The Quantum-Classical Architecture

Variational Quantum Algorithms · Module 1 — Hybrid Quantum Classical Systems

Node 1.2

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Hybrid Quantum-Classical Systems **Node:** 1.2

This node describes the **operational shape** of a variational quantum algorithm: two coupled layers, one quantum and one classical, exchanging information through a tight feedback loop. Everything in Module 1 from here on lives inside one or the other of these two layers.

Node 1.1 justified *why* we care about the number $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$. This node describes *the machine that computes it and decides where to go next.*

The Two Layers

A variational quantum algorithm is built from two distinct subsystems:

The quantum layer. Takes a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^P$ as input. Prepares a parameterized state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}$ on a real or simulated quantum device. Measures a Hamiltonian H to return a noisy scalar estimate $\widehat{C}(\boldsymbol{\theta}) \approx \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$. That is its entire job.

The classical layer. Takes the current parameter vector $\boldsymbol{\theta}_k$ and the scalar feedback $\widehat{C}(\boldsymbol{\theta}_k)$ as input. Produces a new parameter vector $\boldsymbol{\theta}_{k+1}$ according to some update rule:

$$\boldsymbol{\theta}_{k+1} = \mathcal{A}\big(\boldsymbol{\theta}_k, \widehat{C}(\boldsymbol{\theta}_k), \mathcal{H}_k\big),$$

where $\mathcal{A}$ is the optimizer and $\mathcal{H}_k$ is its internal memory (momentum, particle populations, trust-region radii, whatever). That is **its** entire job.

Neither layer understands the other directly. The quantum layer does not know it is being optimized; it is a stateless oracle. The classical layer does not know what is happening inside the quantum device; it sees only numbers.

The Feedback Loop

Execution alternates between the two layers. At iteration k:

1. The classical layer hands $\boldsymbol{\theta}_k$ to the quantum layer.
2. The quantum layer prepares $|\psi(\boldsymbol{\theta}_k)\rangle$, measures, and returns $\widehat{C}(\boldsymbol{\theta}_k)$.
3. The classical layer consumes the feedback, updates its internal state, and produces $\boldsymbol{\theta}_{k+1}$.
4. Repeat.

This is a **closed-loop discrete-time control system**. The "plant" is the quantum objective oracle. The "controller" is the classical optimizer. The "reference signal" is implicit: reduce $\widehat{C}$.

Framing VQA this way is not just metaphor. It means you can import every concept from classical control theory — stability, observability, disturbance rejection — and apply it to VQA directly. Module 7 develops this in detail.

Why Hybrid and Not All-One-Or-The-Other

Why not fully quantum? A fully quantum algorithm for ground-state estimation (quantum phase estimation, for example) requires circuit depth and coherence time that NISQ devices do not have. Fault-tolerant quantum computing is the long game. VQAs are the near-term workaround: use a shallow quantum circuit to generate a stochastic cost, and spend the classical compute budget on optimization.

Why not fully classical? A fully classical simulation of $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ requires storing the state vector, which is exponential in qubit count. For 50+ qubits this is infeasible. The whole point of running the circuit on quantum hardware is that state preparation is efficient on a quantum device regardless of qubit count — the exponential cost shows up in **measurement**, not in **storage**.

The hybrid design is the minimum-viable use of quantum hardware given those two constraints. You run the part of the computation that classical computers cannot do (preparing a superposition of 2^n amplitudes) on the quantum device, and you run everything else — parameter selection, state tracking, decision logic — on a classical computer that is actually good at that.

Structural Role

This node is the **scaffold** onto which the rest of Module 1 bolts.

- Nodes 1.3, 1.4 zoom into the quantum layer and explain what "measuring" and "estimating a gradient" actually mean in practice.
- Node 1.5 zooms into the classical layer and describes what the optimization landscape looks like from the inside.
- Node 1.6 reframes the whole architecture as a system-identification problem.
- Nodes 1.7, 1.8, 1.9 re-describe the same architecture in the thesis vocabulary (epistemic vs ontological, objective generator, separation of concerns).

