Theorem 4.1. *Let G and H be graphs on six vertices and suppose there is an $\mathbb{F}_2$-linear map $g\colon \mathbb{F}_2^6 \to \mathbb{F}_2^6$ with*

$$w_H(gx) = w_G(x) \qquad \textit{for all } x \in \mathbb{F}_2^6.$$

Then H is obtained from G by a finite sequence of local complementations and transpositions of adjacent vertices. No invertibility of g is assumed.

Corollary 4.2. *For all graphs G, H on six vertices, $C_G \sim_w C_H$ if and only if $G \approx H$. Consequently $W(6) = t(6) = 26$: in Pllaha's notation, for self-dual stabilizer codes of length six,* $\mathrm{Symp}(\mathcal{C}, \mathcal{C}') \neq \emptyset$ *implies* $\mathrm{rMon}(\mathcal{C}, \mathcal{C}') \neq \emptyset$.

The proof splits into an easy direction carried by explicit data, a hard direction carried by an exhaustive search whose completeness is proved rather than assumed, and a reduction gluing them.

The easy direction, as data. Local complementation acts on a graph code by a monomial map [2, Thm. 11] and vertex permutation by a permutation of coordinates, so both preserve weights. Rather than cite this, the development stores, for each of the 32768 graphs c, the six basis images of an explicit $\mathbb{F}_2$-linear weight isometry $C_c \to C_{\mathrm{rep}[c]}$ and the six basis images of one in the opposite direction, and checks both on all $2^6 = 64$ values of x: that is $32768 \cdot 64 \cdot 2$ weight comparisons. The stored maps were built by composing, along the descent tree of Theorem 3.1, the two generator-level maps

$$x \longmapsto x \oplus \big(\langle a, x\rangle \cdot e_u\big) \quad (a = \Gamma_G e_u) \qquad \text{and} \qquad x \longmapsto \text{the transposition of bits } p, q,$$

the first for local complementation at u, the second for the transposition (p, q). For the first: the local Clifford is $\sqrt{X}$ at u and $\sqrt{Z}$ at each neighbour, which on codewords is $(x_i, z_i) \mapsto (x_i, z_i + x_i)$ for i a neighbour of u and $(x_u, z_u) \mapsto (x_u + z_u, z_u)$; chasing this through $z = \Gamma_G x$ gives $\tilde{x} = (I + e_u a^\mathsf{T})x$ and $\tilde{z} = (\Gamma_G + \operatorname{diag} a)x$, which is $\Gamma_{G\star u}\tilde{x}$ because $\Gamma_{G\star u} = \Gamma_G + aa^\mathsf{T} + \operatorname{diag}(a)$ and $a^\mathsf{T} e_u = 0$. Each of the three coordinatewise maps involved is an $\mathbb{F}_2$-linear bijection of $\mathbb{F}_2^2$, so it preserves the property "this coordinate is nonzero": not merely the weight but the union of supports is preserved. This paragraph explains how the data was produced; the proof uses only the checked fact that the stored maps are weight isometries.

The hard direction: the search, and why a negative answer means what it says. Fix two graphs G, H on n vertices and their weight vectors $u_G, u_H \in \mathbb{N}^{2^n}$, $u_G[x] = w_G(x)$. The search chooses the images $v_0, v_1, \ldots, v_{n-1}$ of the standard basis in turn, each from the 2^n candidates, and prunes: having chosen a prefix $v_0, \ldots, v_k$, it tests the weight condition $u_H[g(x)] = u_G[x]$ only on the indices $x \in [2^k, 2^{k+1})$ that are new at that prefix, the lower indices having been tested at shorter prefixes. It returns `true` as soon as a full list passes and `false` when the tree is exhausted.

Lemma 4.3 (completeness of the search)**.** *If some list $(v_0, \ldots, v_{n-1})$ of candidates satisfies the weight condition on all of $[0, 2^n)$, then the search returns* `true`*. Consequently, if the search returns* `false`*, there is no $\mathbb{F}_2$-linear map at all — invertible or not — carrying w_G to w_H.*

Proof (a proof, not a computation). Induction on the list of images still to be chosen. The step needs that the prefix test is sound: for $x < 2^k$ the map presented by the whole list agrees with the map presented by its first k images, because the later images are multiplied by bits of x that vanish. Hence every prefix of a full solution passes its incremental test and the depth-first traversal reaches the solution. The lemma, and the elementary algebra of the presentation of linear maps it rests on (additivity, composition, truncation), are proved in the formal development and involve no computation. □

The pruning is not merely an optimisation. At $n = 6$, testing the whole span at every node costs 144.5 million elementary comparisons against 18.7 million for the incremental version, a factor of 7.7 for the same answer; this was measured in an instrumented C copy of the same traversal before the search was written.

