Polynomial-time decision of local-unitary equivalence for graph states

Yuxuan Zhang
Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Department of Physics, Princeton University
Princeton, New Jersey 08544, USA

22 September 2026
Working manuscript: AI-assisted proof candidate
Independent expert review and historical-priority verification pending

Abstract

Claudet and Perdrix gave a quasi-polynomial-time algorithm for local-unitary equivalence of labelled graph states. Its expensive step enumerates incidence constraints for generalized local complementations at a logarithmic level. We replace that enumeration by a polynomial-time module-closure computation. Over $\mathbb{Z}/2^r\mathbb{Z}$, all relevant weighted incidence rows are generated from pairs and triples under explicit diagonal endomorphisms. An integer-lattice implementation stabilizes after at most $r|X|$ strict enlargements, where X is the prescribed independent support. We construct the complete modular kernel and its binary edge-toggle image, retaining multiplicity witnesses. Combining this computation with the credited graph reductions gives a deterministic decision algorithm with polynomial bit complexity for arbitrary labelled graph states. The integration treats disconnected inputs, the connectedness and even-degree conditions of the constrained local-Clifford step, and the zero-image case. Exact finite diagnostics accompany the proofs; they do not constitute an implementation of the full graph-state algorithm.

1 Introduction

A graph state associates an n-qubit pure state with a finite simple graph. Its graph representation makes local-Clifford transformations amenable to discrete methods, and local-Clifford equivalence can be decided efficiently [3]. Local-unitary equivalence allows arbitrary single-qubit unitaries and is a different decision problem. Throughout this paper the vertices are labelled: the algorithm does not optimize over qubit permutations.

Claudet and Perdrix [1] reduced local-unitary equivalence to a sequence of graph and linear-algebra computations and obtained the bound $n^{\log_2 n+O(1)}$. Their standard-form reduction leaves one generalized local complementation, together with ordinary local complementations on a specified set of vertices. The generalized operation is described by linear congruences, but the direct description has a row for every relevant high-order subset. At level r, this creates $O(n^{r+1})$ rows. The level required for the general local-unitary question is logarithmic in n. Whether the general decision problem admits a polynomial-time algorithm is also explicitly posed in Claudet's thesis [2, Chapter 8].

The purpose of this paper is to remove that subset enumeration. The weighted rows have more structure than a generic exponentially indexed linear system: increasing the subset by a new vertex corresponds, except for the pair-to-triple transition, to applying a fixed diagonal

linear map. Pair and triple rows therefore generate the whole constraint module under closure. Since the ambient module has finite length $r|X|$, the closure stabilizes after polynomially many strict steps. A bounded Hermite-normal-form representation turns this observation into an algorithm with polynomial bit complexity.

Theorem 1.1 (Main result)**.** *There is a deterministic classical algorithm that, given two finite simple graphs G and H on the same labelled vertex set of size $n \geq 1$, decides in polynomial bit complexity whether their graph states are local-unitary equivalent, up to global phase. No connectivity, degree, or graph-class promise is required.*

The theorem uses the published standard-form, level, and constrained local-Clifford reductions of Claudet and Perdrix. We do not reprove those graph-theoretic results. Our contribution is the incidence-module compression and the explicit verification that it supplies the required complete toggle space. Section 5 records the source interfaces; Section 6 checks their applicability, including two cases that cannot be left implicit: the connectivity of the augmented graphs and the branch in which no nonzero toggle is possible.

The manuscript develops the proof candidate obtained in research round 21 from an earlier prescribed-support compression argument. The claim and frozen proof were publicly reported in QIQC issue 97. Historical AI reviews concern that frozen submission, not an independent human review of this manuscript. The present version supplies a conventional mathematical exposition and additional targeted exact checks. It remains a working manuscript pending expert scrutiny and a fuller priority review.

2 Problem and notation

For a finite simple graph $G = (V, E)$ with $V = [n]$, define

$$|G\rangle = \prod_{\{u,v\}\in E} \mathrm{CZ}_{uv} |+\rangle^{\otimes n}, \qquad |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{1}$$

Local-unitary equivalence means that

$$|H\rangle = e^{i\phi} \left(\bigotimes_{v=1}^{n} U_v \right) |G\rangle \quad \text{for some } U_v \in U(2),\ \phi \in \mathbb{R}. \tag{2}$$

The labels in (2) are fixed. A local complementation at v toggles every edge among distinct neighbors of v. We write $G \star v$ for the resulting graph and use LC for equivalence under sequences of these operations.

