Module 8 — Expressivity Vs Trainability

Variational Quantum Algorithms

hbar.university

Expressivity Theory

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.1

This is the **opening node** of Module 8 — the module that develops, in quantitative form, the central tradeoff that has been hovering implicitly behind every module since Module 4: **the more states an ansatz can reach, the harder it is to optimize over**.

Before we can talk about the tradeoff, we need to define **expressivity** with enough precision to be useful. That is the job of this node and node 8.2.

The intuitive picture: an ansatz $U(\boldsymbol{\theta})$ defines a *family* of quantum states $\{|\psi(\boldsymbol{\theta})\rangle\}$ as $\boldsymbol{\theta}$ ranges over its domain. Expressivity is "how big the family is" — formally, how much of the Hilbert space the family covers, in some appropriate measure.

This node introduces the formal definitions, explains the multiple ways expressivity can be measured, and sets up the trainability counterpart in node 8.4.

What "Expressivity" Means

Several closely-related but distinct definitions appear in the literature:

1. Reachability: The set of states $\mathcal{R}(U) = \{|\psi(\boldsymbol{\theta})\rangle : \boldsymbol{\theta} \in \Theta\}$. The "reachable manifold" of the ansatz. Expressivity is, intuitively, the size of $\mathcal{R}(U)$.

2. Volume: For an ansatz with P parameters, the manifold $\mathcal{R}(U)$ has dimension at most P. The "volume" of $\mathcal{R}(U)$ (in the Fubini-Study metric) is a measure of how much of the state space it covers.

3. Distribution distance: Compare the distribution induced on states by random parameter values to the Haar measure on the full state space. An ansatz is "highly expressive" if these distributions are close, "low expressivity" if the induced distribution is far from Haar.

4. Approximation error: For a given target state $|\phi\rangle$, the minimum distance from $|\phi\rangle$ to any state in $\mathcal{R}(U)$. A low approximation error (for any target) means high expressivity.

5. Entanglement coverage: The range of entanglement entropies that states in $\mathcal{R}(U)$ can achieve. Higher coverage = higher expressivity (in the entanglement sense).

These definitions are not equivalent. An ansatz can be highly expressive in one sense and not in another. The thesis (and most recent VQA literature) treats expressivity as a *family* of related but distinct quantities, each useful for different purposes.

Why Expressivity Matters

Expressivity matters for two reasons that pull in opposite directions:

1. You need enough expressivity to represent the answer. If your target state isn't in the reachable manifold of your ansatz, no amount of optimization will find it. The ansatz must be expressive enough that the ground state (or a good approximation to it) lies in $\mathcal{R}(U)$.

2. You need expressivity to be limited or you'll hit a barren plateau. An ansatz that can reach all of Hilbert space is, by Module 4's analysis, exactly the ansatz that has barren plateaus. Too much expressivity is fatal for trainability.

The right amount of expressivity is "just enough." Enough to contain the answer, no more. Designing an ansatz to hit this sweet spot is what Module 2's discussion of problem-inspired ansätze (UCCSD, HVA, ADAPT-VQE) was trying to achieve.

Reachable Manifold Dimension

The simplest expressivity measure is the **dimension** of the reachable manifold.

For a parameterized circuit with P parameters, the manifold $\mathcal{R}(U)$ has dimension at most P. It can be **less** than P if there are gauge redundancies (Section 3.4) or if some parameter combinations are degenerate.

For example, the ansatz $R_X(\theta_1)R_X(\theta_2)|0\rangle$ has 2 parameters but dimension 1 (because only $\theta_1 + \theta_2$ matters).

The full state space of n qubits is a manifold of dimension $2 \cdot 2^n - 2$ (real coordinates on the projective Hilbert space). For an ansatz to be "fully expressive" in the dimension sense, it needs at least this many parameters.

For $n = 4$, that's 30 parameters minimum. For $n = 10$, 2046. For $n = 20$, about 2 million. **Truly universal ansätze are exponentially expensive in parameter count.**

In practice, useful ansätze have parameter counts polynomial in n and reach a polynomially-large submanifold of the exponentially-large state space. This "polynomial expressivity" regime is where almost all VQA work lives.

The Sim-Johnson-Killoran Expressibility

The most widely-used quantitative expressivity measure for VQA was introduced by **Sim, Johnson, Killoran (2019)**. The idea: compute the distribution of fidelities between pairs of random ansatz states, and compare to the Haar distribution.

Let $P_{\text{Haar}}(F)$ be the Haar distribution of $F = |\langle\psi_1|\psi_2\rangle|^2$ for two Haar-random states. This has known closed form: $P_{\text{Haar}}(F) = (d-1)(1-F)^{d-2}$ where $d = 2^n$.

Let $P_{\text{ansatz}}(F)$ be the analogous distribution for two random ansatz states $|\psi(\boldsymbol{\theta}_1)\rangle, |\psi(\boldsymbol{\theta}_2)\rangle$ with $\boldsymbol{\theta}_i$ drawn uniformly.

Expressibility is the KL divergence:

$$\text{Expr} = D_{KL}\big(P_{\text{ansatz}}(F) \parallel P_{\text{Haar}}(F)\big).$$

Lower KL divergence = ansatz distribution is closer to Haar = "more expressive." Higher KL divergence = ansatz is structured = "less expressive."

This is the **standard expressivity metric** in the VQA literature. Most ansatz comparison papers report it.

Practical computation: sample many parameter pairs, compute the empirical fidelity distribution, and bin it. Compute the KL divergence to the analytic Haar distribution.

Expressivity Of Common Ansätze

The Sim-Johnson-Killoran metric has been computed for many standard ansätze. Some representative results (for $n = 4$ qubits):

- **Hardware-Efficient Ansatz, deep:** expressibility close to 0 (Haar-like) — fully expressive.
- **Hardware-Efficient Ansatz, 1 layer:** expressibility ~ 0.1 — limited but nontrivial.
- **UCCSD:** expressibility ~ 0.5 — strongly restricted by particle-number conservation.
- **Tensor Network ansatz (MPS):** expressibility depends on bond dimension; for low bond dimension, very limited.
- **Random Pauli circuit, depth 1:** expressibility ~ 1.0 — almost no coverage.

The trend: **as you add structure (symmetry, sparsity, restricted connectivity), expressibility decreases**. As you remove structure (random gates, deep circuits, generic parameterization), it increases.

This is the operational measurement of the tradeoff Module 4 discussed qualitatively.

Expressivity And Entanglement

Expressivity correlates with — but is not the same as — entanglement coverage.

A high-expressivity ansatz typically reaches highly entangled states (because Haar-random states are highly entangled, by Page's theorem from Section 4.3). A low-expressivity ansatz typically reaches a restricted entanglement range.

But the relationship is not exact:

- A product-state ansatz has zero entanglement and zero expressivity.
- A maximally-entangled (Bell-state-only) ansatz has high entanglement but low expressivity (small reachable set).
- A Haar-random ansatz has high entanglement and high expressivity.

The two axes are related but distinct. Module 4.3 framed barren plateaus in terms of entanglement; this module frames them in terms of expressivity. Both are valid; they capture overlapping but not identical structure.

For practical purposes, **entanglement-based reasoning** (Module 4.3) is more physical and gives you mitigation strategies; **expressivity-based reasoning** (this module) is more mathematical and gives you metrics for ansatz comparison.

Lie Algebraic Expressivity

A more recent and theoretically clean view: expressivity is governed by the **dynamical Lie algebra** of the ansatz generators.

Let $U(\boldsymbol{\theta}) = e^{-i\theta_L H_L} \cdots e^{-i\theta_1 H_1}$ for Hermitian generators $H_1, \ldots, H_L$. The dynamical Lie algebra (DLA) $\mathfrak{g}$ is the smallest Lie algebra containing all the H_j and closed under commutators $[H_i, H_j]$.

Theorem: the reachable manifold $\mathcal{R}(U)$ is contained in the orbit $\{e^{iG}|0\rangle : G \in \mathfrak{g}\}$. The dimension of this orbit is at most the dimension of $\mathfrak{g}$.

Implications: - Small DLA $\rightarrow$ small reachable manifold $\rightarrow$ low expressivity $\rightarrow$ high trainability. - Large DLA $\rightarrow$ large reachable manifold $\rightarrow$ high expressivity $\rightarrow$ low trainability.

For example: a UCCSD ansatz has a DLA equal to the (small) algebra of particle-number-conserving operators. A deep HEA has a DLA equal to the full $\mathfrak{u}(2^n)$, the Lie algebra of all n-qubit unitaries — exponentially large.

Ragone et al. (2024) developed a Lie-algebraic theory of barren plateaus that uses this DLA structure to give precise expressivity bounds. The DLA is the cleanest theoretical proxy for "how expressive is this ansatz" available in 2026.

Structural Role

This node is the **definitions** node of Module 8. It introduces the multiple senses of "expressivity" and previews the trainability counterpart in node 8.4.

The metric definitions in 8.2 and the Lie-algebraic view sketched here are the two technical foundations Module 8 will use throughout.

