Ansatz Design

Variational Quantum Algorithms · Module 2 — Parameterized Circuits

Node 2.1

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Parameterized Circuits **Node:** 2.1

This is the first node of Module 2 and the **anchor of the quantum half** of the course. Module 1 told you *that* the optimizer interacts with a parameterized state $|\psi(\boldsymbol{\theta})\rangle$. Module 2 starts here by asking the prior question: **what is $U(\boldsymbol{\theta})$, and how is it chosen?**

The answer determines almost everything downstream — what states are reachable, what the cost landscape looks like, whether a barren plateau will appear, and how many shots an optimizer will need. Choosing an ansatz is the single highest-leverage design decision in a VQA.

What an Ansatz Is

An **ansatz** is a parameterized recipe for preparing a family of quantum states. Formally, it is a unitary

$$U(\boldsymbol{\theta}) = \prod_{\ell=1}^{L} U_\ell(\boldsymbol{\theta}_\ell)$$

built from a sequence of layers U_ℓ, each of which depends on a subset of the parameter vector. Acting on a fixed reference state (typically $|0\rangle^{\otimes n}$), the ansatz defines

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}.$$

The set of states reachable by varying $\boldsymbol{\theta} \in \mathbb{R}^P$ is the **ansatz manifold** — a strict, structured subset of the full Hilbert space.

The word "ansatz" is borrowed from German theoretical physics and means "starting form" or "trial function." A variational quantum algorithm is, at its core, the act of letting an optimizer slide along this trial form looking for the best parameter assignment.

The Three Design Axes

Every ansatz design choice can be located along three axes.

1. Expressivity — *what states can it reach?* A more expressive ansatz covers more of Hilbert space. At the extreme, a fully universal ansatz can prepare any state, but at the cost of needing exponentially many parameters or exponentially deep circuits. At the other extreme, a 1-parameter ansatz can only prepare a 1-parameter family, regardless of how cleverly you optimize. The interesting region is in between: **just expressive enough to contain a good approximation to the target state, and no more.**

2. Trainability — *can the optimizer find a good point on the manifold?* A maximally expressive ansatz often has barren plateaus (Module 4), meaning the cost gradient vanishes over almost all of parameter space and the optimizer cannot make progress. A less expressive ansatz with the right inductive bias may be much easier to train, even if its theoretical reach is smaller. **Expressivity and trainability are in tension** — Module 8 is dedicated entirely to this tension.

3. Hardware compatibility — *can it run on the device you have?* Real near-term devices have a limited gate set (typically single-qubit rotations + a 2-qubit entangler such as CNOT or CZ), restricted qubit connectivity (only some pairs of qubits can be entangled directly), and short coherence times (gates must complete before noise destroys the state). An ansatz that is mathematically beautiful but compiles to a circuit that is too deep, requires too many SWAPs, or uses gates the device doesn't support is operationally useless.

A good ansatz design is a Pareto-optimal point on these three axes for the problem you actually care about.

The Two Cultural Camps

VQA ansatz design splits into two communities with very different starting points.

Hardware-first designers (node 2.2): build the ansatz from gates the hardware natively supports, lay them out in a regular pattern (often just "alternating layers of single-qubit rotations and entanglers"), and hope the resulting parameterization is rich enough to contain the answer. This is the **hardware-efficient ansatz**. Simple, agnostic to the problem, runs on anything. Frequently barren-plateau-prone.

Problem-first designers (node 2.3): start from physics or chemistry knowledge, write down a state-preparation recipe motivated by the problem structure (e.g., a unitary coupled-cluster expansion for molecules), then compile that down to hardware gates. This is the **chemically-inspired** or more generally **problem-inspired** ansatz. Better inductive bias, often deeper, often less hardware-friendly.

Both camps are correct about something the other camp is wrong about, and the right answer for any given VQA usually involves elements of both. Modules 2.2 and 2.3 examine each in detail.

What an Ansatz Is Not

It is worth being explicit about what the ansatz does not do.

- **An ansatz does not encode the Hamiltonian.** The Hamiltonian H is the *measurement*, applied to the state the ansatz prepares. The ansatz only prepares the state. (See node 2.5 for how the Hamiltonian is encoded — separately — into a measurement protocol.)
- **An ansatz does not adapt during a single run.** The structure of $U(\boldsymbol{\theta})$ is fixed throughout the optimization; only the parameters $\boldsymbol{\theta}$ change. *Adaptive ansatz methods* (ADAPT-VQE) violate this by growing the ansatz iteratively, but they are a special case treated separately.
- **An ansatz is not unique.** For any target state, infinitely many ansätze can prepare it. The art is choosing one whose parameter landscape is friendly to the optimizer you intend to use.