For a multiset S of vertices, $s(x)$ denotes its nonnegative integer multiplicity and $\mathrm{supp}(S) = \{x : s(x) > 0\}$. For $K \subseteq V$, let

$$c_S(K) = \sum_{x:\ K \subseteq N_G(x)} s(x) \tag{3}$$

be its common-neighbor multiplicity. The following conventions are those of [1, Definitions 2.1–2.2].

Definition 2.1. Let $r \geq 1$. An independent multiset S is r-incident if, for every $0 \leq j < r$ and every $K \subseteq V \setminus \mathrm{supp}(S)$ with $|K| = j + 2$,

$$c_S(K) \equiv 0 \pmod{2^{r-j-\delta_{j0}}}. \tag{4}$$

Its r-local complementation toggles a pair outside $\mathrm{supp}(S)$ precisely when its common-neighbor multiplicity is 2^{r-1} modulo 2^r. Pairs meeting the support are unchanged.

Let LC_r denote the local-equivalence relation generated by single-qubit Hadamards and $Z(\pi/2^r)$ rotations, in the convention of [1]. Thus $\mathrm{LC}_1 = \mathrm{LC}$, and every r-local complementation is a local-unitary transformation. We use the equivalence between the graph operations and these unitary relations as a credited result from [1, Section 2.3].

Our input graphs have adjacency-matrix encodings. Arithmetic complexity means bit complexity. The auxiliary parameter r is charged according to its value, not merely its binary encoding length. This is sufficient because the final algorithm chooses $r = O(\log n)$. The ring $R = \mathbb{Z}/q\mathbb{Z}$, where $q = 2^r$, is not a field when $r > 1$; its kernels need not be free modules.

3 Compression of the incidence module

We first state the algebraic result independently of graph standard forms. Let $a_1, \dots, a_t \in \{0,1\}^m$. For $K \subseteq [t]$, define the coordinatewise product

$$a_K = \bigodot_{v \in K} a_v, \qquad w(k) = \begin{cases} 1, & k = 2, \\ k-2, & k \geq 3. \end{cases} \tag{5}$$

The weighted incidence module is

$$M_r = \mathrm{span}_R\{2^{w(|K|)} a_K : 2 \leq |K| \leq r+1\} \subseteq R^m. \tag{6}$$

Subsets larger than t contribute no rows.

Theorem 3.1 (Polynomial incidence compression)**.** *A generating matrix for M_r with at most m rows can be constructed deterministically in time polynomial in m, t, r. The construction uses at most rm strict lattice enlargements, and its retained integer basis entries have $O(r+1)$ bits.*

3.1 The closure identity

For each $v \in [t]$, define the R-linear map

$$T_v(z) = 2a_v \odot z. \tag{7}$$

Let N_r be the smallest submodule containing $2a_K$ for $|K| = 2$ or 3 and invariant under all the maps T_v.

Lemma 3.2. $M_r = N_r$.

Proof. The pair seeds occur in (6). The triple seeds occur when $r \geq 2$; when $r = 1$, all seeds are zero modulo two. Thus every seed belongs to M_r.

We check invariance on a generating row $2^{w(k)} a_K$, where $k = |K|$. If $v \in K$, the binary coordinates satisfy $a_v \odot a_K = a_K$, so T_v gives twice that row. If $v \notin K$ and $k = 2$, the result is $4a_{K \cup \{v\}}$, twice the triple seed. For $r = 1$ this result is zero. If $v \notin K$ and $k \geq 3$, then

$$T_v(2^{k-2} a_K) = 2^{k-1} a_{K \cup \{v\}}, \tag{8}$$

which is the next prescribed row, unless $k = r+1$, in which case it vanishes modulo 2^r. Hence M_r is invariant and $N_r \subseteq M_r$.