The reduction. Let g be a weight isometry $C_G \to C_H$. Let A be the stored isometry $C_{\mathrm{rep}[G]} \to C_G$ and B the stored isometry $C_H \to C_{\mathrm{rep}[H]}$. Then $B \circ g \circ A$ is a weight isometry between the codes of the two representatives. The search, run on all $26 \cdot 25 = 650$ ordered pairs of distinct members of R_6, returns `false` on every one; so by Lemma 4.3 the representatives coincide, and Theorem 3.1 joins G to $\mathrm{rep}[G] = \mathrm{rep}[H]$ and back down to H. This is why the certificate stores both directions of the isometry and both directions of the descent. Corollary 4.2 is Theorem 4.1 together with the easy direction, which at this length is the stored data again.

Proof of Theorem 4.1 and Corollary 4.2 (exhaustive computation). Four finite checks, all run: the classification certificate of Theorem 3.1 ($32768 \cdot 11$ generator applications and 32768 descent steps); the isometry certificate ($32768 \cdot 128$ weight comparisons); that every representative value lies in R_6 and that each member of R_6 is its own representative; and the 650 searches, together 6.7 million nodes of depth-first traversal, each node a choice among 64 candidate basis images. The reduction and Lemma 4.3 are proofs and are not computed. $\square$

5. Length seven

Theorem 5.1. *Let R_7 be the following* 59 *integers, in the encoding of Section 2 with $n = 7$:*

0, 1, 3, 7, 15, 31, 63, 68, 69, 76, 77, 92, 93, 124, 125, 202, 216, 217, 218, 248,
249, 250, 470, 498, 502, 2320, 2321, 2323, 2352, 2353, 2355, 2380, 2387, 2388, 2396,
2416, 2417, 2419, 2420, 2428, 2856, 2858, 2922, 2924, 6808, 6820, 6821, 6824, 6825,
6840, 15796, 42096, 42100, 42232, 43807, 43814, 43839, 44280, 44336.

They are 59 *pairwise distinct integers, and for any two distinct $r, r' \in R_7$ there is no $\mathbb{F}_2$-linear $g\colon \mathbb{F}_2^7 \to \mathbb{F}_2^7$ with $w_H(gx) = w_G(x)$ for all $x \in \mathbb{F}_2^7$, where G and H are the graphs encoded by r and r'.*

Proof (exhaustive computation). The search of Section 4 is run on all $59 \cdot 58 = 3422$ ordered pairs of distinct members of R_7, a total of 168 million depth-first nodes (83.8 million for the unordered half), each node a choice among 128 candidate basis images; every run returns `false`, and Lemma 4.3 converts each `false` into the nonexistence of a linear map. The run is split into eight blocks by source representative and recombined, so that a slow machine loses at most one block rather than the whole run. That the 59 integers are pairwise distinct is a separate finite check. No stored data beyond the 59 integers enters the statement or its proof. $\square$

Corollary 5.2. *Granting $t(7) = 59$* [2, Table 2] *— so that R_7 is a complete set of class representatives — and the easy direction,* [2, Thm. 11] *for local complementation and the invariance of weights under coordinate permutations for the rest, so that weight isometry is a relation between classes — Theorem 5.1 says $W(7) = 59$: at length seven, weight isometry merges no two classes.*

The two inputs to Corollary 5.2 are cited, not proved here, and this is the one place where the paper's conclusions at $n = 7$ rest on the literature. Both were reproduced outside the formal development: the 59 integers are the least member of each orbit, computed by an orbit closure over all 2^{21} labelled graphs on seven vertices (11 seconds), whose class count is 59, and independently by assembling Danielsen's published *connected* representatives into disjoint unions (Section 9). Repeating the certificate of Theorem 3.1 at $n = 7$ would need about 13 megabytes of representative, distance and move data and roughly $2 \cdot 10^{10}$ evaluation steps, and was left undone; that is the precise gap between Theorem 4.1 and Theorem 5.1.

