QUANTUM WEIGHTED HYPERGRAPH OPERATOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Many learning problems depend on higher-order interactions among groups of variables. Hypergraphs make these interactions explicit, motivating quantum representations that preserve their structure and support recovery of their weights. We introduce the Quantum Weighted Hypergraph Operator (Q-WHO), a unified framework for encoding weighted hypergraphs into quantum circuits and decoding their interaction weights. Each hyperedge defines a gate built from a shared non-scalar local Hermitian matrix. This choice gives the circuit a common tensor spectral structure while preserving its hypergraph organization. From this structure, we derive an explicit inverse and exact weight recovery conditions for real, continuous-phase, and discrete weights. The operator acts on arbitrary input states; decoding uses exact classical descriptions of a known input-output state-vector pair with nonzero input amplitudes throughout the product eigenbasis. The same tensor factorization supports in-place state-vector simulation and decoding without constructing a dense unitary matrix. Reversible coordinate transformations further show how operator choice can express the same circuit action up to global phase with fewer nonzero hyperedge weights. Graph states, weighted hypergraph states, and commuting Pauli-Z gates follow from particular choices of the local matrix, weights, and input. Together, these results provide a common framework for constructing, recovering, and adapting quantum representations of higher-order interactions.

1 INTRODUCTION

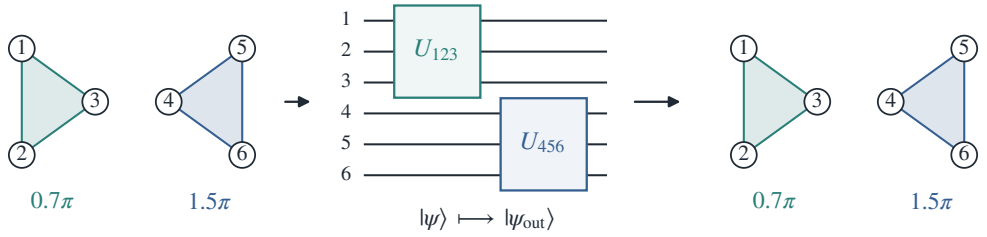
Many learning problems involve interactions among groups of variables rather than pairs alone. Hypergraphs express these interactions by allowing an edge to join several vertices. This preserves higher-order information that may be lost in a graph projection (Wang & Kleinberg, 2024) and supports models that learn directly from multi-way relations (Chien et al., 2022; Wang et al., 2023). A useful representation of a hypergraph should retain both its interaction weights and the identity of the variables they connect.

Quantum circuits provide a natural setting for such representations because a gate can act jointly on the qubits associated with a hyperedge. Quantum graph and hypergraph states use this correspondence to construct states from prescribed phase interactions (Qu et al., 2013b; Rossi et al., 2013; Qu et al., 2013a). For a broader operator framework, two questions arise together: (1) how should weighted hyperedges determine a circuit that can act on an arbitrary input, and (2) when does a known input-output pair contain enough information to recover the original hypergraph?

We address both questions with the *Quantum Weighted Hypergraph Operator* (Q-WHO), illustrated in Figure 1. Each hyperedge specifies a gate assembled from one shared local Hermitian matrix. Their product is a composite unitary operator whose organization follows the hypergraph: relabeling vertices permutes circuit wires, and disconnected components give independent operator factors. The shared building block makes the gates commute and places them in a common spectral coordinate system. Encoding and decoding can therefore be derived from the same tensor structure.

The inverse has a simple interpretation. In the known local eigenbasis, the circuit changes each input amplitude only by a phase. When every input coordinate is nonzero, ratios of output and input amplitudes reveal these phase changes. Removing their global phase and applying a tensor inverse recovers the hyperedge weights under the specified real or modular convention. We call this procedure *hypergraph decoding* and use *exact weight recovery* for its guarantee. The input can be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Weighted hypergraph (b) Q-WHO circuit (c) Decoded hypergraph

Figure 1: Encoding and decoding with Q-WHO. (a) A weighted hypergraph. (b) Its circuit, with each logical gate acting on the vertices in its label. (c) The recovered hypergraph. The illustrated projector weights use representatives in $[0, 2\pi)$. Decoding requires a known input with full support in the ordered eigenbasis and the stated recovery conditions. The classical algorithms apply and decode the whole operator from state vectors without constructing a dense unitary matrix.

nonuniform or entangled. Under the declared weight convention, full support is the only restriction on its amplitudes.

The choice of local operator also determines how compactly interactions can be expressed. A single weighted edge under one Hermitian matrix may expand into many tied phase edges under another while producing the same circuit action up to global phase. This gives an explicit way to compare and select interaction coordinates. Established graph-state, weighted-hypergraph-state, and commuting Pauli constructions then follow by choosing the local matrix, allowed hyperedges, weights, and input state. Their familiar inverse transforms emerge from the general decoder.

Our contributions have three parts. First, we introduce a unified operator framework parameterized by a shared non-scalar local Hermitian matrix. It preserves hypergraph structure, acts on arbitrary inputs, and contains established state and gate constructions as special cases. Second, we derive a general tensor decoder and characterize exact weight recovery from known input-output state vectors for real, continuous-phase, and discrete weights. Third, we establish reversible changes of interaction coordinates and show how operator choice can reduce the number of nonzero hyperedge weights for the same operator up to global phase. The same factorization gives in-place state-vector simulation and decoding algorithms with nearly linear arithmetic cost in the explicit state-vector dimension, without forming a dense operator matrix. The source code will be made publicly available if the paper is accepted.

2 Related Work

Hypergraph representation learning. AllSet (Chien et al., 2022) and equivariant hypergraph diffusion neural operators (Wang et al., 2023) learn directly from multi-way relations. Ordinary graph projection can discard higher-order information (Wang & Kleinberg, 2024). Directional sheaf models use complex Hermitian hypergraph operators for feature propagation (Mule et al., 2026), while quantum graph neural networks use circuits for graph representation learning and prediction (Ye et al., 2025). Q-WHO studies a reversible representation of these interactions: weighted hyperedges specify a circuit, and a known input-output pair supports decoding of its weights under explicit recovery conditions.

Quantum graph and hypergraph states. Graph states encode pairwise relations through phase operations (Hein et al., 2004). Hypergraph states extend the construction to arbitrary groups (Qu et al., 2013b; Rossi et al., 2013). Continuous weights are already present in weighted graph and hypergraph states (Hartmann et al., 2007; Qu et al., 2013a). Their connectivity also carries information about entanglement (Rossi et al., 2013; Xiong et al., 2018). Qudit extensions allow more than two local levels and include calibrated hypergraph-state (Steinhoff et al., 2017; Zucchini, 2026a;b). Poderini et al. (2026) review the broader state family. Q-WHO develops encoding and inverse transforms for a shared local Hermitian matrix within this broader family of phase-state representations.