Conversely, if $|K| = k \geq 4$, choose three vertices of K and start with their triple seed. Applying T_v for the remaining $k-3$ distinct vertices gives coefficient $2 \cdot 2^{k-3} = 2^{k-2}$. Every generator in (6) is therefore in N_r. Pair and triple generators were already seeds. This proves equality; in particular, repeated application of a vertex introduces no extraneous constraints. □

3.2 A finite-length algorithm with bounded representations

Proof of Theorem 3.1. The case $m = 0$ is immediate, so assume $m > 0$. Lift the seed rows to integers and let $L_0 \subseteq \mathbb{Z}^m$ be the lattice generated by those rows and $q\mathbb{Z}^m$. Extend (7) to the same integer diagonal map and iterate

$$L_{i+1} = L_i + \sum_{v=1}^{t} T_v(L_i). \tag{9}$$

After each step, retain a full-rank row Hermite basis. Applying T_v to a basis suffices to generate $T_v(L_i)$.

Every lattice in this chain contains $q\mathbb{Z}^m$ and lies in $\mathbb{Z}^m$. Thus $L_i/q\mathbb{Z}^m$ is a subgroup of a group of order $q^m = 2^{rm}$. A strict enlargement at least doubles its order. There can be at most rm strict enlargements. If equality occurs at one step, the lattice is invariant under every T_v, so no later step can enlarge it. Its image modulo q is exactly $N_r = M_r$ by Lemma 3.2.

The initial matrix has at most $m + \binom{t}{2} + \binom{t}{3}$ rows. Every subsequent matrix has at most $m(1+t)$ rows. A reduced triangular Hermite basis of a lattice containing $q\mathbb{Z}^m$ has each diagonal entry dividing q: reduce the vectors qe_j against the triangular basis successively, using the orientation corresponding to that basis. Off-diagonal entries are reduced below the corresponding positive pivots. Consequently every retained entry has magnitude at most q and has $O(r+1)$ bits. Applying T_v multiplies selected coordinates by two, preserving this bit bound up to one bit.

Deterministic Hermite-normal-form algorithms have polynomial bit complexity in their matrix dimensions and input bit lengths [4]. Polynomially many calls on the bounded matrices above establish the asserted bit complexity, including intermediate arithmetic inside each call. Reducing the final m basis rows modulo q gives a generating matrix B for M_r. Zero rows may be omitted. □

Algorithm 1: weighted incidence compression

1. Form the pair and triple rows $2a_K$ and the rows of qI_m.
2. Replace their integer row span by a canonical Hermite basis H.
3. Compute a Hermite basis H' of the rows of H and all $T_v(H)$.
4. If $H' = H$, return H mod q; otherwise set $H \leftarrow H'$ and repeat step 3.

For $m = 0$, return the empty matrix immediately.

Corollary 3.3. *For $s \in R^m$, the exponentially indexed conditions*

$$2^{w(|K|)} a_K \cdot s = 0 \pmod{q} \quad (2 \le |K| \le r+1) \tag{10}$$

hold if and only if $Bs = 0$ in R.

Proof. Annihilating generators of a row module is equivalent to annihilating the entire module. Apply Theorem 3.1. □

4 The complete toggle space and its witnesses

Fix a graph G, an independent set $X \subseteq V$, and $Y = V \setminus X$. Put $m = |X|$, $q = 2^r$, and $h = 2^{r-1}$. For each $v \in Y$, let $a_v \in \{0,1\}^X$ be its adjacency vector into X. Let B be the compressed matrix and

$$\Sigma = \ker_R B. \tag{11}$$

4.1 Matching the precise support convention

Lemma 4.1. *The vectors in Σ, represented by multiplicities in $\{0, \dots, q-1\}$ on X, are exactly the r-incident multisets supported in X.*

Proof. For $K \subseteq Y$, the common-neighbor multiplicity is $a_K \cdot s$. When $k = |K|$, condition (4) requires divisibility by $2^{r-w(k)}$, exactly equivalent to (10). This equivalence also holds for $r = 1$, when the pair condition is vacuous.

The actual support of a multiplicity vector may be smaller than X. If a set $K \subseteq V \setminus \mathrm{supp}(S)$ contains some $x \in X$, no vertex of X is adjacent to every vertex of K: independence of X forbids adjacency to x. Its common-neighbor multiplicity in S is therefore zero. All such extra conditions in Definition 2.1 hold automatically. These observations establish both directions. Reducing multiplicities modulo q preserves all relevant divisibilities. □

For every unordered pair $p \in \binom{Y}{2}$, define

$$f(s)_p = \frac{a_p \cdot \widetilde{s}}{h} \pmod 2, \qquad s \in \Sigma, \tag{12}$$

where $\widetilde{s}$ is any integer lift. Let $\Omega = f(\Sigma)$.