6. The weight enumerator is strictly weaker

Theorem 4.1 would be uninteresting if the weight enumerator already separated the classes, since the enumerator is a coarser invariant that any weight isometry preserves (Remark 2.1(iii)). It does not.

Proposition 6.1 ($E(6) = 23 < 26 = t(6)$; the count is [2, Table 7])**.** *Among the* 26 *classes at length six, exactly* 23 *distinct weight enumerators occur — the value $E(6)$ of* [2, Table 7]*. Explicitly, the star $K_{1,3}$ with centre* 0 *and leaves* 1, 2, 3 *(encoded* 7*) and the two disjoint edges* $\{0, 3\}, \{1, 2\}$ *(encoded* 36*), both with two isolated vertices, have the same weight enumerator*

$$1 + 2z + 7z^2 + 12z^3 + 15z^4 + 18z^5 + 9z^6,$$

and are inequivalent.

Proof (exhaustive computation). The enumerator of each of the 26 representatives is computed over all $2^6 = 64$ values of x and the distinct values counted, giving 23. The displayed enumerator is computed for the two named graphs; their inequivalence is Theorem 3.1, both being members of R_6. This is the classical $n = 4$ coincidence — the $[[4, 0]]$ codes of the star and of two Bell pairs — padded out to six vertices. □

So three inequivalent pairs at length six share an enumerator, and sixteen do at length seven — those two pair counts come from the C computation of Section 9 and not from the formal development, which counts enumerators and not collisions — and Theorems 4.1 and 5.1 say that none of these coincidences is upgraded to a weight isometry. That is the content of the results: the weight *function* up to $\mathrm{GL}(n, 2)$ separates, at these lengths, exactly what the weight enumerator cannot.

7. Controls

An exhaustive search that never succeeds, or a model that computes the wrong quantity, proves nothing.

Proposition 7.1 (controls)**.** *All of the following are machine-checked.* The model computes the intended quantity*: the weight function of the empty graph on six vertices is the Hamming weight, checked on all* 64 *inputs; the graph encoded by* 1 *has neighbourhood masks* $\{1\}, \{0\}, \emptyset, \ldots,$ *i.e. it is the single edge* $\{0, 1\}$*; and its weight function takes the value* 2 *at* e_0*,* e_1 *and* $e_0 + e_1$ *and the value* 1 *at* e_2*.* The encoding is a bijection*: decoding and re-encoding is the identity on all* 32768 *six-vertex codes, so the generators act on labelled graphs and no two graphs are conflated.* A too-small claim is refuted*:* $\approx$ *is not the total relation — the empty graph and a single edge on six vertices are inequivalent.* The hypothesis is satisfiable*: two* distinct *labelled graphs on six vertices are weight-isometric, so Theorem 4.1 is not vacuous on its hypothesis.* The search is not broken*: run with a graph against itself, the search of Section 4 returns* `true`*, so a* `false` *carries information. The too-large control — that the weight enumerator alone does not suffice — is Proposition 6.1.*

The weight model was also checked against the literature from outside: the enumerator counts $E(n)$ produced by the same definition reproduce [2, Table 7] exactly for $n = 1, \ldots, 11$, and a wrong weight function would not.

8. A correction to Pllaha's Example 5.10

Which text is corrected. Everything in this section refers to the e-print arXiv:1807.09107v1, which is the only arXiv version of [7] and the only text of it we have read; "Example 5.10", "Theorem 5.8" and "Problem 6.3" are its numbers. The published article (J. Algebra Appl. **19** (2020), 2050021) is not open access and we have not seen it; the abstract deposited by its publisher already differs in wording from the e-print's, so the article was copy-edited at least, and we make no claim about its wording or its numbering. What we can say is that the example survived unchanged into Pllaha's dissertation [8] of January 2019, where it appears as Example 7.55 with the same two matrices and the same sentence quoted below. Mahmoud likewise scopes his remark on it to *"the arXiv version"* [5, §1.4].