Relations to Other Concepts

- **Sim-Johnson-Killoran (2019)** — the standard expressibility metric.
- **Ragone et al. (2024)** — the Lie-algebraic theory.
- **Page's theorem** — Haar-random state entanglement.
- **Module 2 (Ansatz design)** — the practical context for the theory.
- **Module 4 (Barren plateaus)** — the trainability obstacle expressivity creates.

Worked Example — Counting Reachable Dimensions

Consider $|\psi(\theta_1, \theta_2)\rangle = R_Y(\theta_1)R_X(\theta_2)|0\rangle$ on 1 qubit.

The state space of 1 qubit is the Bloch sphere S^2, dimension 2. The ansatz has 2 parameters.

Compute reachability: $R_X(\theta_2)|0\rangle = \cos(\theta_2/2)|0\rangle - i\sin(\theta_2/2)|1\rangle$. Then $R_Y(\theta_1)$ rotates this around the Y-axis.

By inspection, the ansatz can reach every state on the Bloch sphere — it is universal for 1 qubit. Reachable manifold has dimension 2.

For 2 qubits, the state space has dimension 6. A 2-qubit ansatz with only single-qubit rotations $R_Y(\theta_1)\otimes R_Y(\theta_2)$ can only reach product states — a 2-dimensional submanifold of the 6-dimensional state space. Adding a CNOT after the rotations expands this to a 4-dimensional submanifold (entangled states reachable from product states by one CNOT). Adding more layers expands further, eventually reaching the full 6-dimensional manifold.

This dimension counting is the simplest expressivity calculation. The Sim-Johnson-Killoran metric, the Lie-algebraic dimension, and other measures all generalize this idea to higher-dimensional state spaces.

Visualization

Pending. A 2D projection of the Bloch sphere with reachable regions of three different ansätze highlighted: (a) a product-only ansatz (small dot), (b) a 1-layer HEA (larger region), (c) a deep HEA (the entire sphere). Caption: "Expressivity is the volume the ansatz reaches."

Exercises

1. For $|\psi(\theta)\rangle = R_Y(\theta)|0\rangle$ (1 qubit, 1 parameter), what is the reachable manifold? What is its dimension?
2. For an ansatz with 5 parameters in 4 qubits (state space dimension 30), what is the maximum possible reachable manifold dimension? Is the ansatz universal?
3. (VQA) The Sim-Johnson-Killoran expressibility for a deep HEA approaches the Haar value. Show analytically that for a 1-qubit ansatz $R_Y(\theta_1)R_X(\theta_2)R_Y(\theta_3)$, the expressibility is exactly Haar. (Hint: this is universal for 1 qubit.)

Research Extensions

- **Tighter expressivity-trainability bounds** — pinning down the exact constants in the tradeoff.
- **Problem-aware expressivity** — measures that account for the specific Hamiltonian, not just the ansatz.
- **Computational expressivity** — relating expressivity to computational hardness of the optimization.
- **Expressivity under noise** — how hardware noise modifies the effective expressivity of an ansatz.

Cross-links to Other Courses

- **Lie Theory** — Lie algebras, Lie groups, dynamical Lie algebras of control systems.
- **Differential Geometry** — manifolds, dimension, embedding.
- **Statistical Learning Theory** — VC dimension, capacity, generalization (the classical analogue of expressivity).
- **Module 2 (Parameterized Circuits)** — the operational ansatz design.
- **Module 4 (Barren Plateaus)** — the trainability obstacle.

Expressibility Metrics

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.2

Node 8.1 introduced expressivity as a *family* of related but inequivalent quantities. This node makes those quantities concrete: it walks through the **specific metrics** in the literature, how each one is computed, what each one tells you, and when each is the right tool.

The pragmatic motivation: if you are designing or comparing ansätze, you need a number — preferably one you can actually compute on a laptop. The pure theoretical view (reachable manifold dimension, Lie algebra dimension) doesn't scale. The metrics in this node **do** scale, at the cost of capturing only a slice of the full picture.

What A Good Expressibility Metric Should Do

Before walking through the metrics, the desiderata:

1. Computability. You should be able to compute it from samples (parameter draws + cost evaluations or fidelity calculations). No exact integrals, no exponential scaling.

2. Comparability. Two ansätze with the same number of parameters and qubits should give comparable numbers. The metric shouldn't be dominated by ansatz size.

3. Sensitivity. It should distinguish ansätze that differ in expressivity, not just in trivial ways.

4. Interpretability. A user should be able to read the number and know what it means qualitatively ("more expressive" or "less expressive").

5. Connection to trainability. Ideally, the metric correlates with trainability outcomes — knowing the number should help you predict whether the ansatz will train well.

No single metric meets all five. Each one trades off some desiderata for others. The literature has converged on a small handful that are "good enough" for most purposes.

Metric 1: Sim-Johnson-Killoran Expressibility

Already introduced in 8.1 — included here for completeness with the computational details.

Definition. Let $P_{\text{ansatz}}(F)$ be the distribution of fidelities $F = |\langle\psi(\boldsymbol{\theta}_1)|\psi(\boldsymbol{\theta}_2)\rangle|^2$ over uniformly sampled parameter pairs. Let $P_{\text{Haar}}(F) = (d-1)(1-F)^{d-2}$ for $d = 2^n$. Then

$$\text{Expr}_{\text{SJK}} = D_{KL}(P_{\text{ansatz}} \,\|\, P_{\text{Haar}}).$$

Computational protocol: 1. Sample M parameter pairs $(\boldsymbol{\theta}_1^{(m)}, \boldsymbol{\theta}_2^{(m)})$. 2. For each, compute $F^{(m)} = |\langle\psi(\boldsymbol{\theta}_1^{(m)})|\psi(\boldsymbol{\theta}_2^{(m)})\rangle|^2$. 3. Bin the fidelities into B bins to get an empirical distribution $\hat{P}_{\text{ansatz}}$. 4. Compute the KL divergence to the analytic Haar distribution.

Cost: M state preparation pairs, each requiring fidelity computation (which is $O(\text{poly}(n))$ on a simulator and harder on hardware — usually estimated via SWAP test or inversion test).

Interpretation: - $\text{Expr}_{\text{SJK}} \to 0$: ansatz fidelity distribution matches Haar — fully expressive (Haar-like coverage). - Expr_{SJK} large: ansatz is structured, far from Haar.

Strengths: computable, well-studied, comparable across ansätze with the same n. **Weaknesses:** dominated by global structure; misses fine-grained expressivity differences. The KL is sensitive to bin width and to the number of samples.

Metric 2: Entangling Capability

Also from Sim-Johnson-Killoran (2019). Measures the **average entanglement** of states produced by the ansatz.

Definition. Let $Q(|\psi\rangle)$ be the **Meyer-Wallach Q-measure**:

$$Q(|\psi\rangle) = 2\left(1 - \frac{1}{n}\sum_{k=1}^{n} \mathrm{Tr}(\rho_k^2)\right),$$

where ρ_k is the reduced density matrix on qubit k. $Q = 0$ for product states, $Q = 1$ for maximally entangled (in a specific sense). Then

$$\mathrm{Ent}(U) = \mathbb{E}_{\boldsymbol{\theta}}[Q(|\psi(\boldsymbol{\theta})\rangle)].$$

Computational protocol: sample parameters, prepare states, compute Q for each, average.

Cost: M state preparations and $O(n)$ reduced density matrices each.

Interpretation: - Ent ≈ 0: the ansatz produces near-product states. Limited expressivity. - Ent ≈ 1: the ansatz produces highly entangled states. - Haar value: roughly $1 - 1/(2^n + 1) \approx 1$ for large n.

Strengths: physically meaningful, easy to compute, complementary to SJK expressibility. **Weaknesses:** entanglement is a *consequence* of expressivity, not the same thing — high entanglement does not imply high reachability.

A good practice is to report **both** SJK expressibility and entangling capability, since they measure different facets.

Metric 3: Dynamical Lie Algebra Dimension

Already sketched in 8.1. A theoretically clean but computationally intensive metric.

Definition. Given the ansatz $U(\boldsymbol{\theta}) = \prod_j e^{-i\theta_j H_j}$, construct the dynamical Lie algebra $\mathfrak{g}$ by closing $\{H_j\}$ under commutators. Then

$$\mathrm{Expr}_{\mathrm{DLA}}(U) = \dim \mathfrak{g}.$$

Computational protocol: symbolic Lie algebra closure. Start with the generators, compute pairwise commutators, add new linearly independent operators, repeat until closure.

Cost: $O(\dim \mathfrak{g}^2)$ commutator computations and linear-independence checks. For an ansatz where $\dim \mathfrak{g}$ is polynomial in n, this is feasible. For deep ansätze where $\dim \mathfrak{g}$ is exponential, intractable.

Interpretation: - Small DLA dimension → small reachable manifold → low expressivity. - DLA dimension polynomial in n → "polynomially expressive" — typical regime for problem-inspired ansätze. - DLA dimension exponential in n → "fully expressive" — typical for HEA at sufficient depth.