Lemma 4.2. *The map in (12) is well-defined and satisfies*

$$f(s+t) = f(s) + f(t), \qquad f(cs) = (c \bmod 2) f(s). \tag{13}$$

Consequently Ω is a binary linear subspace.

Proof. The pair constraint is $2a_p \cdot s = 0 \pmod q$, so every integer lift has $a_p \cdot \widetilde{s}$ divisible by h. For $r = 1$, this is automatic because $h = 1$. Changing the lift by qz changes the quotient by $2a_p \cdot z$, an even integer. The map is therefore well-defined.

Reducing an integer representative of $s + t$ modulo q subtracts qz for some integer vector z; the same calculation removes this carry after reduction modulo two. Multiplication by a ring scalar behaves identically. Reduction $R \to \mathbb{F}_2$ is well-defined because q is even, establishing (13). □

4.2 Generating a possibly nonfree kernel

Lemma 4.3. *Given a $b \times m$ matrix B over R, with $b \le m$, a generating set of $\ker_R B$ with at most m elements can be computed deterministically in polynomial bit complexity in m, r.*

Proof. Use invertible row and column operations to obtain

$$UBV = D, \tag{14}$$

where the nonzero diagonal entries of D are $2^{e_1}, \dots, 2^{e_t}$ with $0 \le e_i < r$. Track the column transformation V.

Here is an explicit elimination procedure. In the remaining submatrix choose a nonzero entry of minimum two-adic valuation e and move it to the pivot position. Multiply its row by an odd unit to make the pivot exactly 2^e. Every entry in the remaining submatrix is divisible by 2^e. Integer quotients of residue representatives therefore permit row subtraction to clear the pivot column, followed by column subtraction to clear the pivot row. Clearing the row does not restore entries below the pivot, because that column is already zero there. Repeat on the smaller submatrix.

All operations are invertible over R. Odd inverses can be computed by the Euclidean algorithm. Reduce every stored entry modulo q after each operation; the matrices and the tracked transformations then contain only $O(r)$-bit entries. There are polynomially many operations. A zero submatrix terminates the procedure without introducing a pivot.

Let $\mathbf{e}_i$ denote coordinate vectors. The kernel of D is generated by

$$2^{r-e_i}\mathbf{e}_i \quad (1 \le i \le t), \qquad \mathbf{e}_i \quad (t < i \le m). \tag{15}$$

A pivot with $e_i = 0$ contributes the zero generator and may be omitted. Equation (14) gives $B(Vz) = 0$ if and only if $Dz = 0$, so applying V to (15) gives the promised generators. No freeness assumption on the kernel is used. Empty matrices and $m = 0$ satisfy the same conclusion. □

Theorem 4.4 (Complete image with preimages)**.** *In polynomial bit complexity in n, r, one can compute a basis $\omega_1, \dots, \omega_d$ of Ω, with $d \le |X|$, and vectors $s_i \in \Sigma$ satisfying $f(s_i) = \omega_i$. Membership in Ω is decidable in polynomial time, with a witnessing multiplicity vector for every member.*

Proof. Apply Lemma 4.3 to obtain generators $g_1, \dots, g_\ell$ of Σ, with $\ell \le m$. For each g_j and pair p, compute the residue $c_p = a_p \cdot g_j \bmod q$ in $\{0, \dots, q-1\}$. It belongs to $\{0, h\}$ by the pair constraint. Thus $f(g_j)_p = c_p/h$ is an exact integer division; it does not divide by a nonunit in R.

Every $s \in \Sigma$ is an R-linear combination $\sum_j c_j g_j$. By Lemma 4.2,

$$f(s) = \sum_j (c_j \bmod 2) f(g_j). \tag{16}$$

Conversely, every binary combination of the images is the image of the corresponding sum of the generators. Hence these at most m images span all of Ω.