Spectral representations and inversion. Diagonal-unitary representations and circuit synthesis can use the Fourier–Walsh basis (Welch et al., 2014). In Q-WHO, the shared local Hermitian matrix determines a tensor spectral transform that connects hyperedge weights to circuit phases. Its inverse supports weight decoding, and its spectrum determines the corresponding recovery conditions. Projector and Pauli-Z constructions arise from particular matrix choices, as detailed in Section 5.3. Reversible changes of coordinates connect these choices and reveal how they affect interaction sparsity. Thus the framework jointly describes operator construction, weight recovery, and representation choice; circuit synthesis addresses the implementation of the resulting gates.

3 Preliminaries and Problem Formulation

3.1 Weighted hypergraphs and phase conventions

Let $V = \{1, \dots, N\}$ and $D = 2^N$. We index subsets of V by their binary indicators in the order $00\cdots0, \dots, 11\cdots1$, with the first bit most significant. We write $\chi_e(q) = \mathbb{I}[q \in e]$ and $\boldsymbol{x} \in \{0,1\}^N$ for a binary configuration. The empty subset and the reference configuration $\mathbf{0}$ both have index zero. Bold symbols denote column vectors. The symbols $\otimes$, $\dagger$, and T denote the Kronecker product, conjugate transpose, and transpose. We use I_m for the $m \times m$ identity and take an empty product to be one.

A real-weighted hypergraph is $G = (V, w)$, where $w : 2^V \setminus \{\varnothing\} \to \mathbb{R}$ assigns a weight to every nonempty subset. Its active edges are those with $w_e \neq 0$. We collect the weights in $\boldsymbol{w}_\star \in \mathbb{R}^{D-1}$ and use $\boldsymbol{w} = \mathsf{Pad}(\boldsymbol{w}_\star)$ to insert an initial empty-subset coefficient $w_\varnothing = 0$. The operation Drop removes that coordinate. The auxiliary empty coefficient will account for global phase when changing interaction coordinates.

For a real angle z, $[z]_{2\pi}$ denotes its equivalence class modulo 2π, and

$$\bar{z} = \mathrm{wrap}_{2\pi}(z) := z \bmod 2\pi \in [0, 2\pi) \tag{1}$$

is its chosen representative. These operations apply entrywise to vectors. An *unwrapped* phase retains its real value before reduction. For $T \geq 2$, $\mathbb{Z}_T = \mathbb{Z}/T\mathbb{Z}$ has representatives $0, \dots, T-1$. Thus T-ary phases $2\pi s/T$ lie on a uniform grid in the same interval, with binary phases represented by 0 and π. This phase convention does not restrict general real hyperedge weights.

3.2 States and the local operator

A pure N-qubit state $|\psi\rangle$ is a unit vector in $\mathbb{C}^D$. The computational basis consists of tensor products of $|0\rangle = (1,0)^\mathsf{T}$ and $|1\rangle = (0,1)^\mathsf{T}$, and $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. A unitary operator preserves the state norm. Multiplying a state by a global phase $e^{i\gamma}$ leaves its density matrix $|\psi\rangle\langle\psi|$ unchanged.

Our local building block is a Hermitian matrix $A \in \mathbb{C}^{2\times2}$ that is not a scalar multiple of I_2. Choose an eigendecomposition

$$A = Q\Lambda Q^\dagger, \qquad \Lambda = \mathrm{diag}(\lambda_0, \lambda_1), \qquad \lambda_0, \lambda_1 \in \mathbb{R}, \quad \lambda_0 \neq \lambda_1. \tag{2}$$

Here Q is unitary and is computed from A. We use the same eigenvalue order and corresponding columns of Q throughout a computation, without requiring the eigenvalues to be numerically sorted. The product basis is $Q^{\otimes N}$, so $(Q^\dagger)^{\otimes N}|\psi\rangle$ gives the state's coordinates in that basis. Eigenvector phases may be chosen arbitrarily and used consistently. They cancel when applying the operator and in the input-output amplitude ratios used below.

3.3 Hypergraph encoding, circuit application, and decoding

Hypergraph encoding maps G and A to a unitary operator expressed as a product of hyperedge gates. *Circuit application* maps a chosen input $|\psi\rangle$ to $|\psi_{\mathrm{out}}\rangle = U_A(G)|\psi\rangle$. *Hypergraph decoding* uses A and exact descriptions of this pair to recover weights under a declared real or modular convention. *Exact weight recovery* means that the recovered weights agree with the original ones in that convention. Applying $U_A^\dagger$ to the output instead recovers the input state.

Q-WHO can act on any input state. Decoding requires *full support*: every input amplitude in the product eigenbasis of A must be nonzero. We first derive the common spectral transform, then use it for classical state-vector computation and weight recovery.

4 Q-WHO Encoding and Decoding

The shared local operator makes all hyperedge generators diagonal in the same product basis. Their spectral values depend linearly on the weights, which gives both a direct rule for computing the output state and an explicit inverse.

4.1 Encoding through hyperedge gates

For a hyperedge e, place A on its vertices and identities elsewhere:

$$A_e = \bigotimes_{q=1}^{N} A^{[\chi_e(q)]}, \qquad A^{[0]} = I_2, \quad A^{[1]} = A, \tag{3}$$

where I_2 is a 2×2 identity matrix. Define its gate and the full Hermitian generator by

$$U_{A,e}(w_e) := \exp(iw_e A_e), \tag{4}$$

$$H_A(\boldsymbol{w}) := \sum_{e\subseteq V} w_e A_e. \tag{5}$$

We define the *Quantum Weighted Hypergraph Operator* (Q-WHO) as the gate product, following the subset indexing:

$$U_A(\boldsymbol{w}) := \prod_{e\subseteq V} U_{A,e}(w_e). \tag{6}$$

We use "generator" for H_A and "operator" for U_A. Any evolution duration and sign are absorbed into the real weights. For an input hypergraph G with $w_\varnothing = 0$, we also write $H_A(G)$ and $U_A(G)$.

Proposition 1 (Unitarity and commutativity)**.** *For real weights, $U_A(\boldsymbol{w})$ is unitary, and $A_eA_f = A_fA_e$ for all $e, f \subseteq V$. Consequently, the gate product in Equation 6 is unchanged when its gates are permuted and satisfies $U_A(\boldsymbol{w}) = \exp(iH_A(\boldsymbol{w}))$.*

At an overlapping vertex, both generators use the same matrix A, so their local factors commute. Appendix A.1 gives the proof. Since $A_\varnothing = I_D$, an empty-edge weight contributes only $e^{iw_\varnothing}$. We set it to zero for input hypergraphs and retain it temporarily when changing coordinates.