Strengths: rigorous, ansatz-intrinsic (doesn't depend on parameter sampling), connects directly to trainability via the Ragone et al. theory. **Weaknesses:** scales poorly with depth; doesn't capture continuous expressivity differences (it's a single integer).

Use DLA dimension for theoretical analysis; use SJK expressibility for empirical comparison.

Metric 4: Effective Dimension

Introduced by Abbas et al. (2021) — "The power of quantum neural networks."

The idea: measure the **information geometric dimension** of the parameter space, weighted by the local geometry of the cost landscape.

Definition. Let $I(\boldsymbol{\theta})$ be the Fisher information matrix at $\boldsymbol{\theta}$. The effective dimension at scale γ is

$$d_{\text{eff}}(\gamma) = \frac{2\log\left(\frac{1}{V_\Theta}\int_\Theta \sqrt{\det(I_d + \gamma I(\boldsymbol{\theta}))}\, d\boldsymbol{\theta}\right)}{\log(\gamma/(2\pi\log n))},$$

where I_d is the identity, V_Θ is the parameter-space volume, and γ is a scale parameter. (Don't be intimidated — this is essentially "average rank of the Fisher information.")

Interpretation: - d_{eff} close to P (number of parameters): all parameters are independently informative — fully effective ansatz. - $d_{\text{eff}} \ll P$: many parameters are redundant — the effective parameter space is much smaller than P.

Strengths: captures redundancy, connects to generalization theory. **Weaknesses:** computing the Fisher information is expensive; the metric is defined at a scale γ and depends on it.

Effective dimension is the right metric when you suspect your ansatz has many redundant parameters (gauge degrees of freedom, near-degenerate directions). For most problem-inspired ansätze, it's overkill.

Metric 5: Frame Potential

A more theoretical metric: the **frame potential**.

Definition. For an ansatz $U(\boldsymbol{\theta})$ inducing a distribution on unitaries, the t-th frame potential is

$$\mathcal{F}_t(U) = \mathbb{E}_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2}\left[|\mathrm{Tr}(U(\boldsymbol{\theta}_1)U(\boldsymbol{\theta}_2)^\dagger)|^{2t}\right].$$

For Haar-random unitaries, the frame potential takes a specific value depending on t and $d = 2^n$. An ansatz forms a **t-design** if its frame potential matches the Haar value:

$$\mathcal{F}_t(\text{ansatz}) = \mathcal{F}_t(\text{Haar}).$$

Interpretation: - The ansatz is a 1-design if its first moment matches Haar. - A 2-design if both first and second moments match. - More expressivity = higher t-design.

Strengths: rigorous, connects to the proof of barren plateaus (which assumes 2-designs). **Weaknesses:** needs to be computed at each order t; only practical for low t.

The frame potential is the *right* metric for connecting to barren plateau theory (Module 4 used it implicitly via "approximate 2-design"). For practical comparison of two ansätze, SJK or entangling capability is easier.

Metric 6: Approximation Error On A Test Suite

A purely operational metric. Define a set of "test states" $\{|\phi_k\rangle\}_{k=1}^{K}$. For each, compute the minimum error over the ansatz:

$$\epsilon_k = \min_{\boldsymbol{\theta}} \| |\phi_k\rangle - |\psi(\boldsymbol{\theta})\rangle \|.$$

Average over the test suite:

$$\text{Expr}_{\text{approx}} = \frac{1}{K} \sum_{k=1}^{K} \epsilon_k.$$

Interpretation: - Low average error → ansatz can approximate diverse target states → high expressivity. - High average error → ansatz misses many targets → low expressivity.

Strengths: directly operational — measures what you actually care about (can the ansatz represent my target?). **Weaknesses:** depends on the test suite. A test suite biased toward your ansatz's structure will give an unfair advantage.

Use this when you have a specific application in mind and a representative set of target states for that application.

Comparing The Metrics

A summary table:

Metric	Cost	Output range	Captures	Best for
SJK Expressibility	Moderate (samples)	KL divergence (positive)	Distributional coverage	General ansatz comparison
Entangling Capability	Cheap	$[0, 1]$	Entanglement structure	Quick check
DLA Dimension	Expensive (symbolic)	Integer	Reachable manifold dim	Theoretical analysis
Effective Dimension	Expensive	$[0, P]$	Parameter redundancy	Over-parameterized ansätze
Frame Potential	Moderate	Real number	t-design property	Plateau theory
Approximation Error	Application-dep.	Real number	Target representability	Application-driven

For most VQA work, the practical recommendation is: report **SJK expressibility + entangling capability** as the standard pair. Add **DLA dimension** if you can compute it. Add **effective dimension** if your ansatz has visible redundancy.

A Worked Comparison

Consider two 4-qubit ansätze:

Ansatz A: Hardware-Efficient, 2 layers. Single-qubit R_X, R_Y, R_Z on each qubit, then nearest-neighbor CNOTs. Repeat. Total: 24 parameters.

Ansatz B: UCCSD with 4 electrons. Trotterized particle-number-conserving excitations. Total: ~12 parameters.

SJK Expressibility: - A: ≈ 0.05 (close to Haar, highly expressive). - B: ≈ 0.5 (far from Haar — strongly restricted by particle-number conservation).

Entangling Capability: - A: ≈ 0.85 (nearly Haar entanglement). - B: ≈ 0.4 (limited entanglement — restricted to particle-number subspace).

DLA Dimension: - A: full $\mathfrak{u}(2^4) = 256$ — exponential in n. - B: $\binom{8}{4}^2 = 4900$ but restricted to particle-number sector, so $\binom{8}{4} = 70$.

Effective Dimension at $\gamma = 1$: - A: ≈ 18 (out of 24, some redundancy from gauge freedoms). - B: ≈ 11 (out of 12, near-fully effective).

Approximation Error on the H_2 ground state: - A: ≈ 0.001 (excellent — can represent the ground state). - B: ≈ 0.0001 (even better — UCCSD is *designed* for chemistry).

Conclusion: Ansatz A is more expressive in every general sense, but Ansatz B is *more expressive on the actual problem* because its restricted form happens to align with the chemistry symmetries. Both can solve H_2; they just use very different mechanisms.

Caveats On Metric Interpretation

1. Metrics measure structure, not performance. A more expressive ansatz isn't always better. As Module 4 showed, expressivity correlates with trainability obstacles. The right amount is "just enough."

2. Metric values depend on the parameter range. SJK expressibility, entangling capability, and effective dimension all depend on the parameter sampling distribution. Default is uniform on $[0, 2\pi]^P$, but if you constrain parameters (e.g., $[-\pi/4, \pi/4]$), the metrics change.

3. Sample noise. Empirical metrics (SJK, entangling capability) have sampling noise that scales as $1/\sqrt{M}$. For meaningful comparison, you need M large enough to resolve the difference between ansätze.

4. Comparability across qubit count. SJK expressibility and DLA dimension scale differently with n, so comparing ansätze on different qubit counts requires care. Within a fixed n, comparison is meaningful.

5. No metric captures cost-landscape geometry. None of the standard metrics tells you about local minima, barren plateaus, or convergence rates. They capture *static* expressivity, not *dynamic* trainability.

For trainability-aware ansatz comparison, you need the metrics from node 8.4 in addition.

Structural Role

This node is the **operational toolkit** of Module 8. It enumerates the actual numbers you compute and report when designing or analyzing an ansatz, with practical guidance on which metrics to use for which purpose.

It is the bridge between the abstract definitions of 8.1 and the trainability bounds of 8.4.

Relations to Other Concepts

- **Sim-Johnson-Killoran (2019)** — the original SJK metric paper.
- **Abbas et al. (2021)** — effective dimension.
- **Meyer-Wallach Q-measure (2002)** — the entanglement metric.

- **Frame potentials (Scott 2008)** — t-design theory.
- **Ragone et al. (2024)** — DLA expressivity.

Worked Example — Computing SJK Expressibility For A Toy Ansatz

Consider the 2-qubit ansatz $U(\theta_1, \theta_2) = R_Y(\theta_1) \otimes R_Y(\theta_2)$ — pure product, no entanglement.

Step 1: Sample parameter pairs. Draw $M = 1000$ pairs of (θ_1, θ_2) and (θ_1', θ_2') uniformly from $[0, 2\pi]^2$.

Step 2: Compute fidelities. $|\langle\psi(\theta_1, \theta_2)|\psi(\theta_1', \theta_2')\rangle|^2 = |\langle 0|R_Y^\dagger(\theta_1)R_Y(\theta_1')|0\rangle|^2 \cdot |\langle 0|R_Y^\dagger(\theta_2)R_Y(\theta_2')|0\rangle|^2$.

Each factor is $\cos^2((\theta_1' - \theta_1)/2)$, uniformly distributed in $[0, 1]$. The product is the product of two independent uniforms — distribution is $-\log(F)$ (concentrated near $F = 0$, heavy near 1).

