Hamiltonian Encoding

Variational Quantum Algorithms · Module 2 — Parameterized Circuits

Node 2.5

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Parameterized Circuits **Node:** 2.5

The Hamiltonian H is the **definition of the problem**. But "the Hamiltonian" lives in different mathematical worlds depending on where you are in the pipeline:

- The chemist writes H in second-quantized form, using fermionic creation/annihilation operators.
- The condensed matter physicist writes H in spin variables, using Pauli operators directly.
- The combinatorial optimizer writes H as an Ising or QUBO cost.
- The VQA practitioner needs H as a **sum of Pauli strings on qubits**, because that is what the device can measure.

This node is about the **encoding step** — the translation from problem-native representation to a Pauli decomposition that a quantum device can act on. It is conceptually distinct from the ansatz (which prepares a state) and conceptually distinct from the optimizer (which acts on a scalar). The encoding is the third design choice that determines what the optimizer sees.

The Required Output

What every VQA needs is a representation of H of the form

$$H = \sum_{j=1}^{M} h_j P_j,$$

where each $h_j \in \mathbb{R}$ is a coefficient and each P_j is a tensor product of single-qubit Pauli operators (and identities) on n qubits — i.e., a "Pauli string."

This form is required because it is the only thing a quantum device can directly measure. Each Pauli string can be measured by rotating each qubit into the appropriate basis (Z for I/Z entries, H rotation for X entries, S†H rotation for Y entries) and reading out in the computational basis. The measured eigenvalue is the product of ± 1 outcomes per qubit.

Once H is in this form, the cost function

$$C(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle = \sum_j h_j \langle \psi(\boldsymbol{\theta}) | P_j | \psi(\boldsymbol{\theta}) \rangle$$

becomes a weighted sum of Pauli expectation values, each of which can be estimated by repeated measurement (see node 2.7).

The encoding step is therefore: **rewrite the problem Hamiltonian, in whatever form it started, as a Pauli sum.**

Encoding Spin Hamiltonians

For spin systems, the encoding is essentially trivial — the Hamiltonian is **already** a Pauli sum.

The transverse-field Ising model on n sites:

$$H = -J \sum_{\langle i,j \rangle} Z_i Z_j - h \sum_i X_i.$$

This is already in the required form. $M = |\text{edges}| + n$ terms, each a 1- or 2-local Pauli string.

The Heisenberg model:

$$H = J \sum_{\langle i,j \rangle} (X_i X_j + Y_i Y_j + Z_i Z_j).$$

Also already in the required form, with $M = 3|\text{edges}|$ terms.

For these problem classes, encoding is no work and the Pauli sum is short — $M = O(n)$. The qubit count equals the spin count.

Encoding Combinatorial Optimization

For QUBO problems (Quadratic Unconstrained Binary Optimization), the cost function has the form

$$f(\mathbf{x}) = \sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j, \quad x_i \in \{0, 1\}.$$

To encode this as a quantum Hamiltonian, substitute $x_i = (1 - Z_i)/2$ so that $x_i = 0 \leftrightarrow Z_i = +1$ and $x_i = 1 \leftrightarrow Z_i = -1$. Expanding gives an Ising-like Hamiltonian:

$$H = c_0 + \sum_i c_i Z_i + \sum_{i<j} c_{ij} Z_i Z_j.$$

This is already a Pauli sum. $M = O(n^2)$ for dense problems, much less for sparse ones. The qubit count equals the number of binary variables. QAOA uses exactly this encoding.

Encoding Fermionic Hamiltonians

This is where the encoding step earns its name. A second-quantized fermionic Hamiltonian for an electronic structure problem looks like

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s,$$

where $a_p^\dagger, a_q$ are fermionic creation/annihilation operators and the coefficients come from one- and two-electron integrals over molecular orbitals.

Fermions are not the same as qubits — the operators $a_p^\dagger, a_q$ **anticommute**, while qubit operators on different qubits **commute**. Translating between these algebras is the work of a **fermion-to-qubit mapping**.

There are three standard mappings.

Jordan-Wigner

Map each spin-orbital to one qubit. Define:

$$a_p^\dagger = \frac{1}{2}(X_p - iY_p) \otimes Z_{p-1} \otimes Z_{p-2} \otimes \cdots \otimes Z_0.$$

The trailing Z chain is what enforces fermionic anticommutation: each fermion operator carries a "parity string" through all earlier orbitals. This is the simplest mapping, conceptually: **one fermion per qubit**. Its weakness is that the Pauli strings are *long* — operators acting on widely separated orbitals produce Pauli strings spanning all qubits in between. For sparse fermionic Hamiltonians, JW gives long strings.

Bravyi-Kitaev

Use a more clever data structure (a binary tree of qubits) to encode parity information. Each fermion operator becomes a Pauli string of length $O(\log n)$ instead of $O(n)$. Trade-off: the implementation is more complex, and the resulting Pauli strings are *shorter on average* but spread across non-adjacent qubits, which can hurt on devices with local connectivity.

Parity Encoding

Encode the *partial sums of occupation numbers* rather than the occupations themselves. Useful in some contexts, less common in practice than JW or BK.

The choice between JW and BK matters in practice because it determines:

- Number of Pauli terms in the Hamiltonian (typically $O(n^4)$ for either, but with different prefactors).
- Locality of the Pauli strings (JW: linear; BK: $\log n$).
- Which strings can be measured simultaneously (see node 2.8 on measurement grouping).

Standard libraries (OpenFermion, Qiskit Nature) implement both and let you switch with a single function call.

Symmetry Reductions

After mapping to qubits, you can often **further reduce the qubit count** by exploiting symmetries.