This product gives a quantum-circuit implementation with one logical gate per active hyperedge. Each gate acts only on that edge's qubits and may require further decomposition into the hardware gate set. Relabeling vertices permutes circuit wires, while disjoint hypergraphs give tensor products of their operators, as proved in Appendix B.2. For classical simulation, we can instead evaluate the whole operator through its spectral factorization.

4.2 Spectral factorization and state-vector simulation

In the product eigenbasis, the spectral value of the generator at configuration $\boldsymbol{x}$ is

$$\phi_{\boldsymbol{x}} = \sum_{e\subseteq V} w_e \prod_{q\in e} \lambda_{x_q}. \tag{7}$$

Let $\boldsymbol{\phi}$ collect these real, unwrapped phases and define

$$B_A = \begin{bmatrix} 1 & \lambda_0 \\ 1 & \lambda_1 \end{bmatrix}. \tag{8}$$

The columns select an identity factor or an A factor, and the rows select the corresponding eigenvector coordinates.

Proposition 2 (Kronecker spectral transform)**.** *The phase vector and operator satisfy*

$$\boldsymbol{\phi} = L_A\boldsymbol{w}, \qquad L_A = B_A^{\otimes N}, \tag{9}$$

$$U_A(\boldsymbol{w}) = Q^{\otimes N}\,\mathrm{diag}(e^{i\phi_{\boldsymbol{x}}})_{\boldsymbol{x}}(Q^\dagger)^{\otimes N}. \tag{10}$$

The proof appears in Appendix A.2. Thus encoding consists of a linear transform of the weights followed by entrywise exponentiation in the product eigenbasis. Since $\det B_A = \lambda_1 - \lambda_0 \neq 0$, every real phase vector has unique full coefficients. Choosing real lifts of diagonal phases therefore represents every unitary diagonal in this fixed product basis. Omitting the empty coefficient changes only global phase. Different eigenvalues give different interaction coordinates within this same unrestricted operator family, a distinction developed in Section 5.

Applying Q-WHO to a state vector. First express a normalized input in the product eigenbasis:

$$\boldsymbol{z} = (Q^\dagger)^{\otimes N} |\psi\rangle . \tag{11}$$

The diagonal factor in Equation 10 multiplies each coordinate by its spectral phase. Transforming back gives

$$y_{\boldsymbol{x}} = e^{i\phi_{\boldsymbol{x}}} z_{\boldsymbol{x}}, \qquad |\psi_{\text{out}}\rangle = Q^{\otimes N} \boldsymbol{y}. \tag{12}$$

This evaluates the entire circuit using local basis changes and a length-D phase table, without forming a dense unitary or simulating each hyperedge gate separately.

4.3 Relative phases from a known input-output pair

For decoding, express the known input and output in the product eigenbasis to obtain $\boldsymbol{z}$ and $\boldsymbol{y}$. Their coordinate ratios reveal spectral phases wherever the input amplitude is nonzero. To remove the independent global phases of the two state descriptions, use configuration $\mathbf{0}$ as a reference and set

$$\delta_{\boldsymbol{x}} = \phi_{\boldsymbol{x}} - \phi_{\mathbf{0}}, \qquad \boldsymbol{x} \neq \mathbf{0}. \tag{13}$$

Theorem 3 (Relative phases from a known input-output pair)**.** *Let $|\psi_{\text{out}}\rangle = U_A(\boldsymbol{w})|\psi\rangle$, with A known. If every $z_{\boldsymbol{x}}$ in Equation 11 is nonzero, the pair determines*

$$e^{i\delta_{\boldsymbol{x}}} = \frac{y_{\boldsymbol{x}} z_{\mathbf{0}}}{z_{\boldsymbol{x}} y_{\mathbf{0}}}. \tag{14}$$

These ratios determine U_A up to global phase within the family diagonal in the product eigenbasis of A.

Proof. Equation 12 gives $y_{\boldsymbol{x}}/z_{\boldsymbol{x}} = e^{i\phi_{\boldsymbol{x}}}$. Dividing by the reference ratio yields Equation 14. Independent global phases on the input and output multiply all ratios $y_{\boldsymbol{x}}/z_{\boldsymbol{x}}$ by the same factor, which cancels. Rephasing an eigenvector multiplies the corresponding input and output coordinates by the same factor and also cancels. The relative phases fix the diagonal operator up to a global phase. □

This result uses exact state-vector descriptions and allows nonuniform or entangled inputs. It recovers wrapped relative phases. Recovering the original weights from them requires the conditions below.

4.4 Weight recovery and its conditions

Subtracting the reference row of the spectral transform gives $\boldsymbol{\delta}_\star = R_A \boldsymbol{w}_\star$, where

$$(R_A)_{\boldsymbol{x},e} = \prod_{q \in e} \lambda_{x_q} - \lambda_0^{|e|}, \quad \boldsymbol{x} \neq \mathbf{0}, \quad e \neq \varnothing. \tag{15}$$

Both the full and relative transforms have explicit inverses:

$$B_A^{-1} = \frac{1}{\lambda_1 - \lambda_0} \begin{bmatrix} \lambda_1 & -\lambda_0 \\ -1 & 1 \end{bmatrix}, \qquad R_A^{-1} = \mathsf{Drop}\,(B_A^{-1})^{\otimes N} \mathsf{Pad}. \tag{16}$$

Appendix A.3 proves this identity. If the original unwrapped relative phases are available, it recovers real weights as $\boldsymbol{w}_\star = R_A^{-1} \boldsymbol{\delta}_\star$.

An input-output pair alone does not supply these unwrapped values. Two weight vectors represent the same operator up to global phase exactly when

$$\boldsymbol{w}'_\star - \boldsymbol{w}_\star \in 2\pi R_A^{-1} \mathbb{Z}^{D-1}. \tag{17}$$

The inverse can therefore select a unique real representative in

$$\Omega_A = R_A^{-1}\big([0, 2\pi)^{D-1}\big) . \tag{18}$$

Exact recovery of the original real weights requires the promise that they lie in this domain. Otherwise the inverse returns a representative of the same operator up to global phase. The domain Ω_A is generally not a coordinatewise interval for weights.