Step 3: Compare to Haar. For $d = 4$, $P_{\text{Haar}}(F) = 3(1 - F)^2$. The ansatz distribution is dominated by small fidelities (most product states are nearly orthogonal), but the *concentration profile* is very different from Haar.

Step 4: Compute KL. The KL divergence is large (~ 1.5 in natural log units), reflecting that this ansatz is *far* from Haar.

For comparison, a 2-qubit ansatz with one CNOT after the rotations has SJK expressibility ≈ 0.3 — much closer to Haar but still structured.

A deep HEA on 2 qubits has SJK expressibility ≈ 0.01 — essentially Haar.

This worked example shows the kind of empirical comparison that drives ansatz design in practice.

Visualization

Pending. A bar chart of SJK expressibility for 6 standard ansätze (HEA-1, HEA-3, HEA-deep, UCCSD, MPS-2, MPS-8) on 4 qubits, with the Haar value marked as a horizontal line. Caption: "Lower bars are closer to Haar — more expressive."

Exercises

1. Compute the entangling capability of $U(\theta) = R_Y(\theta) \otimes R_Y(\theta)$ (2 qubits, 1 parameter, no entangling gate). What is it?
2. The SJK expressibility for an ansatz that produces only the state $|+\rangle^{\otimes n}$ regardless of parameters is what? (Hint: the fidelity distribution is a delta at 1.)
3. (VQA) For a 6-qubit HEA with 5 layers and a UCCSD ansatz with 6 electrons, predict the relative ordering of (a) SJK expressibility, (b) entangling capability, (c) DLA dimension. State any assumptions.

Research Extensions

- **Trainability-aware expressibility metrics** — metrics that incorporate gradient variance, not just static structure.
- **Application-conditioned expressibility** — metrics defined relative to a specific Hamiltonian or target distribution.
- **Hierarchical metrics** — multi-resolution expressibility (local vs global).
- **Sample-efficient expressibility estimation** — getting useful numbers from few samples.

Cross-links to Other Courses

- **Information Geometry** — Fisher information, effective dimension.
- **Random Matrix Theory** — Haar measure, frame potentials.
- **Module 4 (Barren Plateaus)** — the trainability cost of high expressibility.
- **Module 2 (Ansatz Design)** — the practical context.

Universal Approximation in Quantum Circuits

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.3

The classical neural network literature has a famous theorem: **a feedforward network with one hidden layer of sufficient width can approximate any continuous function** (Cybenko 1989, Hornik 1991). This is the *universal approximation theorem* and it is what justifies neural networks at the foundational level.

Quantum variational circuits have analogous universality results, and this node walks through them. The key questions:

1. **Is there a quantum analogue of universal approximation?** (Yes, several.)
2. **What does "universal" mean in the quantum setting?** (More than one thing.)
3. **What does universality cost in terms of circuit depth, parameter count, etc.?** (Exponentially much, in general.)
4. **How does universality relate to barren plateaus?** (Negatively — universal ansätze plateau.)

The punchline: quantum universal approximation theorems exist, but they are *existence statements* about ansätze that are exponentially deep or exponentially wide. The practically useful ansätze are not universal — they are restricted, problem-specific, and that restriction is what makes them trainable.

What "Universal" Could Mean For A Quantum Circuit

There are multiple senses in which a parameterized circuit could be "universal."

1. State universality. The ansatz can prepare any state in the Hilbert space:

$$\forall |\phi\rangle \in \mathcal{H},\ \exists \boldsymbol{\theta} : |\psi(\boldsymbol{\theta})\rangle = |\phi\rangle.$$

2. Unitary universality. The ansatz can implement any unitary on the Hilbert space:

$$\forall U \in U(2^n),\ \exists \boldsymbol{\theta} : V(\boldsymbol{\theta}) = U.$$

State universality is weaker (you only need to reach states starting from $|0\rangle^{\otimes n}$). Unitary universality is stronger (you need to be able to act on arbitrary inputs).

3. Function approximation universality. The map $\boldsymbol{\theta} \to \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ (or similar observable expectations) can approximate any continuous function on the parameter space. This is the closest analogue of classical universal approximation.

4. Distribution universality. The ansatz can sample from any distribution over computational basis states. Relevant for quantum generative models.

5. Operator universality. The ansatz can prepare any Hermitian observable as a linear combination of its parameters. Relevant for quantum machine learning kernels.

These are *not* equivalent. A circuit can be universal in one sense and not in another. The literature usually means **state universality** unless otherwise specified.

State Universality: Existence

Theorem (state universality). A parameterized circuit on n qubits with at least $2 \cdot 2^n - 2$ real parameters can be state-universal — that is, there exists a circuit construction with this many parameters that can reach every state in the projective Hilbert space.

Proof idea. The projective Hilbert space $\mathbb{CP}^{2^n-1}$ has real dimension $2 \cdot 2^n - 2$. A smooth, surjective map from a parameter space to this manifold must have parameter space dimension at least $2 \cdot 2^n - 2$. With this many parameters, an explicit construction (e.g., the **Schmidt decomposition based circuit** of Iten et al. 2016) achieves universality.

Cost: $2 \cdot 2^n - 2$ parameters means **exponential growth in qubit count**. For $n = 10$, 2046 parameters; for $n = 20$, 2 million; for $n = 50$, 10^{15}.

State universality is theoretically important but **practically infeasible** for any $n > 10$ or so. Universal ansätze are not used in practice; they are reference points to anchor the theory.

Unitary Universality: Existence

Theorem (unitary universality). A parameterized circuit on n qubits can implement any unitary in $U(2^n)$ if and only if its parameter count is at least $(2^n)^2 - 1 = 4^n - 1$.

Cost: $4^n - 1$ parameters — even worse than state universality. Doubly exponential in qubit count.

Unitary universality is **even more theoretical** than state universality. The only reason it's worth knowing about is that it's the natural notion when the ansatz needs to act on arbitrary inputs (e.g., quantum kernels, quantum data processing).

Practical Universality: What "Enough" Looks Like

The interesting question is not "what is the minimum parameter count for universality" but **"how many parameters give an ansatz expressive enough to solve the problem at hand?"**

For chemistry (VQE on small molecules): - Hartree-Fock state plus a UCCSD ansatz with $O(N^4)$ parameters (for N electrons) is enough to reach chemical accuracy on most small molecules. This is *much* less than full universality.

For optimization (QAOA): - A QAOA ansatz with $O(p)$ parameters at depth p is enough to solve many MaxCut and constraint-satisfaction problems with provable approximation guarantees. The required p is small (often $p = 1$ or 2) for many problems, vastly less than universality.

For machine learning (data-encoded models): - A re-uploading ansatz with $O(L)$ data-encoding layers can express functions of growing complexity. Schuld-Sweke-Meyer (2021) showed that L layers gives Fourier series of degree up to L.

The pattern: **practical ansätze trade universality for trainability**. The right amount of expressivity is just enough to contain the answer, and not more.

The Schuld-Sweke-Meyer Theorem

A particularly clean universality result for quantum machine learning models.

Setup. A quantum model is a function $f(x; \boldsymbol{\theta}) = \langle 0|U^\dagger(x, \boldsymbol{\theta}) M U(x, \boldsymbol{\theta})|0\rangle$, where x is a data input encoded into the circuit, $\boldsymbol{\theta}$ are trainable parameters, and M is an observable.

Theorem (Schuld-Sweke-Meyer 2021). If the data x is encoded via Pauli rotations e^{-ixH}, and the encoding is repeated L times with trainable circuits in between, then $f(x)$ is a **truncated Fourier series** in x of degree $\leq L$. The Fourier coefficients are determined by the trainable circuits; in particular, all Fourier modes up to degree L can be reached by suitable choice of $\boldsymbol{\theta}$.

Implication: with enough re-uploading layers, the model can approximate any continuous function of x on a compact domain (Stone-Weierstrass for trigonometric polynomials). This is a quantum universal approximation theorem in the function-approximation sense.

The catch: the Fourier coefficients are constrained by the structure of the circuit. The model is universal in the sense that *some* coefficients can be chosen freely, but not all combinations are achievable. There are dependencies that the proof doesn't explicitly characterize.

This result is the cleanest connection between quantum circuits and classical function approximation theory, and it's the theoretical foundation for **data re-uploading** as a quantum machine learning architecture.

Universality And Barren Plateaus

The deep tension: **the most universal ansätze are exactly the ansätze most prone to barren plateaus**.

Why: universality (or near-universality) means the ansatz reaches a Haar-like distribution over states. Module 4 showed that Haar-2-design ansätze have exponentially small gradient variance — barren plateau.

Quantitative statement: for an ansatz that is an ϵ-approximate state-2-design,

$$\mathrm{Var}[\partial_j C] \leq \frac{1}{2^n - 1}(\|H\|_2^2 + \epsilon \cdot \mathrm{poly}(n)).$$

As $\epsilon \to 0$ (true 2-design), the bound goes to $1/(2^n - 1)$ — exponentially small.

Conclusion: a fully universal ansatz cannot be trained by gradient descent on a generic Hamiltonian. This is a *theorem*, not a heuristic. The only escape is to use a non-universal (structured, restricted) ansatz.