Perform binary Gaussian elimination on them, carrying along their preimages over R. Whenever image rows are added, add the corresponding preimages modulo q. The invariant $f(s_i) = \omega_i$ survives by (13). Remove dependent zero image rows. If $\delta \in \Omega$, solve $\delta = \sum_i \epsilon_i \omega_i$ over $\mathbb{F}_2$ and return $s = \sum_i \epsilon_i s_i \bmod q$.

There are $\binom{|Y|}{2} = O(n^2)$ image coordinates and at most m generators. Image computation uses $O(n^2 m^2)$ elementary arithmetic operations, and elimination with the retained witnesses has polynomial size. Combined with the compression and kernel bounds, this proves polynomial bit complexity. An equation description of Ω may alternatively be obtained from a basis of its binary orthogonal complement. □

Corollary 4.5 (One prescribed-support operation)**.** *A graph H is obtained from G by one valid r-local complementation supported in X if and only if the two graphs agree on pairs meeting X and their edge difference on Y belongs to Ω. The condition is decidable in polynomial bit complexity in n, r, and an accepting instance has constructible binary-encoded multiplicities.*

Proof. Lemma 4.1 identifies all valid multiplicities. For a pair in Y, the toggle predicate is exactly (12). A pair meeting the actual support is unchanged by definition. A pair meeting X outside that support has no common neighbor in X, so its common-neighbor multiplicity is zero and it is unchanged as well. These facts prove necessity and sufficiency. Representatives below q have at most r bits each. □

5 Interfaces with the published graph reduction

We record the interfaces from Claudet and Perdrix [1] needed for the full decision procedure. Numbering in this section refers to the long version, arXiv:2502.06566v3. The corresponding statements used in the frozen proof were checked in version 2 as well. These are dependencies, not new graph-theoretic claims of this paper.

An MLS cover is a family of inclusion-minimal nonempty local sets covering the vertices, where a local set has the form $D \cup \mathrm{Odd}_G(D)$ and $\mathrm{Odd}_G(D)$ consists of vertices with an odd number of neighbors in D. The source defines vertex types relative to such a cover and a standard form. We use its polynomial routines to construct those objects; we never enumerate all minimal local sets. In a standard form there are no type-Y vertices, and the type-X set is independent with every neighbor of type Z.

R1. Common standard form. Lemma 3.12 either rejects LU-equivalence soundly or returns ordinary-LC-equivalent representatives sharing an MLS cover. Its running time is $O(n^{6.38})$. Lemma 3.13 permits rejection if the resulting types disagree or the neighborhoods of corresponding type-X vertices disagree.

R2. Prescribed support. Lemma 3.14 reduces LC_r-equivalence of surviving standard forms to one r-local complementation on the type-X vertices and ordinary complementations on type-$\perp$ vertices. These operations commute. The generalized operation changes only pairs of type-Z vertices.

R3. Image-to-gadget reduction. Given a basis of the full toggle space, the construction before Lemma 3.17 deletes the type-X vertices and adds a degree-two auxiliary vertex for each toggled pair in each basis vector, with matched labels in both graphs. Lemma 3.17 gives equivalence to a constrained ordinary-LC problem: no complementation at a type-Z vertex, and either every auxiliary vertex for a given basis vector occurs once or none does.

R4. Constrained LC. Lemma 3.18 implements those constraints by linear conditions on local-Clifford variables. Its use of Proposition 3.5 requires connected input graphs with an even-degree vertex. On graphs with N vertices and ℓ additional equations, that proposition gives $O((N^2+\ell)N^2)$ binary operations.

R5. Level bound. Corollary 3.29 gives LU $\Rightarrow$ LC_r for graphs with at most $2^{r+3}-1$ vertices. The reverse implication is automatic from the local-unitary interpretation of the operations.

The expensive source step is Lemma 3.15, which builds the full incidence system and then computes the toggle image. Equations (6) and (10) have exactly its weights. Theorems 3.1 and 4.4 replace its construction without changing the resulting space. In particular, the vector-space structure of the toggle image and the downstream gadget reduction are already in [1]; the new point is computing their input with polynomial bit complexity.