When λ_0, λ_1 are integers, adding 2π to any individual weight leaves the operator unchanged. Continuous phase weights can then be represented in $[0, 2\pi)$, and T-ary weights take the form $w_e = 2\pi s_e/T$ with $s_e \in \mathbb{Z}_T$. An edge is active when its weight residue is nonzero. The relative phases of a T-ary model lie on the same grid, with labels $t_{\boldsymbol{x}} \in \mathbb{Z}_T$ defined by $e^{i\delta_{\boldsymbol{x}}} = e^{2\pi i t_{\boldsymbol{x}}/T}$.

Algorithm 1 State-vector simulation of Q-WHO

Input: A, weights $\boldsymbol{w}_\star$, and a state buffer $\boldsymbol{v}$ containing $|\psi\rangle$.
Output: the same buffer containing $U_A(\boldsymbol{w})\,|\psi\rangle$.

1 Compute $A = Q\,\mathrm{diag}(\lambda_0, \lambda_1)Q^\dagger$ and form B_A.
2 Set $\boldsymbol{\phi} \leftarrow \mathsf{Pad}(\boldsymbol{w}_\star)$. Apply $\mathsf{KronMatVec}(B_A, \boldsymbol{\phi})$.
3 Apply $\mathsf{KronMatVec}(Q^\dagger, \boldsymbol{v})$.
4 Set $v_{\boldsymbol{x}} \leftarrow e^{i\phi_{\boldsymbol{x}}} v_{\boldsymbol{x}}$ for every $\boldsymbol{x}$.
5 Apply $\mathsf{KronMatVec}(Q, \boldsymbol{v})$ and return $\boldsymbol{v}$.

Theorem 4 (Exact weight recovery)**.** *Under the full-support hypothesis of Theorem 3, let* $\bar{\boldsymbol{\delta}}_\star \in [0, 2\pi)^{D-1}$ *be the relative phase representatives obtained from the pair.*

1. *For* $\boldsymbol{w}_\star \in \Omega_A$*, the original real weights are*
$$\boldsymbol{w}_\star = R_A^{-1}\bar{\boldsymbol{\delta}}_\star. \tag{19}$$
2. *For integer* λ_0, λ_1*, all continuous weights modulo* 2π *are uniquely recoverable if and only if* $|\lambda_1 - \lambda_0| = 1$*. Their representatives are* $\mathrm{wrap}_{2\pi}(R_A^{-1}\bar{\boldsymbol{\delta}}_\star)$.
3. *For integer* λ_0, λ_1*, all* T*-ary labels are uniquely recoverable if and only if* $\gcd(\lambda_1 - \lambda_0, T) = 1$*. In that case,*
$$\boldsymbol{s}_\star \equiv \mathsf{Drop}\,(B_{A,T}^{-1})^{\otimes N}\mathsf{Pad}(\boldsymbol{t}_\star) \pmod T, \tag{20}$$
where $B_{A,T}^{-1}$ *is the inverse of* B_A *over* $\mathbb{Z}_T$.

For real weights in Ω_A, the observed phase representatives are the intended unwrapped relative phases, so R_A^{-1} recovers the weights. For integer eigenvalues, R_A is integral, with $\det R_A = (\lambda_1 - \lambda_0)^{N2^{N-1}}$. Its map modulo 2π is bijective exactly when $\det R_A = \pm 1$; over $\mathbb{Z}_T$, it is bijective exactly when $\gcd(\det R_A, T) = 1$. These criteria give the two spectral-gap conditions. Appendices A.3–A.4 provide the full proofs. The last two conditions concern bijectivity on the full modular coefficient spaces. They do not exclude additional recovery results for restricted weight families. Changing the phase representative interval does not change these conditions.

4.5 State-vector simulation and decoding algorithms

Both algorithms use the same Kronecker multiplication routine. Write $\mathsf{KronMatVec}(C, \boldsymbol{u})$ for an in-place evaluation of $C^{\otimes N}\boldsymbol{u}$ by successive 2×2 updates along each bit position. Appendix C gives the routine. Algorithm 1 applies the complete operator classically, while Algorithm 2 recovers weights from a known pair. Each algorithm computes a consistent eigendecomposition of its input A.

Each Kronecker multiplication uses N stages with $D/2$ pair updates per stage. Both algorithms therefore cost $\mathcal{O}(ND)$ classical arithmetic and $\mathcal{O}(D)$ numerical storage. Algorithm 1 overwrites the state buffer and uses one additional phase buffer. Decoding preserves the supplied vectors by working on copies. Neither algorithm forms a $D \times D$ matrix. These costs are nearly linear in the explicit state-vector dimension $D = 2^N$, rather than polynomial in the qubit count. Discrete complexity counts operations in $\mathbb{Z}_T$.

If unwrapped relative phases are supplied separately, weight recovery only requires the inverse in Equation 16. If exact discrete phase labels are supplied, Algorithm 2 starts directly from its discrete inverse data. These are additional input models, rather than information extracted by unwrapping a single pair.

5 Operator Choice and Established Special Cases

The spectral transform separates a circuit's phases from the coordinates used to describe its interactions. We now use this distinction to compare local operators, explain changes in coefficient sparsity, and recover familiar state and gate constructions.

Algorithm 2 Classical hypergraph decoding from a known input-output pair

Input: A, $|\psi\rangle$, $|\psi_{\text{out}}\rangle$, and a recovery convention satisfying Theorem 4.
Output: nonempty real weights, phase representatives, or discrete labels.

1 Compute an eigendecomposition of A. Copy the input and output into $\boldsymbol{z}, \boldsymbol{y}$ and apply $\mathsf{KronMatVec}(Q^\dagger, \cdot)$ to both.
2 Require $z_{\boldsymbol{x}} \neq 0$ for every $\boldsymbol{x}$. Extract $\bar{\boldsymbol{\delta}}_\star$ from Equation 14 in $[0, 2\pi)$.
3 Prepare the inverse data for the declared convention:
  Real or continuous phase weights: $\boldsymbol{v} \leftarrow \mathsf{Pad}(\bar{\boldsymbol{\delta}}_\star)$, $C \leftarrow B_A^{-1}$.
  T**-ary labels:** identify the exact grid labels $\boldsymbol{t}_\star$ of the phase ratios, then set $\boldsymbol{v} \leftarrow \mathsf{Pad}(\boldsymbol{t}_\star)$ and $C \leftarrow B_{A,T}^{-1}$.
4 Apply $\mathsf{KronMatVec}(C, \boldsymbol{v})$, using arithmetic modulo T in the discrete case. Drop the empty coordinate.
5 Return the real values, wrap continuous phase weights to $[0, 2\pi)$, or return discrete representatives in $\{0, \ldots, T-1\}$.