This is the central theoretical message of Module 8: **universality and trainability are in fundamental opposition**, and any practical VQA strategy must explicitly choose where to land on the spectrum.

The Practical Take

Universal approximation theorems for quantum circuits exist, but they:

1. **Require exponentially many parameters** (state univ: $2 \cdot 2^n - 2$, unitary univ: $4^n - 1$).
2. **Imply barren plateaus** (universal = Haar-like = no gradients).
3. **Are not how successful VQAs actually work** (chemistry uses problem-inspired, optimization uses problem-restricted).

The honest summary: universal approximation is a theoretical curiosity for quantum circuits, not a practical guide. **The practically useful theorems are about restricted ansätze**: how much function approximation can a *specific structured* ansatz do, given its symmetries and depth?

This is the territory of nodes 8.4 (trainability bounds for restricted ansätze) and 8.5 (the explicit tradeoff curve).

Universal Approximation In Classical ML, For Comparison

The classical analogue is much friendlier:

- A neural net with one hidden layer of width $O(\mathrm{poly})$ is a universal approximator (Cybenko, Hornik). The width grows polynomially in the function complexity, not exponentially.
- Backpropagation works regardless of width — no analogue of barren plateaus that scales with width.

- In practice, classical NNs are *vastly* over-parameterized, and that over-parameterization is *helpful*, not harmful (Belkin-Hsu-Ma-Mandal "double descent").

In quantum circuits, **the relationship is inverted**: over-parameterization is harmful, and the right amount of structure is essential. This is one of the fundamental qualitative differences between quantum and classical machine learning, and it shapes everything about how VQAs are designed.

Structural Role

This node is the **theoretical reference point** of Module 8. It establishes what universal approximation means in the quantum setting, explains why universal ansätze are not what we want, and motivates the move to restricted ansätze that is the focus of 8.4 and 8.5.

It also provides the connection to classical machine learning theory (and the cautionary tale about how the classical results do *not* translate directly).

Relations to Other Concepts

- **Cybenko 1989, Hornik 1991** — classical universal approximation.
- **Iten et al. 2016** — explicit universal state preparation circuits.
- **Schuld-Sweke-Meyer 2021** — Fourier series structure of data re-uploading.
- **McClean et al. 2018** — barren plateau theorem (the cost of universality).
- **Module 4 (Barren plateaus)** — the trainability cost.

Worked Example — Counting Parameters For A 4-Qubit Universal Ansatz

For 4 qubits, the projective Hilbert space has real dimension $2 \cdot 2^4 - 2 = 30$.

State-universal ansatz: at least 30 parameters. **Unitary-universal ansatz:** at least $4^4 - 1 = 255$ parameters.

A Hardware-Efficient Ansatz with single-qubit R_X, R_Y, R_Z on each qubit (12 parameters per layer) plus 4 nearest-neighbor CNOTs would need at least $\lceil 30/12 \rceil = 3$ layers for state universality (if the CNOTs effectively connect the parameters). In practice, the SJK expressibility approaches Haar at around 3-5 layers, consistent with this dimension counting.

For unitary universality, you'd need much more: at least 21 layers. In practice, even 21 layers doesn't quite achieve unitary universality for HEA because the circuit's connectivity restricts the reachable unitaries.

The dimension counting is a *lower bound* — actual universality requires the right topology of parameter dependence, which not every counting-correct circuit provides.

Visualization

Pending. A chart showing parameter count vs qubit count for state universality, unitary universality, and "practical" UCCSD/HEA. State universality grows as 2^{n+1}, unitary as 4^n, UCCSD as $O(n^4)$, HEA polynomial in n per layer. Caption: "Universality is exponential. Practical ansätze are polynomial."

Exercises

1. For a 1-qubit ansatz $R_Y(\theta_1)R_X(\theta_2)R_Y(\theta_3)$, is it state-universal? Unitary-universal? Justify.
2. The Schuld-Sweke-Meyer theorem says L data-encoding layers give Fourier degree L. For a function $f(x) = \cos(3x) + 0.5\cos(5x)$, what is the minimum L you need to express it exactly?
3. (VQA) For the LiH molecule (10 qubits in a minimal basis), estimate the parameter count of (a) a state-universal ansatz, (b) a UCCSD ansatz, (c) a Hardware-Efficient Ansatz with 5 layers. Comment on the gap.

Research Extensions

- **Quantitative universality vs trainability tradeoffs** — explicit curves for specific ansatz families.
- **Constrained universality** — universality on a symmetry-restricted subspace (e.g., particle-number sectors).
- **Practical universality with depth annealing** — starting with restricted ansätze and growing toward universality.
- **Universal approximation under noise** — how hardware errors affect the universality results.

Cross-links to Other Courses

- **Approximation Theory** — universal approximation theorems, density results.
- **Functional Analysis** — Stone-Weierstrass theorem, denseness in continuous function spaces.
- **Module 4 (Barren Plateaus)** — the trainability cost of universality.
- **Module 2 (Ansatz Design)** — the practical alternative to universality.

Trainability Bounds

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.4

This is the **trainability counterpart** of nodes 8.1-8.3. Where those nodes asked "how big is the family of states an ansatz can reach?", this one asks "how hard is it to optimize over that family?"

The two questions are not independent. Module 4 already showed that a fully expressive (Haar 2-design) ansatz has barren plateaus. This node makes the relationship **quantitative**: it states the trainability bounds in their precise forms, walks through the most important variants, and connects each one to the corresponding expressivity property.

The result of this node is the inputs to node 8.5, which combines them into the **expressivity-trainability tradeoff curve** that is the central object of Module 8.

What "Trainability" Means

Several closely-related but distinct quantities can serve as "trainability":

1. Gradient variance. $\mathrm{Var}_{\boldsymbol{\theta}}[\partial_j C(\boldsymbol{\theta})]$ for randomly initialized parameters. Small variance = barren plateau = low trainability.

2. Gradient signal-to-noise ratio. $\mathbb{E}[\partial_j C]^2/\mathrm{Var}[\partial_j C]$ — how big is the expected gradient compared to its fluctuation? Low SNR = the gradient is dominated by noise.

3. Expected number of iterations to convergence. Operationally — how many optimizer steps does it take to reach a target accuracy? More iterations = lower trainability.

4. Probability of successful initialization. Out of a random sample of initial parameters, what fraction lead to successful training? Lower fraction = lower trainability.

5. Worst-case landscape geometry. Are there many local minima? Saddle points? Flat regions? More such structures = lower trainability.

The first one — **gradient variance** — is the standard quantitative measure in the VQA literature, and the one most barren plateau theorems are stated for. The others are operationally important but less amenable to formal analysis.

The Master Trainability Inequality

The most general statement, due to Cerezo et al. (2021) and refined by McClean et al. and Holmes et al.:

Theorem (general gradient variance bound). For an ansatz that forms an ϵ-approximate state-2-design, and a global cost $C = \langle\psi|H|\psi\rangle$:

$$\mathrm{Var}_{\boldsymbol{\theta}}[\partial_j C] \leq \frac{2\|H\|_2^2}{2^n - 1} + O(\epsilon).$$

For a true 2-design ($\epsilon = 0$), the bound is exactly $2\|H\|_2^2/(2^n - 1)$ — exponentially small in n.

This is the master inequality. All other trainability bounds in the VQA literature can be viewed as refinements that:

1. **Tighten the bound** for non-2-design ansätze.
2. **Loosen the assumption** to cover specific structured cost functions or ansatz families.

3. **Replace** the 2-design assumption with a Lie-algebraic condition (Ragone et al. 2024).
4. **Add corrections** for noise (Wang et al. 2021).

Bound 1: Original McClean Bound (Global Cost, Deep HEA)

The original barren plateau theorem of McClean et al. 2018:

$$\mathrm{Var}[\partial_j C_{\mathrm{global}}] \in O(2^{-n}).$$

Assumptions: the ansatz at depth $O(n)$ forms an approximate 2-design, the cost is global (sum of single-qubit operators on all qubits), parameters are uniformly initialized.

Implication: the gradient is exponentially small. To resolve it above shot noise requires $O(4^n)$ shots per evaluation. Infeasible.

This is the canonical statement and the one most often cited. Module 4 covered it in detail.

Bound 2: Cerezo Local-Cost Refinement

For a *local* cost function (sum of single-qubit observables), the bound improves dramatically.

Theorem (Cerezo et al. 2021). For $C_{\mathrm{local}} = \sum_i \langle\psi|O_i|\psi\rangle$ where each O_i acts on $O(\mathrm{poly}\log n)$ qubits:

$$\mathrm{Var}[\partial_j C_{\mathrm{local}}] \in \Omega(\mathrm{poly}(1/n)),$$

provided the ansatz is "shallow" (depth $O(\mathrm{poly}\log n)$) and the cost is built from local observables.

Implication: local costs evade the barren plateau in shallow ansätze. The polynomial lower bound on gradient variance is enough for gradient descent to make progress.