6 The full algorithm

6.1 Components and labels

Lemma 6.1. *LU-equivalent labelled graph states have the same partition of their vertex set into connected components. Equivalence then holds if and only if it holds on each corresponding component.*

Proof. A graph state factors across a labelled cut $A : V \setminus A$ if and only if no edge crosses that

cut. To see the nontrivial direction, remove the diagonal phases contributed by edges lying entirely within either side of the cut. Up to a common normalization, the bipartite coefficient matrix becomes

$$C_{x,y} = (-1)^{x^{\mathsf{T}}\Gamma_{A,V\setminus A}y}. \tag{17}$$

If a crossing entry of Γ equals one, the rows indexed by the zero vector and the corresponding single-bit vector, and the analogous two columns, give the minor

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{18}$$

It has nonzero determinant. The state therefore has Schmidt rank at least two across the cut. With no crossing edges, the coefficient matrix factors.

Product unitaries preserve factorization across every fixed labelled cut. The finest partition into graph-state product factors is thus the connected-component partition, which must be the same for both inputs. When partitions coincide, a global product-unitary equality implies equality of the pure-state factors up to component phases, for example by taking reduced states on each component. Conversely, tensoring component transformations supplies a transformation on the whole graph, with their phases absorbed into the global phase. □

Compute these partitions and reject if they differ. One-vertex components are immediate. On a common connected component of order $v \geq 2$, choose

$$r = \max\{1, \lceil \log_2(v+1) \rceil\}. \tag{19}$$

This gives $v \leq 2^{r+3} - 1$ and $r = O(\log v)$, so interface R5 makes the LU and LC_r questions identical.

Apply R1 and the type and neighborhood tests. If they succeed, let G_1, G_2 be the standard forms and let X, Z, W be their common type-X, type-Z, and type-$\perp$ sets. The graphs remain connected: any edge removed by ordinary complementation at a vertex is replaced by a two-edge path through that vertex. Compute the image basis from Theorem 4.4 using G_1 and X. Its only possibly nonzero coordinates are pairs in Z, since vertices outside $X \cup Z$ have no neighbor in X. Restricting to these coordinates loses no information.

6.2 The nonzero-image branch

Suppose $d = \dim \Omega > 0$. For each basis vector ω_i and each pair $\{u, w\} \subseteq Z$ on which it is one, add a new vertex p^i_{uw} joined only to u, w, after deleting X. Use the same new labels in both graphs. Let P_i be the set of new vertices for ω_i. Their total number is at most $d\binom{|Z|}{2}$, and

$$N \leq v + d\binom{v}{2} \leq v + |X|\binom{v}{2} = O(v^3). \tag{20}$$

Lemma 6.2 (Applicability of constrained LC)**.** *When $d > 0$, both augmented graphs are connected and have an even-degree vertex.*

Proof. Every basis vector is nonzero, so at least one auxiliary vertex is added; each such vertex has degree two.

For connectivity, fix $x \in X$. The multiplicity vector $h\mathbf{e}_x$ is in Σ, because every weighted incidence row has an even coefficient and hence its scalar product with this vector is divisible by q. Its toggle vector is exactly the clique on $N_{G_1}(x)$. As the basis vectors span Ω, every pair of distinct neighbors of x must occur in at least one basis vector. Some auxiliary vertex therefore supplies a path of length two between each such pair.

Take any two vertices outside X and a simple path between them in G_1. Independence of X ensures that each passage through an internal vertex $x \in X$ has two neighbors outside X. Replace that passage by the auxiliary path just described. Vertices of X with degree at most one cannot occur internally on the original simple path. Thus the remainder together with the auxiliary vertices is connected. The remainder is nonempty because G_1 is connected, has at least two vertices, and X is independent. All new vertices attach to that remainder. The same argument applies to G_2, using its connectedness and the identical neighborhoods on X. □

By R3, the required LC_r predicate is now the constrained ordinary-LC predicate on these augmented graphs. In the source's binary local-Clifford coordinates, write the relevant variables as β_u, γ_u to avoid confusion with our incidence matrix B. R4 imposes

$$\beta_u = 0 \ (u \in Z), \qquad \gamma_p = 0 \ (p \in \textstyle\bigcup_i P_i), \qquad \beta_p = \beta_{p_i} \ (p \in P_i), \tag{21}$$

where p_i is one representative of each nonempty group. There are $O(N + v)$ extra equations. Lemma 6.2 verifies the hypotheses of Proposition 3.5, so the constrained predicate can be decided in polynomial time in the actual augmented size N. Equation (20) transfers that bound to a polynomial in v.