5.1 Changing interaction coordinates

Consider two nondegenerate local Hermitian matrices A and C with the same ordered eigenbasis Q. Their full coefficient vectors describe the same unwrapped phase vector when

$$\boldsymbol{v} = L_C^{-1} L_A \boldsymbol{w} = (B_C^{-1} B_A)^{\otimes N} \boldsymbol{w}. \tag{21}$$

Equivalently, write $A = \alpha I_2 + \beta C$, with real α and $\beta \neq 0$; this is possible for any such pair. Expanding the local factors gives the explicit weight transformation and its inverse:

$$v_f = \beta^{|f|} \sum_{e \supseteq f} \alpha^{|e|-|f|} w_e, \qquad f \subseteq V, \tag{22}$$

$$w_e = \sum_{f \supseteq e} (-\alpha)^{|f|-|e|} \beta^{-|f|} v_f, \qquad e \subseteq V. \tag{23}$$

With the empty coefficient retained, this gives $H_C(\boldsymbol{v}) = H_A(\boldsymbol{w})$ and $U_C(\boldsymbol{v}) = U_A(\boldsymbol{w})$. Dropping the transformed empty coefficient changes only global phase. The map is reversible over real coefficients, but need not preserve which hyperedges are active. Appendix B derives these formulas. For modular weights, each representation uses its own phase-equivalence relation from Equation 17.

To compare with controlled-phase gates, take $P = |1\rangle\langle 1| = \operatorname{diag}(0, 1)$ in eigenbasis coordinates and write $\Lambda = \lambda_0 I_2 + (\lambda_1 - \lambda_0) P$. Let Π_f be the embedding of P on f, with identities on the remaining qubits and $\Pi_\varnothing = I_D$.

Proposition 5 (Equivalent projector coordinates)**.** *Define*

$$\theta_f = (\lambda_1 - \lambda_0)^{|f|} \sum_{e \supseteq f} \lambda_0^{|e|-|f|} w_e, \qquad f \subseteq V, \tag{24}$$

where $\lambda_0^0 = 1$*, also when* $\lambda_0 = 0$*. Then*

$$U_A(\boldsymbol{w}) = e^{i\theta_\varnothing} Q^{\otimes N} \left(\prod_{\varnothing \neq f \subseteq V} e^{i\theta_f \Pi_f} \right) (Q^\dagger)^{\otimes N}. \tag{25}$$

The nonempty-coordinate map is invertible over $\mathbb{R}$*.*

The proof and inverse appear in Appendix B.1. The gate $e^{i\theta_f \Pi_f}$ applies phase θ_f when every qubit in f is one. Thus the original weights and the projector coefficients describe the same operator in different interaction coordinates. Its nonempty projector residues $[\boldsymbol{\theta}_\star]_{2\pi}$ are unique for the fixed ordered basis. In a sparse or restricted model, the choice of local matrix determines which interactions can be expressed compactly.

5.2 Operator choice adapts hypergraph sparsity

The conversion gives an exponential difference in nonzero coefficient counts for a single interaction. Consider

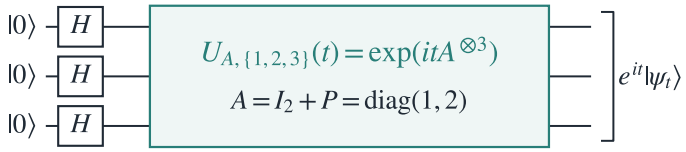
$$P = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad A = I_2 + P = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}. \tag{26}$$

For one edge e of size k with real weight t, expanding the local sums gives

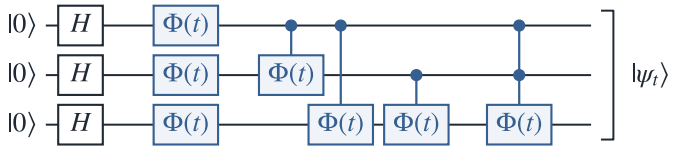
$$A_e = \prod_{q \in e} (I_D + \Pi_{\{q\}}) = \sum_{f \subseteq e} \Pi_f, \qquad e^{itA_e} = e^{it} \prod_{\varnothing \neq f \subseteq e} e^{it\Pi_f}. \tag{27}$$

Thus one nonzero A-edge produces $2^k - 1$ nonempty projector coefficients, all equal to t. When $t \not\equiv 0 \pmod{2\pi}$, all remain active phase weights: k are single-qubit phases and $2^k - k - 1$ join multiple qubits. The empty coefficient is t and contributes only the displayed global phase.

For $N = 3$ and $e = \{1, 2, 3\}$, Figure 2 shows the resulting circuits. A single three-qubit A-interaction equals three single-qubit phase gates, three two-qubit controlled-phase gates, and one three-qubit controlled-phase gate. Their actions agree up to global phase for every input. The figure shows the shared product input. Let $|\psi_t\rangle$ denote the expanded circuit's output. The expanded description has seven nonzero weights and one independent parameter, because all seven are tied to t.



(a) Shifted-projector coordinates: one nonzero weight



(b) Projector coordinates: seven nonzero weights, all equal to t

Figure 2: Different hypergraph coordinates give the same operator up to global phase. With $A = I_2 + P$, circuit (a) uses one nonzero encoder weight. Circuit (b) expands it into seven projector phase gates. Wires follow vertex order from top to bottom. Each H prepares $|+\rangle$ from $|0\rangle$. The gate $\Phi(t) = \mathrm{diag}(1, e^{it})$ applies a phase to $|1\rangle$. Dots condition it on the other participating qubits also being 1. Both outputs have the same density matrix because they differ only by e^{it}. The seven weights in (b) share the same parameter. The large block in (a) denotes the specified interaction. Its hardware decomposition depends on the available gate set.

The advantage is *representation sparsity*: selecting a local operator can collect a structured family of phase terms into fewer weighted edges. It also supplies a parameter-sharing pattern for fitting restricted models. The transformation is reversible, so a sparse projector model can conversely be dense in another choice. Appendix B gives the general affine change of coordinates. Both choices in Equation 26 have consecutive integer eigenvalues, so both retain branch-independent weight recovery modulo 2π and recovery over every discrete alphabet.

Changing Q instead conjugates the operator by a product of local unitaries.