Caveat: as the ansatz becomes deeper than $O(\log n)$, even the local-cost bound deteriorates. The refinement only works in the *shallow* regime.

This is the most important practical refinement. It justifies using **depth-restricted, local-cost** VQAs as the default, with global-cost variants only when necessary.

Bound 3: Marrero Entanglement Bound

A more physical view: trainability is related to the **average entanglement** of the ansatz states.

Theorem (Marrero-Kieferová-Wiebe 2021). Let $S(\rho_A)$ be the average entanglement entropy of the reduced state on subsystem A (averaged over parameter initializations). Then

$$\mathrm{Var}[\partial_j C] \leq f(\dim A) \cdot 2^{-S(\rho_A)},$$

where f is a function of subsystem size only.

Implication: high entanglement $\to$ small gradient variance $\to$ barren plateau. The "volume law" entanglement of Haar-random states is exactly the case where S scales linearly in n, giving $2^{-S} = 2^{-O(n)}$ — barren plateau.

This bound provides the *physical mechanism* for the plateau (entanglement spreading) and a recipe for avoiding it: keep the entanglement low (area-law or log-law).

Bound 4: Ragone Lie-Algebraic Bound

The most recent and theoretically clean variant. Replaces the 2-design assumption with a Lie-algebraic condition.

Theorem (Ragone et al. 2024). Let $\mathfrak{g}$ be the dynamical Lie algebra of the ansatz (Section 8.1). Then

$$\mathrm{Var}[\partial_j C] \leq \frac{P_{\mathrm{purity}}(\mathfrak{g}, H)}{\dim \mathfrak{g}},$$

where P_{purity} is a "purity" of the cost Hamiltonian relative to the algebra (a combinatorial quantity that can be computed from $\mathfrak{g}$ and H).

Implication: trainability is governed by the dimension of the dynamical Lie algebra. Small DLA $\rightarrow$ large variance $\rightarrow$ trainable. Large DLA $\rightarrow$ small variance $\rightarrow$ barren plateau.

This is the **cleanest form** of the bound because:

1. It does not assume the ansatz forms a 2-design (which is hard to verify).
2. It works for *any* polynomial-dimension DLA, not just specific families.
3. It connects directly to the expressivity theory (8.1) — the same DLA dimension that controls expressivity also controls trainability.

This unifies expressivity and trainability under a single Lie-algebraic framework, and is the basis for the explicit tradeoff curve in 8.5.

Bound 5: Wang Noise-Induced Plateau

A very different bound, capturing the noise-induced plateau (Module 4 node 4.5).

Theorem (Wang et al. 2021). For an ansatz of depth L under depolarizing noise with rate p per gate:

$$\mathrm{Var}[\partial_j C] \leq C_0 \cdot (1-p)^{2L},$$

for a constant C_0.

Implication: the gradient variance decays *exponentially in depth times noise rate*. Even on a non-2-design ansatz, sufficient depth and noise will create a barren plateau.

Distinction from the unitary plateau: this bound has nothing to do with expressivity or 2-designs. It is a *purely noise-driven* effect. Increasing the ansatz expressivity does not change this bound; only reducing depth or noise does.

This bound matters because it sets the **noise-corrected ceiling** on practical ansatz depth: even if you have a structured (low-DLA) ansatz, deep noise will plateau it.

Bound 6: Local Approximate 2-Designs

A partial-design refinement: ansätze that are approximate 2-designs only on a *subspace*.

Theorem. If the ansatz is a 2-design on a k-dimensional subspace of the full 2^n-dimensional Hilbert space, then for cost functions whose support is contained in that subspace:

$$\mathrm{Var}[\partial_j C] \in \Theta(1/k).$$

Implication: if you can restrict the ansatz to a polynomially-large subspace (e.g., particle-number conservation gives $k = \binom{2n}{n}$), super-polynomial but much less than 2^{2n}), the trainability is only polynomially small, not exponentially.

This is the formal statement of "symmetry-restriction helps." Module 2's UCCSD ansatz uses particle-number conservation; Module 4 explained the heuristic; this bound provides the rigorous version.

The Bounds As A Hierarchy

The bounds form a hierarchy. From most pessimistic to most optimistic:

1. **Wang (noise-induced):** $(1-p)^{2L}$ — applies whenever there is gate noise.
2. **McClean (deep HEA, global cost):** $O(2^{-n})$ — barren plateau.
3. **Ragone (Lie algebraic):** $O(1/\dim \mathfrak{g})$ — depends on ansatz structure.
4. **Marrero (entanglement):** $O(2^{-S})$ — depends on average entanglement.
5. **Subspace 2-design:** $O(1/k)$ — depends on accessible subspace size.
6. **Cerezo (local cost, shallow):** $\Omega(\mathrm{poly}(1/n))$ — trainable.

The right bound depends on the situation:

- **Are you using a deep generic ansatz with global cost?** → McClean / Ragone.
- **Are you using a deep ansatz with local cost?** → Cerezo or Wang (depth/noise).
- **Are you using a structured ansatz with symmetries?** → Subspace 2-design.
- **Are you on noisy hardware?** → Wang dominates.

The practical question is: which of these bounds is *tightest* for your specific case?

How To Use These Bounds In Practice

A workflow for ansatz design using these bounds:

Step 1: Compute (or estimate) the DLA dimension of your candidate ansatz. If polynomial in n, Ragone gives a polynomial bound — likely trainable.

Step 2: Check the cost function locality. If global, you'll need either a shallow ansatz or a structured DLA. If local, Cerezo gives polynomial trainability for shallow circuits.

Step 3: Estimate ansatz entanglement. If average entanglement is volume-law, Marrero predicts a plateau. If area-law or log-law, you're likely fine.

Step 4: Account for noise. Multiply by Wang's $(1-p)^{2L}$ factor. If your depth times noise rate exceeds 1, the noise plateau dominates.

Step 5: Verify with empirical gradient variance. Sample 100 random initializations, compute the gradient norm at each, look at the distribution. If most are below the shot noise floor, you have a plateau in practice.

This workflow is what experienced VQA practitioners do, often informally. The bounds in this node make it formal.

Lower Bounds Vs Upper Bounds

A subtle but important point: **all these are upper bounds on gradient variance**.

An upper bound says "the variance is at most X." If X is small, you're guaranteed a plateau. A lower bound would say "the variance is at least X." If X is large, you're guaranteed *not* to have a plateau.

The literature has many upper bounds (showing plateaus are inevitable in various regimes) and very few lower bounds (showing trainability is guaranteed in various regimes). The Cerezo bound for shallow local-cost circuits is one of the few clean lower bounds.

This asymmetry reflects the difficulty: it's easier to prove "this thing is bad" than "this thing is good." For practical VQA design, the lack of clean lower bounds means you often have to verify trainability empirically rather than rely on theory.

Structural Role

This node is the **trainability theorem dump** of Module 8. It collects the formal trainability bounds, organizes them into a hierarchy, and provides the practitioner workflow for using them.

It directly enables node 8.5 (the explicit expressivity-trainability tradeoff curve), which combines the bounds in this node with the expressivity metrics in 8.1-8.2 to produce the central object of Module 8.

Relations to Other Concepts

- **McClean et al. 2018** — original barren plateau theorem.
- **Cerezo et al. 2021** — local cost refinement.
- **Marrero-Kieferová-Wiebe 2021** — entanglement view.
- **Ragone et al. 2024** — Lie-algebraic theory.
- **Wang et al. 2021** — noise-induced plateaus.
- **Module 4 (Barren Plateaus)** — the qualitative version of all this.

Worked Example — Comparing Bounds On A 6-Qubit Ansatz

Consider a 6-qubit Hardware-Efficient Ansatz with 4 layers and a global cost $C = \langle Z^{\otimes 6} \rangle$.

McClean bound: $\mathrm{Var}[\partial_j C] \leq 2 \cdot 1/(2^6 - 1) = 2/63 \approx 0.032$.

The shot noise floor at $S = 1000$ shots is $\sim 1/\sqrt{S} \approx 0.032$. The gradient is **right at the noise floor**. Training will be very hard.

Cerezo bound (if cost were local): if $C = \sum_i \langle Z_i \rangle$ instead, the bound would be $\Omega(1/n^c)$ for some small c, giving Var $\approx 0.1 - 0.5$ — well above the noise floor. Training would be feasible.

Ragone bound: the DLA of a 4-layer 6-qubit HEA is approximately $2^{12} = 4096$ (full unitary algebra). $P_{\text{purity}} \approx 1$. So Var $\leq 1/4096 \approx 0.0002$. **Tighter than McClean** — predicts a worse plateau than McClean's bound. Empirically, this is consistent with what's observed.

Wang bound at $p = 10^{-3}$ noise: $(1 - 10^{-3})^{2 \cdot 24} \approx 0.95$. Negligible noise contribution at this depth.

Marrero bound: average entanglement of a 4-layer HEA on 6 qubits is approximately the Page value, $S \approx 3$. So $2^{-3} = 0.125$. Looser than McClean but in the same regime.