Structural Role

This node opens Module 2 and sets up everything that follows in it. The next four nodes are direct elaborations of the design axes introduced here:

- 2.2: hardware-efficient ansätze (the hardware-first culture)
- 2.3: chemically-inspired ansätze (the problem-first culture)
- 2.4: compilation, depth, and width — the hardware compatibility axis in detail

- 2.5: Hamiltonian encoding — the *separate* problem of how the Hamiltonian becomes a measurement

The remaining nodes (2.6-2.9) deal with how states are represented, how measurement actually happens on hardware, and the geometric view of the ansatz as a manifold — the deeper structural understanding that pays off when you start reasoning about cost landscapes (Module 3) and barren plateaus (Module 4).

Relations to Other Concepts

- **Variational principle (1.1)** — guarantees that minimizing $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ is a meaningful thing to do regardless of how restricted the ansatz is.
- **Inductive bias in classical ML** — the choice of ansatz is the quantum analogue of the choice of neural network architecture. Both restrict the function class; both inject prior knowledge; both can be over- or under-expressive.
- **Tensor network states** — DMRG, MPS, PEPS — classical analogues of restricted state families with controlled expressivity.
- **Universal quantum computation** — any unitary can in principle be approximated by a sufficiently deep circuit; ansatz design is the art of *not* using a universal one.

Worked Example — Counting the Reachable States

Consider the simplest non-trivial ansatz: a single $R_Y(\theta)$ gate on one qubit, applied to $|0\rangle$.

$$|\psi(\theta)\rangle = R_Y(\theta)|0\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle.$$

The set of reachable states is the **great circle** in the Bloch sphere passing through $|0\rangle$, $|+\rangle$, $|1\rangle$, $|-\rangle$. This is a 1-parameter family — a 1D submanifold of a 2D Hilbert space.

Now add a second layer: $R_Z(\phi)R_Y(\theta)|0\rangle$. The reachable set is now the **entire Bloch sphere** — a 2-parameter family covering all pure single-qubit states. Adding a third parameter $R_Y(\alpha)R_Z(\phi)R_Y(\theta)|0\rangle$ gains nothing in terms of *reachable states* (you can already prepare every single-qubit pure state), but it changes the *parameterization geometry* — the same target state can now be reached via multiple paths, which can either help (more escape routes for an optimizer) or hurt (redundant directions in parameter space).

Lesson: **the number of parameters is not the same as the dimension of the reachable set.** Adding parameters past the point of universality only changes geometry, not coverage — and that geometry change can be either friendly or hostile to optimization.

Visualization

Pending. Three-panel diagram. Panel 1: Bloch sphere with a great circle traced through it (single R_Y ansatz, 1-parameter). Panel 2: full Bloch sphere covered (2-parameter ansatz). Panel 3: same Bloch sphere, but with multiple curves all passing through the same point (3-parameter ansatz, redundant). _Caption: "More parameters can mean more reach, or just more parameterization — these are different._"

Exercises

1. For the 1-parameter ansatz $R_Y(\theta)|0\rangle$, what is the maximum value of $\langle Z\rangle$? What is the minimum? Sketch $\langle Z\rangle(\theta)$ over $\theta \in [-\pi, \pi]$.
2. Show that the ansatz $R_X(\theta_1)R_X(\theta_2)|0\rangle$ is *not* universal for single-qubit states (despite having 2 parameters). What parameter geometry is going wrong?

3. (VQA) Sketch the design tradeoff between expressivity, trainability, and hardware compatibility as a 2D Pareto front for a hypothetical "ideal" VQA ansatz family. Where on the front would you put a hardware-efficient ansatz vs a UCCSD ansatz?

Research Extensions

- **Adaptive ansatz construction (ADAPT-VQE)** — grow the ansatz one operator at a time, choosing the next operator based on its energy gradient. Trades depth for expressivity *only when it helps*.
- **Symmetry-preserving ansätze** — restrict the manifold to states obeying problem symmetries (particle number, total spin, parity), shrinking it to the physically relevant subspace.
- **Learned ansätze** — meta-learn an ansatz structure across families of problems using classical ML pipelines.

Cross-links to Other Courses

- **Machine Learning** — model architecture as inductive bias, capacity vs trainability.
- **Differential Geometry** — submanifolds of complex projective space.
- **Quantum Computing** — universal gate sets, gate decomposition, circuit synthesis.
- **Statistics Module 5** — restricted models and their bias-variance tradeoff.

Variational Quantum Algorithms · Module 2 — Parameterized Circuits · Node 2.1
Concepts: parameterized-circuits, unitary-evolution, hilbert-space
Prerequisites: 1.2
Enables: 2.2, 2.3, 2.4, 2.9

hbar.university — own.your.cognition