6.3 The zero-image branch

Lemma 6.3. *If $\Omega = \{0\}$, then G_1 and G_2 are LC_r-equivalent if and only if they are ordinarily LC-equivalent.*

Proof. One implication is immediate. For the other, R2 expresses a LC_r equivalence by ordinary complementations on W and one valid generalized operation on X. These operations commute. Since $\Omega = \{0\}$, the latter operation changes no edge of G_1 and can be omitted. Only ordinary complementations remain. □

In this case run the ordinary LC algorithm [3] on the original standard forms G_1, G_2. Do not first delete X and apply the constrained routine to possibly disconnected graphs or to graphs lacking an even-degree vertex. Lemma 6.3 supplies a separate exact decision rule, including the case $X = \varnothing$.

6.4 Correctness and bit complexity

Algorithm 2: labelled graph-state LU equivalence

1. Compare labelled connected-component partitions. Reject if they differ.
2. For each common component of order $v \geq 2$, choose r by (19).
3. Use R1 to standardize and check types and type-X neighborhoods. Reject on failure.
4. Use Algorithms 1 and Theorem 4.4 to compute the complete toggle basis.
5. If $d = 0$, decide ordinary LC on the original standard forms. If $d > 0$, build the matched gadgets and decide constrained LC using (21).
6. Accept exactly when every component test accepts; singletons always accept.

Proof of Theorem 1.1. Lemma 6.1 justifies the first step and all componentwise decisions. R1 makes every standardization or type-based rejection sound. On a surviving component, Lemma 4.1 and Theorem 4.4 compute precisely the full toggle space required by R3. For $d > 0$,

R3, R4, and Lemma 6.2 make the constrained test equivalent to LC_r equivalence. For $d = 0$, Lemma 6.3 gives that equivalence directly. R5 and (19) identify it with LU-equivalence. This proves soundness and completeness, and every construction preserves the original labels.

For the complexity bound, the common-cover stage has the credited polynomial bound in R1. Compression uses at most $r|X|$ strict enlargements, each represented by a matrix with polynomially many rows and $O(r+1)$-bit entries. Its Hermite computations therefore have polynomial bit complexity. Kernel construction, exact image evaluation, and binary elimination have polynomial matrix sizes and bounded residue lengths. Since $r = O(\log v)$ and $2^r \le 2(v+1)$, no hidden exponential precision parameter is introduced.

The augmented graphs have $O(v^3)$ vertices. Evaluating the constrained routine's bound on this actual size, rather than on the original graph size, still gives a fixed polynomial. The zero-image branch uses an ordinary polynomial LC test. Summing a fixed polynomial bound over at most n components remains polynomial in n. We do not optimize the exponent or claim a practical benchmark for the complete procedure. □

7 Exact diagnostics and reproducibility

The proofs above are analytic. The accompanying calculations check algebraic implementations and constants on finite systems. They neither certify the cited graph reductions nor implement the entire decision procedure.

The earlier release contains 45 comparisons between weighted closure and direct high-order incidence enumeration, and 36 kernel/image comparisons against exhaustive multiplicity enumeration. All passed when rerun on 22 September 2026. Inspection showed that all 45 earlier closure examples stabilized at their initial seed lattice. Those examples therefore did not exercise strict closure growth. We retained them as historical checks and added cases chosen to test this missing behavior.

For the new closure tests, take $t \in \{3, 4, 5, 6\}$ and give X one coordinate for every Boolean pattern in $\{0, 1\}^t$. Use levels $r \in \{1, 2, 3, 4, 5, 7\}$. Among these 24 systems, 12 require strict enlargement, with up to three enlargement steps. In each of those 12 cases, omitting the closure iteration and retaining only the seeds gives a different lattice from the direct high-order construction. This detects an implementation that accidentally skips the central step.

The same extended script checks every 2×2 matrix over $\mathbb{Z}/4\mathbb{Z}$, comparing the generated kernel with exhaustive enumeration. It further enumerates every binary incidence matrix of shapes 2×3 and 3×2 at levels one, two, and three, and compares the complete toggle image with the images of every admissible multiplicity vector.