The verdict: for this configuration, all bounds say "barren plateau." Switching to a local cost (Cerezo) or restricting to a structured ansatz (smaller DLA in Ragone) would make it trainable.

Visualization

Pending. A log-scale plot of gradient variance vs qubit count for: (a) McClean (deep HEA global), (b) Cerezo (shallow HEA local), (c) Ragone (depends on DLA dim), (d) Wang (depends on depth and noise). Each is a different curve, with the shot noise floor marked. Caption: "Different bounds; different regimes; different conclusions."

Exercises

1. For a 4-qubit HEA with 2 layers and global cost $\langle Z_0 Z_1 Z_2 Z_3 \rangle$ with $\|H\| = 1$, what does McClean's bound give? Compare to a shot noise floor at 10000 shots.
2. The Ragone bound is $O(1/\dim \mathfrak{g})$. For a UCCSD ansatz on H_2 (2 electrons in 4 spin orbitals), $\dim \mathfrak{g} = ?$ (Hint: it's the dimension of the particle-number-conserving subalgebra.) What does the bound predict for trainability?
3. (VQA) For a 10-qubit HEA at depth 30 with depolarizing noise $p = 10^{-3}$, is the gradient plateau dominated by McClean (unitary) or Wang (noise)? Show your reasoning.

Research Extensions

- **Lower bounds for structured ansätze** — proving trainability rather than disproving it.
- **Bounds for non-i.i.d. parameter initializations** — most theorems assume uniform; what about smarter init?
- **Bounds under realistic hardware noise** — beyond depolarizing.
- **Bounds for hybrid optimizers** — how does HOPSO change the effective trainability bound?

Cross-links to Other Courses

- **Random Matrix Theory** — Haar measures, t-design moments.
- **Lie Theory** — dynamical Lie algebras (Ragone).
- **Module 4 (Barren Plateaus)** — the qualitative discussion these bounds formalize.
- **Module 6 (Optimization Dynamics)** — how the bounds affect optimizer choice.

The Expressivity-Trainability Tradeoff

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.5

This is the **central node** of Module 8 — the one that combines the expressivity machinery (8.1-8.3) and the trainability bounds (8.4) into a single quantitative statement: **a curve relating how much an ansatz can express to how easily it can be optimized**.

The tradeoff is not a heuristic anymore. It is a *theorem*. The Ragone et al. (2024) Lie-algebraic framework provides the cleanest version, and it says (paraphrasing):

> For an ansatz with dynamical Lie algebra dimension D, the gradient variance is at most $1/D$ and the dimension of the reachable manifold is at most D.

So **expressivity grows linearly with D, trainability shrinks linearly with $1/D$**, and the product (expressivity $\times$ trainability) is bounded.

This is the formal version of every intuition Module 4 and Module 8 have been building toward. This node states it precisely, walks through its implications, and discusses how to navigate the tradeoff in practice.

The Tradeoff In One Equation

For a parameterized circuit with dynamical Lie algebra $\mathfrak{g}$ of dimension D:

$$\underbrace{\dim \mathcal{R}(U) \leq D}_{\text{expressivity bound}} \quad \text{and} \quad \underbrace{\text{Var}[\partial_j C] \leq \frac{P(\mathfrak{g}, H)}{D}}_{\text{trainability bound}}$$

where $P(\mathfrak{g}, H)$ is the cost-purity factor (a quantity that depends on how the cost Hamiltonian aligns with the algebra; typically $O(1)$).

Multiplying:

$$\dim \mathcal{R}(U) \cdot \text{Var}[\partial_j C] \leq P(\mathfrak{g}, H).$$

This is the tradeoff. The product of expressivity (in the dimension sense) and trainability (in the gradient-variance sense) is bounded by a constant. You cannot have both large.

What The Tradeoff Says

A few interpretations:

1. There is no free lunch. Every gain in expressivity comes at a proportional cost in trainability. If you want to reach a larger family of states, you pay with smaller gradients.

2. The boundary is a curve. Plotting (expressivity, trainability) as a 2D point, all achievable ansätze lie below a hyperbola. The hyperbola is the **Pareto frontier** — the boundary where you can't improve one without sacrificing the other.

3. Most ansätze are far from the frontier. The bound is an inequality, not an equality. Many ansätze are strictly worse than the frontier — they have low expressivity *and* low trainability. Only a small subset are *Pareto optimal*.

4. Choosing where to land is the design decision. Given the frontier, the practitioner picks a point. High expressivity, low trainability (deep HEA on small problems where you can afford the optimization cost). Low expressivity, high trainability (UCCSD on chemistry where the symmetries do the work). Or somewhere in the middle.

The Ragone Curve, Explicitly

The Ragone et al. theorem provides an explicit curve.

For an ansatz indexed by a "structure parameter" s (e.g., depth, number of generators, allowed symmetries), as s increases:

- The DLA dimension $D(s)$ increases.
- The reachable manifold dimension grows: $\dim \mathcal{R}(s) \leq D(s)$.
- The gradient variance shrinks: $\mathrm{Var}(s) \leq P/D(s)$.

The achievable region in (expressivity, trainability) space is bounded by:

$$\text{trainability} \cdot \text{expressivity} \leq \text{const.}$$

For specific ansatz families, the curve has been computed. Examples:

Deep HEA family, parameterized by depth L: - $D(L)$ saturates at $4^n - 1$ (full unitary algebra) for $L = O(n)$. - Expressivity grows from polynomial to exponential. - Trainability drops from $O(1)$ to $O(2^{-n})$.

UCCSD family, parameterized by excitation order k: - $D(k)$ grows polynomially in k: $D \sim n^{2k}$. - Expressivity grows polynomially. - Trainability drops polynomially. - *Stays in the polynomial regime* — never hits the exponential plateau.

Symmetry-restricted ansätze, parameterized by the symmetry group G: - D is fixed by the dimension of the G-invariant operator algebra. - Both expressivity and trainability are determined by G, not directly tunable. - The "right" G is problem-specific.

Pareto Frontier And Optimal Ansätze

A Pareto-optimal ansatz is one where you cannot improve expressivity without losing trainability, or vice versa. The Pareto frontier is the curve of such ansätze.

Properties of the frontier:

1. **It is monotone:** moving along the frontier, gaining expressivity always loses trainability and vice versa. No "free improvements."
2. **It is convex** (in the appropriate scaling): the tradeoff has diminishing returns — going from low to medium expressivity costs little trainability, but going from medium to high expressivity costs proportionally more.
3. **Different problems have different optimal points:** the right point on the frontier depends on the cost function (specifically, how much expressivity is needed to represent the optimum).
4. **Most published ansätze are sub-optimal**, but the gap to the frontier is small in the cases studied so far.

Practical takeaway: when designing an ansatz, ask: "is this on the frontier, or below it?" If below, can it be improved (more efficient parameterization) to move it up?

Concrete Tradeoff Examples

The bounds in node 8.4 give numerical predictions you can check against this tradeoff.

Example 1: 6-qubit HEA, depth 1. - Expressivity (DLA dim): ~ 100. - Gradient variance (Ragone): ~ 0.01. - Verdict: low expressivity, moderate trainability. Probably below the frontier.

Example 2: 6-qubit HEA, depth 6. - Expressivity (DLA dim): ~ 4000 (full unitary algebra). - Gradient variance (Ragone): ~ 0.0003. - Verdict: maximum expressivity, minimum trainability. On the (exponential) frontier — barren plateau.

Example 3: 6-qubit UCCSD with 2 electrons. - Expressivity (DLA dim): ~ 35 (single + double excitations subspace). - Gradient variance (Ragone): ~ 0.03. - Verdict: structured, low expressivity, high trainability. On the (polynomial) frontier.

Example 4: 6-qubit Hamiltonian Variational Ansatz, 3 layers. - Expressivity (DLA dim): ~ 200 (problem-specific algebra). - Gradient variance: ~ 0.005. - Verdict: moderate expressivity, moderate trainability. Close to the frontier.

The pattern: **structured (problem-aligned) ansätze are on the frontier; generic ansätze are typically below it**.

Why The Tradeoff Is Tight

A natural question: is the tradeoff *fundamental* (a true theorem about quantum mechanics) or *artifact-of-our-bounds* (only as tight as the bounds, which could be loosened by smarter analysis)?

The honest answer: **partially fundamental, partially artifactual**.

Fundamental part: the relationship between Lie algebra dimension and reachable set is exact (it's a theorem of differential geometry). The relationship between Lie algebra dimension and gradient variance has the right scaling (also a theorem). The product is bounded by a constant that depends only on the cost Hamiltonian's purity.

Artifactual part: the constant $P(\mathfrak{g}, H)$ might be tightenable by smarter analysis. The gradient variance bound is an upper bound — empirically the variance can be smaller, which would make the tradeoff *worse* than predicted (more pessimistic). Lower bounds would be needed to show the tradeoff is tight in both directions.

The 2026 state of the art is that the tradeoff is **definitely valid** as a hyperbolic upper bound, and **probably tight up to constants** for generic ansätze. For special structured ansätze, there may be more room.