5.3 Established states, gates, and inverse transforms

For the chosen eigendecomposition of A, the reference product input

$$|s_A\rangle = Q^{\otimes N} |+\rangle^{\otimes N} \tag{28}$$

cancels the initial basis change in Equation 10. Its output is

$$U_A(\boldsymbol{w}) |s_A\rangle = D^{-1/2} Q^{\otimes N} \sum_{\boldsymbol{x}} e^{i\phi_{\boldsymbol{x}}} |\boldsymbol{x}\rangle . \tag{29}$$

Table 1: Specializations of Q-WHO. All rows use $Q = I_2$. The state constructions use $|+\rangle^{\otimes N}$, while the gate construction permits any input.

Construction	Local Matrix	Hyperedges and weights
Graph states	P	Pairwise edges with weights 0 or π
Hypergraph states	P	Arbitrary edges with weights 0 or π
Weighted graph states	P	Pairwise edges with phase weights in $[0, 2\pi)$
Weighted hypergraph states	P	Arbitrary edges with phase weights in $[0, 2\pi)$
Commuting Z-string rotations	Z	Arbitrary edges with real weights and gate angles $-2w_e$

The same Q is used in the input and the operator factorization. The input has full support, and its equal eigenbasis amplitudes reduce Equation 14 to $y_{\boldsymbol{x}}/y_{\mathbf{0}}$. Table 1 lists established constructions obtained by choosing A, the allowed edges, and their weights.

Projector specialization. For $A = P$, use $(\lambda_0, \lambda_1) = (0, 1)$. If $\phi(X)$ denotes the phase at the configuration with support $X \subseteq V$, then

$$\phi(X) = \sum_{e \subseteq X} w_e, \qquad w_e = \sum_{X \subseteq e} (-1)^{|e|-|X|} \phi(X). \tag{30}$$

These are the projector forms of the general forward and inverse transforms. Since $B_P = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ has determinant one, the inverse also holds modulo 2π and over every $\mathbb{Z}_T$. Binary weights give graph and hypergraph states (Qu et al., 2013b; Rossi et al., 2013), and continuous phase weights give their weighted counterparts (Hartmann et al., 2007; Qu et al., 2013a).

Pauli-Z specialization. For $A = Z = \mathrm{diag}(1, -1)$, use $(\lambda_0, \lambda_1) = (1, -1)$ and $Q = I_2$. The spectral transform becomes

$$\phi(X) = \sum_{e \subseteq V} w_e (-1)^{|e \cap X|}, \qquad \boldsymbol{w} = 2^{-N} B_Z^{\otimes N} \boldsymbol{\phi}, \quad B_Z = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{31}$$

This is the Fourier–Walsh representation used for diagonal operators (Welch et al., 2014). Write Z_e for the embedding in Equation 3 with $A = Z$, and define the Pauli-Z string rotation $R_{Z_e}(\vartheta) := \exp(-i\vartheta Z_e/2)$. Each edge gate then satisfies

$$U_{Z,e}(w_e) = \exp(i w_e Z_e) = R_{Z_e}(-2w_e). \tag{32}$$

The full operator is a product of commuting Pauli-Z string rotations. Its spectral gap is -2, so T-ary weight recovery is bijective for odd T. Continuous weights modulo 2π are not uniquely determined without a further restriction such as Ω_Z.

6 Conclusion

We presented Q-WHO as a framework for encoding weighted hypergraphs into quantum circuits and decoding their weights from known input-output pairs. A shared Hermitian building block preserves the organization of hyperedge interactions and yields a common tensor spectral transform. Its inverse establishes exact weight recovery conditions for real, continuous-phase, and discrete weights. The tensor factorization also supports in-place state-vector computation. Operator choice changes interaction coordinates and can reduce the number of nonzero hyperedges without changing the circuit action up to global phase. Established graph, hypergraph, and commuting Pauli constructions follow as special cases.

Future Work. In quantum machine learning (QML), Q-WHO could provide layers for quantum hypergraph neural networks with trainable interaction weights. Sparse hypergraphs or those with bounded hyperedge size would specify which variable groups interact, and learning the local matrix could adapt their representation to data. Evaluating these models on defined learning tasks with classical and quantum baselines is a natural next step.

AI use statement

Generative AI assisted with English-language editing, mathematical exposition, and consistency checks during this revision.

Reproducibility statement

Sections 3–4 specify the representation, classical algorithms, and recovery conditions. Section 5 gives the coordinate transformations and specializations. The appendices provide proofs and the Kronecker multiplication routine. No external dataset or trained parameter is used.

References

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are AllSet: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=hpBTIv2uy_E.

L Hartmann, J Calsamiglia, W Dür, and H J Briegel. Weighted graph states and applications to spin chains, lattices and gases. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 40(9): S1–S44, apr 2007. doi: 10.1088/0953-4075/40/9/S01. URL https://doi.org/10.1088/0953-4075/40/9/S01.

M. Hein, J. Eisert, and H. J. Briegel. Multiparty entanglement in graph states. *Phys. Rev. A*, 69: 062311, Jun 2004. doi: 10.1103/PhysRevA.69.062311. URL https://link.aps.org/doi/10.1103/PhysRevA.69.062311.

Emanuele Mule, Stefano Fiorini, Antonio Purificato, Federico Siciliano, Stefano Coniglio, and Fabrizio Silvestri. Directional sheaf hypergraph networks: Unifying learning on directed and undirected hypergraphs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=d1hQkcDy7c.

Davide Poderini, Dagmar Bruß, and Chiara Macchiavello. Quantum hypergraph states: a review. *Reports on Progress in Physics*, 89(6):066001, jun 2026. doi: 10.1088/1361-6633/ae7735. URL https://doi.org/10.1088/1361-6633/ae7735.

Ri Qu, Yi-ping Ma, Bo Wang, and Yan-ru Bao. Relationship among locally maximally entangleable states, W states, and hypergraph states under local unitary transformations. *Phys. Rev. A*, 87: 052331, May 2013a. doi: 10.1103/PhysRevA.87.052331. URL https://link.aps.org/doi/10.1103/PhysRevA.87.052331.

Ri Qu, Juan Wang, Zong-shang Li, and Yan-ru Bao. Encoding hypergraphs into quantum states. *Phys. Rev. A*, 87:022311, Feb 2013b. doi: 10.1103/PhysRevA.87.022311. URL https://link.aps.org/doi/10.1103/PhysRevA.87.022311.

M Rossi, M Huber, D Bruß, and C Macchiavello. Quantum hypergraph states. *New Journal of Physics*, 15(11):113022, nov 2013. doi: 10.1088/1367-2630/15/11/113022. URL https://doi.org/10.1088/1367-2630/15/11/113022.

F. E. S. Steinhoff, C. Ritz, N. I. Miklin, and O. Gühne. Qudit hypergraph states. *Phys. Rev. A*, 95: 052340, May 2017. doi: 10.1103/PhysRevA.95.052340. URL https://link.aps.org/doi/10.1103/PhysRevA.95.052340.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=RiTjKoscnNd.