For H_2 in a minimal STO-3G basis: - Naive Jordan-Wigner: 4 spin-orbitals $\to$ 4 qubits. - After exploiting parity (Z2 symmetries arising from spin and electron number conservation): **2 qubits**.

This is "qubit tapering" and is implemented in standard libraries. It reduces both the device cost and the parameter count of the ansatz, often by a factor of 2-4. For larger molecules the savings can be larger.

The cost is conceptual: the resulting 2-qubit Hamiltonian no longer has a clean physical interpretation as "electrons in orbitals." It's just a Pauli sum.

What The Optimizer Sees After Encoding

Once H is encoded, the optimizer sees a single scalar function $C(\boldsymbol{\theta})$ — *the encoding has disappeared.* The choice of mapping, the symmetry reductions, the qubit count: all of these are "compiled away" from the optimizer's perspective.

But the encoding choice has determined:

- **The number of Pauli terms**, which sets the per-evaluation shot cost (one shot budget per term, see node 2.8).
- **The variance structure** of the cost estimator, since variance depends on which Pauli terms are easy to measure jointly.
- **The qubit count**, which sets the depth budget for the ansatz.
- **The barren plateau profile**, indirectly, since fewer qubits means smaller Hilbert space and milder concentration.

So the encoding is invisible to the optimizer but it shapes everything the optimizer experiences. This is one of the places where Module 1's "stochastic objective generator" abstraction (1.8) is **leaky** — the noise structure of the generator is set by encoding decisions made elsewhere, and ignoring those decisions misses a major lever.

Structural Role

This node closes out the "what is the problem definition?" half of Module 2. Together with 2.1-2.3 (ansatz design) it answers: *what does the optimizer minimize, and over what?*

Forwards: - 2.7 (how measurement actually works) consumes Pauli strings as input. - 2.8 (measurement grouping and shot budget) consumes the *number* and *commutation structure* of the Pauli strings. - Module 9 (hardware noise models) consumes encoded Hamiltonians to study how noise propagates into objective error.

Relations to Other Concepts

- **Second quantization** — the standard formalism for many-body fermionic and bosonic systems.
- **Jordan-Wigner transformation** — historically the first fermion-to-qubit map, dating to 1928.
- **Bravyi-Kitaev transformation** — modern, more efficient mapping using parity trees.
- **Z2 symmetries / qubit tapering** — exploit conserved quantities to reduce qubit count.
- **Block encoding** — a different paradigm: encode H as a unitary block within a larger circuit. Used in fault-tolerant algorithms (qubitization), not VQAs.

Worked Example — H_2 End-to-End

Starting representation: second-quantized fermionic Hamiltonian for H_2 in STO-3G basis at 0.74 Å bond length.

After computing one- and two-electron integrals (PySCF or similar), the fermionic Hamiltonian has ~30 nonzero coefficients across ~8 distinct fermionic operators.

After Jordan-Wigner mapping to 4 qubits, this becomes a sum of **15 Pauli strings**:

$$H = -1.05 \cdot I - 0.39 \cdot Z_0 + 0.39 \cdot Z_1 - 0.01 \cdot Z_2 + \cdots + 0.18 \cdot X_0X_1Y_2Y_3 + \ldots$$

(approximate coefficients).

After Z2 symmetry tapering: **5 Pauli strings on 2 qubits**.

The optimizer doesn't know any of this. It sees a function from $\mathbb{R}^P$ to $\mathbb{R}$ and asks for evaluations. But each evaluation costs at least 5 measurement bases $\times$ N shots each. And the optimal ansatz on 2 qubits is much smaller than the optimal ansatz on 4 qubits. **The encoding choices set the cost basis and the design space.**

Visualization

Pending. A flow diagram. Top: "Fermionic Hamiltonian (8 distinct operators)". Middle: "Jordan-Wigner (15 Pauli strings, 4 qubits)". Bottom: "Symmetry-tapered (5 Pauli strings, 2 qubits)". Side annotations: "this affects ansatz design", "this affects measurement cost", "this determines what the optimizer sees".

Exercises

1. Apply the Jordan-Wigner transformation to the fermionic operator $a_2^\dagger a_0$ for a 4-orbital system. Write out the resulting Pauli string explicitly.
2. The Jordan-Wigner mapping gives $O(n)$-length Pauli strings; Bravyi-Kitaev gives $O(\log n)$. Estimate the asymptotic crossover qubit count where BK becomes preferable for measurement cost. (Hint: depends on the connectivity of the device and the dominant cost.)
3. (VQA) For a 2-qubit Pauli sum $H = c_0 II + c_1 ZI + c_2 IZ + c_3 ZZ + c_4 XX$, how many distinct measurement bases are needed to estimate $\langle H \rangle$? Which terms can be measured simultaneously?

Research Extensions

- **Custom fermion-to-qubit mappings** — tailoring the mapping to a specific device topology to minimize compiled circuit depth.
- **Locality-preserving mappings** — encoding Hamiltonians so that nearby orbitals map to nearby qubits.
- **Tapering automation** — automatic discovery of Z2 symmetries in arbitrary fermionic Hamiltonians.
- **Bosonic encodings** — encoding bosonic systems (vibrational modes, photonic networks) into qubits with bounded truncation error.

Cross-links to Other Courses

- **Quantum Chemistry** — second quantization, integrals, basis sets.
- **Group Theory** — Z2 symmetries, conserved quantities.
- **Module 8** — encoding choice affects expressivity.
- **Module 9** — encoding choice affects noise propagation.

Variational Quantum Algorithms · Module 2 — Parameterized Circuits · Node 2.5
Concepts: tensor-product, linear-operators, matrix-representation, expectation-value
Prerequisites: 2.1, 2.3
Enables: 2.7, 2.8

hbar.university — own.your.cognition