Diagnostic family	Systems	Result or coverage
Earlier weighted-closure comparisons	45	All passed; no strict growth
Earlier modular-kernel/image comparisons	36	All passed
Boolean-pattern closure systems	24	12 grow; at most 3 steps
All 2×2 matrices over $\mathbb{Z}/4\mathbb{Z}$	256	Complete kernels agree
All selected binary incidence systems	384	42,752 vectors enumerated
Explicit boundary cases	5	All passed

Table 1: Exact finite diagnostics. Counts refer to distinct test systems within each family, not to graphs in an end-to-end LU benchmark.

Among the 256 arbitrary modular matrices, 81 have kernels of cardinality two or eight, which certifies that these kernels are nonfree over $\mathbb{Z}/4\mathbb{Z}$. Among the 384 incidence systems, 129 have zero toggle image. The five explicit boundary cases cover the empty graph, empty

prescribed support, empty outside set, isolated vertices at level one, and a zero toggle image. All extended checks passed using exact integer and symbolic arithmetic in a network-disabled container with NumPy 2.5.3 and SymPy 1.14.0. The extended script itself requires only SymPy in addition to Python's standard library.

The source package preserves the frozen proof, the original scripts and results, the additional script and its complete results, the manuscript source, and an evidence map. Source statements were opened and compared by the AI assistant. The evidence record marks independent human verification as pending.

8 Discussion

The main result follows by changing one representation inside an existing reduction. Listing all incidence rows is unnecessary because their weighted coordinatewise products can be generated by a finite family of linear maps. A finite-length module bounds the number of strict updates, while Hermite reduction bounds the description retained after each update. The binary image remains exactly the image used by the prior graph reduction.

Three distinctions are essential. First, the kernel is computed over a finite ring, not over $\mathbb{F}_2$. Second, the target image contains every admissible toggle, not only a list of sampled or individually found operations. Third, the downstream runtime must be evaluated on the augmented graph size, and the constrained-LC hypotheses must actually hold. The separate zero-image branch and the connectivity argument address these requirements.

The result is a decision theorem for labelled graph states. It does not decide equivalence while allowing arbitrary vertex permutations, and no claim about that different problem is made. Although witnessing multiplicities are constructed for the prescribed-support operation, this manuscript does not supply a full executable graph-to-local-unitary compiler. Developing and benchmarking the complete algorithm, improving its polynomial exponent, and obtaining external mathematical review remain important next steps.

Authorship, preparation, and availability

The proof candidate, its diagnostics, internal reviews, and this exposition were developed with substantial AI assistance under the author's direction. Two separate internal AI reviews reported no error in the same frozen round-21 submission. Those reviews are fallible model assessments and do not constitute peer review or formal proof-assistant verification. The manuscript is presented for scrutiny rather than as an externally accepted resolution. Historical priority remains unverified.

Scientific Agent Skills [5] supplied guidance for evidence tracking and scientific exposition; it is not a source of mathematical verification. The original proof and submission are available in QIQC issue 97. The research page hosts the manuscript and reproducibility package. The frozen proof was contributed under CC BY 4.0 and the original diagnostic scripts under Apache-2.0; cited third-party works retain their own terms.

References

[1] N. Claudet and S. Perdrix, *Deciding Local Unitary Equivalence of Graph States in Quasi-Polynomial Time*, in 52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025), LIPIcs **334**, 59:1–59:20 (2025). doi:10.4230/LIPIcs.ICALP.2025.59. Numbering used here: arXiv:2502.06566v3.

[2] N. Claudet, *Local Equivalences of Graph States*, doctoral thesis (2025). arXiv:2511.22271v2. Polynomial-time question: Chapter 8, p. 107.

[3] M. Van den Nest, J. Dehaene, and B. De Moor, *Efficient algorithm to recognize the local Clifford equivalence of graph states*, Physical Review A **70**, 034302 (2004). doi:10.1103/PhysRevA.70.034302; arXiv:quant-ph/0405023.

[4] R. Kannan and A. Bachem, *Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix*, SIAM Journal on Computing **8**(4), 499–507 (1979). doi:10.1137/0208040.

[5] T. Kassis, V. Agarwal, Y. He, D. Patel, and A. M. Brueckner, *Scientific Agent Skills: A Library of Procedural Knowledge for Research Agents* (2026). doi:10.48550/arXiv.2609.00065.