Yanbang Wang and Jon Kleinberg. From graphs to hypergraphs: Hypergraph projection and its reconstruction. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=qwYKE3VB2h.

Jonathan Welch, Daniel Greenbaum, Sarah Mostame, and Alan Aspuru-Guzik. Efficient quantum circuits for diagonal unitaries without ancillas. *New Journal of Physics*, 16(3):033040, mar 2014. doi: 10.1088/1367-2630/16/3/033040. URL https://doi.org/10.1088/1367-2630/16/3/033040.

Fei-Lei Xiong, Yi-Zheng Zhen, Wen-Fei Cao, Kai Chen, and Zeng-Bing Chen. Qudit hypergraph states and their properties. *Phys. Rev. A*, 97:012323, Jan 2018. doi: 10.1103/PhysRevA.97.012323. URL https://link.aps.org/doi/10.1103/PhysRevA.97.012323.

Xinyu Ye, Hao Xiong, Jianhao Huang, Ziang Chen, Jia Wang, and Junchi Yan. On designing general and expressive quantum graph neural networks with applications to MILP instance representation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=IQi8JOqLuv.

Roberto Zucchini. Calibrated hypergraph states: I calibrated hypergraph and multi qudit state monads. *Journal of Mathematical Physics*, 67(7):072202, 07 2026a. ISSN 0022-2488. doi: 10.1063/5.0264175. URL https://doi.org/10.1063/5.0264175.

Roberto Zucchini. Calibrated hypergraph states. ii. calibrated hypergraph state construction and applications. *Journal of Mathematical Physics*, 67(7):072203, 07 2026b. ISSN 0022-2488. doi: 10.1063/5.0264179. URL https://doi.org/10.1063/5.0264179.

A Proofs of the Main Results

A.1 Proof of Proposition 1

Since A is Hermitian and each w_e is real, every $w_e A_e$ is Hermitian. Hence each gate $U_{A,e}(w_e)$ is unitary, as is their product $U_A(\boldsymbol{w})$. At any vertex q, the local factors of A_e and A_f belong to $\{I_2, A\}$ and commute. Tensor products of these local factors therefore satisfy $A_e A_f = A_f A_e$. The exponential of a sum of mutually commuting matrices equals the product of their exponentials in any order. Using the definitions of H_A and U_A gives $U_A(\boldsymbol{w}) = \exp(iH_A(\boldsymbol{w}))$ for any permutation of the gates.

A.2 Proof of Proposition 2

Conjugating Equation 3 by $(Q^\dagger)^{\otimes N}$ on the left and $Q^{\otimes N}$ on the right gives

$$(Q^\dagger)^{\otimes N} A_e Q^{\otimes N} = \bigotimes_{q=1}^{N} \Lambda^{[\chi_e(q)]}, \tag{33}$$

where $\Lambda^{[0]} = I_2$ and $\Lambda^{[1]} = \Lambda$. This matrix is diagonal. Its entry indexed by $\boldsymbol{x}$ is $\prod_{q \in e} \lambda_{x_q}$. Summing with coefficients w_e gives Equation 7. For one vertex, the map from coefficients $(w_0, w_1)^\mathsf{T}$ to values at eigenvalues (λ_0, λ_1) is multiplication by B_A. Taking the tensor product over N vertices gives Equation 9. Exponentiating the diagonal generator and conjugating back gives Equation 10.

A.3 Relative inverse and phase ambiguity

Let $L_\star$ be L_A with its empty-edge column removed, and let $\boldsymbol{p}^\mathsf{T}$ be the reference row of $L_\star$. Subtracting the reference row of L_A from every other row gives

$$\begin{bmatrix} 1 & \boldsymbol{p}^\mathsf{T} \\ 0 & R_A \end{bmatrix}. \tag{34}$$

The row operations preserve the determinant, so

$$\det R_A = \det L_A = (\lambda_1 - \lambda_0)^{N 2^{N-1}} \neq 0. \tag{35}$$

For any $\boldsymbol{v} \in \mathbb{R}^{D-1}$, reference subtraction gives

$$\mathsf{Pad}(R_A \boldsymbol{v}) = L_\star \boldsymbol{v} - (\boldsymbol{p}^\mathsf{T} \boldsymbol{v}) \mathbf{1}. \tag{36}$$

The first column of L_A is $\mathbf{1}$. Hence $L_A^{-1} L_\star \boldsymbol{v} = \mathsf{Pad}(\boldsymbol{v})$ and $L_A^{-1}\mathbf{1}$ is supported only on the empty coordinate. Applying $\mathsf{Drop}\, L_A^{-1}$ proves Equation 16.

Two operators diagonal in the same basis differ by a global phase exactly when all their relative spectral phases agree modulo 2π. For their nonempty weights, this condition is $R_A(\boldsymbol{w}'_\star - \boldsymbol{w}_\star) \in 2\pi\mathbb{Z}^{D-1}$, which proves Equation 17. The cube $[0, 2\pi)^{D-1}$ selects one representative of each relative phase class, and its inverse image is the fundamental domain Ω_A. Selecting a different reference configuration or phase interval may select different representatives of the same operator class.

A.4 Proof of Theorem 4

Theorem 3 supplies the wrapped relative phases. If $\boldsymbol{w}_\star \in \Omega_A$, their representatives in $[0, 2\pi)^{D-1}$ are the intended unwrapped relative phases. The inverse from Equation 16 therefore recovers the original real weights.

For integer eigenvalues, R_A is an integer matrix and defines a map on the coefficient torus $(\mathbb{R}/2\pi\mathbb{Z})^{D-1}$. This map is bijective exactly when $\det R_A = \pm 1$. Indeed, a unimodular integer matrix has an integer inverse, which is independent of the chosen phase representatives modulo 2π. Conversely, if $|\det R_A| > 1$, the quotient $\mathbb{Z}^{D-1}/R_A\mathbb{Z}^{D-1}$ has $|\det R_A|$ elements. Applying $2\pi R_A^{-1}$ identifies these with distinct weight residues in the kernel of the torus map, so the map is not injective. Equation 35 makes bijectivity equivalent to $|\lambda_1 - \lambda_0| = 1$.