Example 5.10 of [7] displays the two length-four self-dual stabilizer codes generated by

$$S = \langle XZXX,\ ZXIX,\ ZIZI,\ ZZIZ\rangle, \qquad S' = \langle YXXY,\ IZXX,\ IIZZ,\ ZIXX\rangle$$

together with the map f sending the i-th displayed generator of S to the i-th of S', and asserts, verbatim:

> The map $f : \mathcal{C} \longrightarrow \mathcal{C}'$ that maps the i-th row of G to the i-th row of G' is a symplectic isometry and thus $\mathcal{C}$ and $\mathcal{C}'$ are symplectially equivalent. On the other hand, it is easy to see that there cannot exist a $\mathrm{SL}_2(\mathbb{F}_2)$-monomial map between the two.

and concludes that $Q(S)$ and $Q(S')$ are not LCP equivalent, in fact not even LU equivalent. The second sentence of the quotation is false.

Theorem 8.1. *With S and S' as displayed:*

(i) both codes are self-dual of length four — hence 4*-qubit stabilizer states, and by* [2, Thm. 6] *equivalent to graph codes — and both have* 16 *elements;*
(ii) they have the same weight enumerator, $1 + 2z^2 + 8z^3 + 5z^4$*;*
(iii) Pllaha's map f is weight-preserving;

(iv) S and S' are monomially equivalent. *A witness is displayed: transpose qubits* 1 *and* 2*, then apply* $X \mapsto X, Z \mapsto Z, Y \mapsto Y$ *on qubit* 0*,* $X \mapsto Y, Z \mapsto Z, Y \mapsto X$ *on qubit* 1*, and* $X \mapsto X, Z \mapsto Y, Y \mapsto Z$ *on qubits* 2 *and* 3*;*

(v) *exactly* 32 *of the* $6^4 \cdot 4! = 31\,104$ *monomial maps on four qubits carry* S *onto* S'*, and exactly* 0 *of them realise* f*.*

By Theorem 5.8 of [7] *itself, (iv) says that* $Q(S)$ *and* $Q(S')$ *are LCP equivalent. What survives of Example 5.10 is (v)'s second half: the displayed map* f *extends to no monomial map, which is the phenomenon of* [4, Ex. 7.3].

Proof (kernel computation for (i)–(iv), exhaustive computation for (v)). Pauli letters are elements of $\mathbb{F}_2^2$ with multiplication modulo phase given by addition, so each code is the $\mathbb{F}_2$-span of its four displayed generators, 16 words in all. Items (i)–(iv) are decided directly on those 16-element sets: self-duality is $16 \cdot 16$ evaluations of the trace-symplectic form per code, (ii) and (iii) are weight counts over the 16 words, and (iv) applies the displayed monomial map to each of the 16 words of S and compares the resulting set with S'. Item (v) enumerates all 24 qubit permutations against all 6^4 choices of an element of $\mathrm{GL}(2, 2) \cong S_3$ per qubit, 31 104 maps, applying each to all 16 words; it finds 32 carrying S onto S' and none realising f on the nose. □

Remark 8.2 (on the status of this correction)*.* The mathematics of Theorem 8.1(iv)–(v) is not new. Mahmoud [5, §1.4] reports the same count of 32 in prose, hedged: *"If this reading is correct … We would welcome correction on this point."* The contribution here is a machine-checked settlement of that hedge — the witness of (iv) is verified by the proof kernel with no appeal to compiled evaluation — and a third independent implementation, written for this paper, agreeing on both 32 and 0. The generators above were transcribed letter by letter from the LaTeX source of arXiv:1807.09107v1 and cross-checked against a text extraction of the compiled e-print, on 30 August 2026. We make no claim about the effect of the correction on any other statement of [7]: Example 5.10 is an illustration, and we have not audited what depends on it.

Remark 8.3 (the single-code form of the same phenomenon)*.* Example 7.3 of [4] displays the length-four self-dual code C generated by either of two matrices G_1, G_2 and the map sending the i-th row of G_1 to the i-th row of G_2, and shows that this map preserves the symplectic weight but extends to no $\mathrm{SL}_2(\mathbb{F}_2)$-monomial map. That G_1 is, entry by entry, the matrix G of Example 5.10 — so that C is the code S above — was read off the two e-prints, and we checked outside the formal development that G_1 and G_2 do generate one and the same self-dual code, whose displayed self-map is weight-preserving. So [4, Ex. 7.3] is the automorphism ($\mathcal{C} = \mathcal{C}'$) form of Theorem 8.1(v)'s second half, on the same code; none of this enters the formal development, and it is used only as an attribution.