All of those nodes assume you understand the two-layer picture introduced here.

Relations to Other Concepts

- **Model-based vs model-free reinforcement learning** — VQA is model-free: the optimizer never builds a model of the quantum landscape, it just queries it.
- **Bayesian optimization** — a different family of hybrid systems where the classical side **does** build an explicit surrogate of the black-box function.
- **Simulation-based inference** — classical ML with a simulator as the oracle; structurally similar to VQA with the quantum device replaced by a (typically deterministic) simulator.
- **Control-plant decomposition** — the standard block-diagram of classical control theory maps onto VQA cleanly (see Module 7).

Worked Example — A Single Iteration

Suppose you have a 4-qubit hardware-efficient ansatz with 8 parameters. At iteration $k = 0$:

1. You initialize $\boldsymbol{\theta}_0$ randomly in $[-\pi, \pi]^8$.
2. You send $\boldsymbol{\theta}_0$ to the quantum layer.
3. The quantum layer compiles the circuit, runs 1024 shots, measures in the eigenbases of the Pauli decomposition of H, and returns $\widehat{C}(\boldsymbol{\theta}_0) \approx -0.42$.
4. The classical layer is running ADAM. It needs a gradient, so it queries the quantum layer 16 more times (parameter-shift rule, 2 evaluations per parameter, 8 parameters) to get $\widehat{\nabla} C(\boldsymbol{\theta}_0)$.
5. ADAM produces $\boldsymbol{\theta}_1$.

Iteration 0 has cost: 17 circuit evaluations, $17 \times 1024 = 17{,}408$ shots. That is the **per-iteration quantum resource budget**, and it is the real currency of VQA. (See node 1.4 for why the parameter-shift rule gives exact-up-to-noise gradients, and Module 2 for how to design circuits that make this cheap.)

Visualization

Pending. Block diagram: two boxes labeled "Classical Optimization Layer" (top) and "Quantum Objective-Generating Substrate" (bottom). Arrows: down-arrow labeled $\boldsymbol{\theta}_k$, up-arrow labeled $\widehat{C}(\boldsymbol{\theta}_k)$. Dashed box around both labeled "VQA".

Exercises

1. Given a 10-qubit ansatz with 30 parameters and a Hamiltonian with 50 Pauli terms, estimate the number of shots needed per iteration for a gradient-based optimizer using the parameter-shift rule. State your assumptions about shots-per-expectation.
2. Suppose the classical optimizer has unbounded compute but the quantum layer is rate-limited to 1000 circuit evaluations per hour. How does this constrain the choice of optimizer?
3. Why is it incorrect to say "the quantum device is just a subroutine that evaluates $C(\boldsymbol{\theta})$"? What does this framing hide about the cost and noise structure of VQA?

Research Extensions

- **Measurement-adaptive VQA** — closing a second feedback loop inside the quantum layer, where the choice of measurement basis depends on previous outcomes (see Module 10).
- **Latency-aware optimizer design** — choosing optimizers based on the classical-quantum communication overhead rather than iteration count.
- **Co-processing architectures** — tighter hardware integration between quantum and classical components (e.g., IBM's runtime primitives, Quantinuum's TKET runtime).

Cross-links to Other Courses

- **Control Theory** — the feedback-loop framing; see Module 7 for the full mapping.
- **Internet Systems** — request-response latency considerations for cloud-accessed quantum hardware.
- **Systems Thinking** — composition of subsystems with different time scales and observability.

Variational Quantum Algorithms · Module 1 — Hybrid Quantum Classical Systems · Node 1.2
Concepts: feedback, parameterized-circuits, cost-function, state-space

Prerequisites: 1.1
Enables: 1.3, 1.4, 1.5, 1.8, 1.9

hbar.university — own.your.cognition