Over $\mathbb{Z}_T$, a square matrix is invertible exactly when its determinant is a unit. By Equation 35, this holds exactly when $\gcd(\lambda_1 - \lambda_0, T) = 1$. In this case the row-subtraction proof above works over $\mathbb{Z}_T$, giving Equation 20. The local inverse is

$$B_{A,T}^{-1} = g \begin{bmatrix} \lambda_1 & -\lambda_0 \\ -1 & 1 \end{bmatrix} \pmod T, \qquad g(\lambda_1 - \lambda_0) \equiv 1 \pmod T. \tag{37}$$

This argument does not require T to be prime.

B Coordinate Changes and Structural Identities

Let C be a nondegenerate Hermitian 2×2 matrix and let $A = \alpha I_2 + \beta C$, with real α and $\beta \neq 0$. Use the same ordered eigenbasis for both. This includes any two nondegenerate local operators with a common ordered eigenbasis. Let C_f be the embedding of C on f, defined as in Equation 3, and let $\boldsymbol{v}$ denote coefficients in this new representation. Expanding each tensor factor gives

$$A_e = \sum_{f \subseteq e} \alpha^{|e|-|f|} \beta^{|f|} C_f. \tag{38}$$

Collecting the coefficients of C_f gives Equation 22. Consequently $H_A(\boldsymbol{w}) = H_C(\boldsymbol{v})$ and $U_A(\boldsymbol{w}) = U_C(\boldsymbol{v})$ when the transformed empty coefficient is retained. If the input gauge is $w_\varnothing = 0$ and $\boldsymbol{v}^{(0)} = \mathsf{Pad}(\boldsymbol{v}_\star)$, then

$$U_A(\boldsymbol{w}) = e^{iv_\varnothing} U_C(\boldsymbol{v}^{(0)}). \tag{39}$$

The inverse in Equation 23 follows by substituting $C = \beta^{-1}A - \alpha\beta^{-1}I_2$. For a generator with at least one nonempty term, the triangular map preserves the maximum hyperedge size among terms with nonzero real coefficients, but can change its support. Support after reduction modulo a phase period can shrink further.

B.1 Proof of Proposition 5

In the local eigenbasis, $\Lambda = \lambda_0 I_2 + (\lambda_1 - \lambda_0)P$. Let $\Pi_{\{q\}}$ denote P on qubit q and the identity elsewhere. Then, for every edge e,

$$(Q^\dagger)^{\otimes N} A_e Q^{\otimes N} = \prod_{q \in e} \left[\lambda_0 I_D + (\lambda_1 - \lambda_0)\Pi_{\{q\}}\right] = \sum_{f \subseteq e} \lambda_0^{|e|-|f|} (\lambda_1 - \lambda_0)^{|f|} \Pi_f. \tag{40}$$

Multiplying by w_e, summing over e, and collecting equal projectors yields

$$(Q^\dagger)^{\otimes N} H_A(\boldsymbol{w}) Q^{\otimes N} = \sum_f \theta_f \Pi_f,$$

which is Equation 24. All Π_f commute, so exponentiation factorizes, proving Equation 25. For the inverse, specialize Equation 23 to $A = \Lambda$, $C = P$, $\alpha = \lambda_0$, and $\beta = \lambda_1 - \lambda_0$, giving

$$w_e = \sum_{f \supseteq e} (-\lambda_0)^{|f|-|e|} (\lambda_1 - \lambda_0)^{-|f|} \theta_f, \qquad e \neq \varnothing. \tag{41}$$

The coefficient of w_f in θ_f is $(\lambda_1 - \lambda_0)^{|f|}$, so the equivalent triangular map has nonzero diagonal exactly when $\lambda_0 \neq \lambda_1$. The empty-edge coefficient appears only in $\theta_\varnothing$, with unit coefficient.

B.2 Structural statements and proofs

Let π be a permutation of V, let P_π permute the corresponding tensor factors, and define the relabeled weights by $w^{\pi}_{\pi(e)} = w_e$.

Proposition 6 (Permutation covariance)**.** *For every vertex permutation π,*

$$H_A(\pi G) = P_\pi H_A(G) P_\pi^\dagger, \qquad U_A(\pi G) = P_\pi U_A(G) P_\pi^\dagger. \tag{42}$$

Thus isomorphic hypergraphs have permutation-similar Q-WHO *representations. If π is an automorphism, then $[U_A(G), P_\pi] = 0$.*

The identity $P_\pi A_e P_\pi^\dagger = A_{\pi(e)}$ gives $P_\pi H_A(G) P_\pi^\dagger = \sum_e w_e A_{\pi(e)} = H_A(\pi G)$. Unitary conjugation commutes with the matrix exponential, proving the unitary identity.

Proposition 7 (Disjoint-union factorization)**.** *Let G_1 and G_2 have disjoint vertex sets and no crossing hyperedge. Under the tensor ordering V_1 followed by V_2,*

$$U_A(G_1 \uplus G_2) = U_A(G_1) \otimes U_A(G_2). \tag{43}$$

Writing $N_j = |V_j|$ for $j = 1, 2$ gives $H_A(G_1 \uplus G_2) = H_A(G_1) \otimes I_{2^{N_2}} + I_{2^{N_1}} \otimes H_A(G_2)$. The summands commute, and exponentiating them separately proves Equation 43.

C Implementation Details

C.1 In-place Kronecker multiplication

Algorithm 3 evaluates $C^{\otimes N}\boldsymbol{u}$ without constructing the $D \times D$ matrix. At each stage it applies C to pairs of entries whose binary indices differ at one bit position. Stages acting on distinct bit positions commute, so they can proceed from the least significant bit to the most significant bit.

Algorithm 3 In-place Kronecker multiplication $\mathsf{KronMatVec}(C, \boldsymbol{u})$

Input: a 2×2 matrix C and a length-$D = 2^N$ buffer $\boldsymbol{u}$, with zero-based indices.
Output: the same buffer containing $C^{\otimes N}\boldsymbol{u}$.

For $h = 1, 2, 4, \ldots, D/2$:
  For $r = 0, 2h, 4h, \ldots, D - 2h$:
    For $j = 0, 1, \ldots, h - 1$:
      Read $(\alpha, \beta) \leftarrow (u_{r+j}, u_{r+j+h})$.
      Set $u_{r+j} \leftarrow C_{00}\alpha + C_{01}\beta$ and $u_{r+j+h} \leftarrow C_{10}\alpha + C_{11}\beta$. Reduce modulo T when working over $\mathbb{Z}_T$.
Return $\boldsymbol{u}$.

The routine uses $ND/2$ pair updates and two scalar temporaries in addition to its buffer. For fixed A and weights, the eigendecomposition and phase table can be reused across inputs. For fixed A with changing weights, only the phase table needs to be recomputed. If an input such as Equation 28 is constructed from Q, the same decomposition is reused to construct that input and evaluate its output.