Remark 8.4 (which state it is)*.* Both codes of Example 5.10 are monomially equivalent to the graph code of the path $3 - 0 - 1 - 2$, that is, to the four-qubit linear cluster state. This was checked outside the formal development, by running the monomial-equivalence search of Theorem 8.1 against the graph codes of all six classes at $n = 4$: exactly one matches, and it matches both codes. It also serves as a cross-check that the graph-code model of Section 2 and the Pauli-word model of this section describe the same objects.

9. Lengths eight to eleven, outside the formal development

Remark 9.1 (computed in C, not formalised)*.* The same search, implemented in C and run against Danielsen's published orbit database [3], gives $W(n) = t(n)$ for every $n \leq 11$. We record it here as a computation outside the formal development; only $n \leq 7$ is machine-checked, in Theorems 4.1 and 5.1.

The database files hold the *connected* orbit representatives — that is, the indecomposable codes, the sequence $1, 1, 1, 2, 4, 11, 26, 101, 440, 3132, 40457$ of [6, A090899] — and not the full class list; local complementation preserves connected components, so a class of n-vertex graphs is a multiset of connected classes with orders summing to n, and the full representative list is reassembled from the database by disjoint unions. That the reassembly reproduces both $t(n)$ [2, Table 2] and $E(n)$ [2, Table 7] exactly for every n in range is the check that the step is right. Two programs sharing no code were run: one building the classes by orbit closure over all $2^{\binom{n}{2}}$ labelled graphs (reaching $n \leq 7$), one by the disjoint-union reassembly (reaching $n \leq 11$); they agree on the whole overlap $n = 4, \ldots, 7$ and both reproduce the two published tables.

The pair search is then run inside weight-enumerator buckets, which is sound by Remark 2.1(iii): on $1, 1, 3, 16, 94, 892, 31\,263$ and $4\,501\,923$ pairs at $n = 4, \ldots, 11$ respectively. Every run returns "no isometry"; the positive control, a class against itself, returns "isometry found" at every length. The cost is 6.9 CPU-seconds for $n \leq 10$ and a further 425 CPU-seconds at 96 megabytes for $n = 11$ ($2.53 \cdot 10^9$ search nodes). Length 12 was priced and not attempted: it needs $1\,323\,363 \cdot 4096 \approx 5.4$ gigabytes of weight vectors and, extrapolating the collision count, of order 10^9 colliding pairs, i.e. tens of CPU-hours.

10. Verification

Every statement above called a theorem, proposition, lemma or corollary has been formally verified in Lean 4 [10] (toolchain `v4.32.0`), with the exceptions labelled in place as computed outside it: Corollary 5.2, whose two inputs are cited from [2]; the reproduction of the 59 representatives by orbit closure quoted after it; the identification of the two codes of Example 5.10 with the linear cluster state (Remark 8.4); the reading of [4, Ex. 7.3] in Remark 8.3; the $E(n)$ agreement quoted after Proposition 7.1; the two enumerator-collision counts quoted after Proposition 6.1; and all of Section 9. The repository is private; repository reference to be added at submission. The development contains no `sorry`, and its modules import neither *Mathlib* [11] nor any other library — every object is a natural number, a list or an array, and the definitions of Section 2 are transcribed directly, so that "no weight isometry exists" is decided by quantifying over $\mathbb{F}_2$-linear maps themselves rather than through any criterion. The reasoning layer — Lemma 4.3, the algebra of linear maps, the two soundness arguments turning a certificate into mathematics, and the reduction of Section 4 — is checked by the proof kernel and depends only on `propext` and `Quot.sound`; parts (i)–(iv) of Theorem 8.1, including the witness (iv), and the distinctness of the 59 integers of Theorem 5.1 are decided by the kernel as well — the former on `propext` alone, the latter on no axioms whatever. The large finite checks — the classification and isometry certificates, the 650 searches at $n = 6$, the 3422 at $n = 7$, the enumerator count and the 31 104-map enumeration — are carried out by Lean's compiled evaluator (`native_decide`), so those statements additionally depend on one evaluation axiom per check and on nothing else: no `sorryAx`, no `Classical.choice`. Trusting the compiled evaluator is the price of the searches; it is the only trusted step, and the computations most exposed to it were re-run outside Lean by programs sharing no code, as Sections 5, 8 and 9 record. The verification costs about 20 CPU-minutes at 1.6 gigabytes of peak memory: 92 CPU-seconds for the length-six checks over 554 kilobytes of certificate data, 1029 CPU-seconds for the eight length-seven blocks, and about 25 CPU-seconds for the remaining modules.

References

[1] A. Bouchet, *Graphic presentations of isotropic systems*, J. Combin. Theory Ser. B **45** (1988), 58–76; `doi:10.1016/0095-8956(88)90055-X`.

[2] L. E. Danielsen, M. G. Parker, *On the classification of all self-dual additive codes over $GF(4)$ of length up to 12*, J. Combin. Theory Ser. A **113** (2006), no. 7, 1351–1367; `doi:10.1016/j.jcta.2005.12.004`; arXiv:math/0504522v4. The e-print has four versions; v4 (17 February 2006) is the one read here, and the internal numbers cited above — Theorem 6 (every self-dual additive $GF(4)$ code is equivalent to a graph code), Theorem 11 (local complementation), Theorem 12 (Bouchet's criterion), Table 2 (t_n) and Table 7 (distinct weight enumerators, row "All") — are its. We did not consult the journal version and do not assert that its numbering agrees.

[3] L. E. Danielsen, *Database of Self-Dual Quantum Codes*, the local-complementation orbit data accompanying [2], hosted at `codetables.de/larsed/vncorbits/`. The files `vncorbits`k`.g6` hold the connected representatives only: *"These files contain only connected graphs (corresponding to indecomposable codes)"*, as the page states. The host site is M. Grassl's *Bounds on the minimum distance of linear codes and quantum codes*, `https://www.codetables.de`.

[4] H. Gluesing-Luerssen, T. Pllaha, *On quantum stabilizer codes derived from local Frobenius rings*, Finite Fields Appl. **58** (2019), 145–173; `doi:10.1016/j.ffa.2019.04.004`; arXiv:1710.09884v1. Example 7.3, Question 7.4, the sentence calling it wide open, and Question 7.6 are numbered as in the e-print, which is the only arXiv version and the only text of this paper we have read; we did not consult the journal version and do not assert that its numbering agrees.

[5] A. A. Mahmoud, *Minimal Counterexamples of the MacWilliams Extension Theorem for Stabilizer Codes*, arXiv:2607.26214v1 (quant-ph). Section 1.4, *Related work and attribution*, is the remark on Example 5.10 of [7]; v1 was the only version as of 30 August 2026.

[6] OEIS Foundation Inc., *The On-Line Encyclopedia of Integer Sequences*, `https://oeis.org`; entries A094927 (nonisomorphic, possibly decomposable, self-dual quantum codes on n qubits) and A090899 (the indecomposable count).

[7] T. Pllaha, *Symplectic isometries of stabilizer codes*, J. Algebra Appl. **19** (2020), no. 2, 2050021; `doi:10.1142/S0219498820500218`; arXiv:1807.09107v1. Problem 6.3, Theorem 5.8 and Example 5.10 are numbered as in the e-print, which is the only arXiv version (as of 30 August 2026) and the only text of this paper we have read; the published article is not open access. See the opening of Section 8.

[8] T. Pllaha, *Equivalence of Classical and Quantum Codes*, Ph.D. dissertation, University of Kentucky, 2019 (approved 31 January 2019); `doi:10.13023/etd.2019.014`; `https://uknowledge.uky.edu/math_etds/59`. Example 7.55 there is Example 5.10 of [7].

[9] M. Van den Nest, J. Dehaene, B. De Moor, *Graphical description of the action of local Clifford transformations on graph states*, Phys. Rev. A **69** (2004), 022316; arXiv:quant-ph/0308151v2.

[10] L. de Moura, S. Ullrich, *The Lean 4 theorem prover and programming language*, in: A. Platzer and G. Sutcliffe (eds.), Automated Deduction — CADE 28, Lecture Notes in Computer Science **12699**, Springer, Cham, 2021, pp. 625–635; `doi:10.1007/978-3-030-79876-5_37`.

[11] The mathlib Community, *The Lean mathematical library*, in: Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381; `doi:10.1145/3372885.3373824`.