Implications For Ansatz Design

The tradeoff turns ansatz design into a constrained optimization problem:

$$\text{maximize trainability}(U) \quad \text{subject to expressivity}(U) \geq E_{\text{required}},$$

where E_{required} is the minimum expressivity needed to contain the answer to the problem at hand.

This formulation makes the design problem clean:

Step 1: Estimate E_{required} from the problem. For chemistry, this is "enough excitations to capture electron correlations." For optimization, "enough expressivity to express the optimal mixing operator." For machine learning, "enough Fourier modes for the data."

Step 2: Find the ansatz family with smallest possible expressivity that still meets E_{required}.

Step 3: Among that family, pick the variant with the largest trainability.

Step 4: Verify empirically.

This is what UCCSD does for chemistry, what QAOA does for optimization, what re-uploading does for machine learning. They are all different *operational embodiments* of the same constrained optimization principle.

Connection To The Thesis

The thesis frames the tradeoff in terms of **regularization**. The argument:

The expressivity-trainability tradeoff says: an unrestricted (highly expressive) ansatz is untrainable. To make it trainable, you must *restrict* it. There are two ways to restrict:

1. Restrict the ansatz — fewer parameters, smaller DLA, more symmetries. This is the *static* restriction, baked into the ansatz design.

2. Restrict the optimizer dynamics — population-based methods, regularized updates, momentum, trust regions. This is the *dynamic* restriction, baked into the optimizer.

The thesis claims **both** are forms of regularization, and the dynamic version is underexploited. Module 5 (regularization) and Module 6 (optimization dynamics) cover the dynamic version; this module covers the static version.

The key insight: **both forms of restriction move you to a different point on the expressivity-trainability tradeoff curve**. Static restriction trades expressivity at design time; dynamic restriction trades effective expressivity at optimization time. They are complementary.

What The Tradeoff Doesn't Say

A few things the tradeoff curve does *not* tell you:

1. It doesn't tell you where on the curve you should be. That depends on the problem.

2. It doesn't tell you which structured ansatz is best. Many ansätze can be at the same point on the curve, but with different fitness for different problems.

3. It doesn't tell you about local minima. The tradeoff is about *global* trainability properties (gradient variance). It doesn't address whether the landscape has many local minima.

4. It doesn't tell you about noise. Hardware noise modifies the effective tradeoff (Wang's bound) but the noise correction is on top of the unitary tradeoff.

5. It doesn't predict generalization. For VQA-as-machine-learning, there's a separate question about how well the ansatz generalizes from training data, which the tradeoff doesn't address.

These are all separate questions that nodes 8.6 (data encoding) and 8.7 (quantum advantage) take up.

Structural Role

This node is the **central theorem** of Module 8. It combines the expressivity machinery (8.1-8.3) with the trainability bounds (8.4) to produce the explicit tradeoff curve, and it provides the practical workflow for designing ansätze in light of the tradeoff.

It is the conceptual climax of Module 8, and the main reference point for Module 9 (Hardware Noise) and Module 10 (Adaptive Control), which discuss how the tradeoff plays out in real deployment.

Relations to Other Concepts

- **Ragone et al. (2024)** — the primary reference for the explicit tradeoff.
- **Holmes et al. (2022)** — earlier connection between expressibility and trainability.
- **Sim-Johnson-Killoran (2019)** — the empirical version of the same intuition.
- **No-free-lunch theorems (Wolpert-Macready 1997)** — classical analogue from optimization.
- **Module 4 (Barren plateaus)** — the qualitative version.

Worked Example — The Tradeoff For A Family Of Ansätze

Consider 6-qubit ansätze with single and double Pauli rotations, parameterized by the number of layers L.

L	DLA dimension	Expressivity (manifold dim)	Variance bound (P/D)	Where on curve
1	24	24	0.04	Below frontier
2	100	100	0.01	Approaching frontier
3	400	400	0.0025	On frontier
4	1500	~1500	0.0007	On frontier
5	3500	3500	0.0003	Saturated
6+	4096	4096 (max)	0.00024	Plateau regime

The product (expressivity × variance) hovers around 1 for layers 3-6, confirming the tradeoff. Below $L = 3$, the product is below 1 — the ansatz is sub-optimally compressing.

To use this ansatz for a problem requiring expressivity ~400 (manifold dimension), $L = 3$ is the right choice. Going to $L = 5$ or $L = 6$ gains nothing (the ansatz can already represent the answer) but loses an order of magnitude in trainability.

This is the kind of analysis the tradeoff curve enables.

Visualization

Pending. A 2D plot of trainability (y-axis, log scale) vs expressivity (x-axis, log scale) with the Pareto frontier curve, several ansatz points (UCCSD, HEA-1, HEA-3, HEA-deep, problem-aligned), and the "below frontier" region shaded. Caption: "The tradeoff is real. Most ansätze are on the curve. The right point depends on your problem."

Exercises

1. For an ansatz with DLA dimension 50 and cost-purity 1, what is the maximum gradient variance? Is this trainable at 1000 shots?
2. The tradeoff says (expr × var) ≤ const. For two different cost Hamiltonians (one global, one local), the constants differ. Which has the larger constant — global or local? Explain.
3. (VQA) Design a 4-qubit ansatz with target expressivity (manifold dimension) ~20 and predict its gradient variance from the tradeoff. Verify by computing the DLA dimension explicitly.

Research Extensions

- **Tighter constants in the tradeoff** — pinning down $P(\mathfrak{g}, H)$ for various ansatz-Hamiltonian combinations.
- **Tradeoff under noise** — combining Ragone with Wang to get the effective tradeoff under hardware noise.
- **Tradeoff for non-standard cost functions** — non-Hermitian observables, multi-objective costs.
- **Tradeoff and Pareto-optimal ansatz search** — automated discovery of ansätze on the frontier.

Cross-links to Other Courses

- **Multi-objective optimization** — Pareto frontiers in classical optimization.
- **Information theory** — capacity-rate tradeoffs (Shannon's noisy channel theorem is the classical analogue).
- **Module 4 (Barren plateaus)** — the qualitative root of the tradeoff.
- **Module 5 (Regularization)** — the dynamics-side complement.

Data Encoding and Expressivity

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.6

So far, Module 8 has treated expressivity in terms of *parameter* dependence: how many states the ansatz can reach as $\boldsymbol{\theta}$ varies. This node generalizes to the **machine-learning setting**, where the ansatz also depends on a *data input* x.

In quantum machine learning (QML), the circuit becomes $U(x, \boldsymbol{\theta})$ — a function of both data and trainable parameters. The expressivity questions are now:

1. **What functions $f(x)$ can the model express** as x varies?
2. **How does the choice of data encoding affect the function class?**
3. **Is there a tradeoff between data-encoding expressivity and trainability** analogous to the parameter-side tradeoff?

The answers reveal that data encoding is *the* key design choice in QML — more important than the parameterized part. The encoding determines the function class entirely; the trainable part only selects within it.

This node walks through the major encoding strategies, the Schuld-Sweke-Meyer Fourier characterization, and the implications for QML model design.

Why Data Encoding Matters

In classical ML, data is fed to a neural network via a single input layer. The network's expressivity grows with depth and width regardless of how the input was fed in (modulo choice of features).

In quantum ML, **the encoding is part of the model**. Different encoding choices give *fundamentally different* function classes, even with identical trainable layers. There is no "universal feature encoding" that lets the trainable layers do all the work.

The reason: a quantum state $|\psi(x)\rangle$ is a *nonlinear* function of x (because of the cosines and sines in the rotation gates). The set of functions you can express as expectation values $\langle\psi(x)|M|\psi(x)\rangle$ is constrained by what kinds of nonlinearities the encoding produces. Trainable parameters can mix and combine these nonlinearities, but they can't introduce nonlinearities that aren't already in the encoding.

So **encoding choice = function class**. Get it wrong and no amount of training will help.

The Major Encoding Strategies

The literature has converged on a few standard strategies.

1. Basis encoding

$$x \in \{0,1\}^n \to |x\rangle.$$

Each bit of x becomes a computational basis state. Simple but only works for binary inputs.

Pros: trivial to implement; preserves all information. **Cons:** doesn't generalize to continuous inputs; doesn't introduce useful nonlinearities.

2. Angle encoding

$$x \in \mathbb{R}^n \to \bigotimes_{i=1}^{n} R_Y(x_i)|0\rangle.$$

Each component of x becomes a rotation angle on a separate qubit. The state is a product of single-qubit rotations.

Pros: simple, continuous-friendly, hardware-friendly. **Cons:** the function class is restricted to first-order Fourier modes (linear in $\cos x_i, \sin x_i$).

3. Amplitude encoding

$$x \in \mathbb{R}^{2^n}, \|x\| = 1 \to \sum_i x_i|i\rangle.$$

The data vector becomes the amplitudes of a quantum state. **Exponential compression** — 2^n dimensional vectors fit in n qubits.

Pros: maximally compact, supports arbitrary inputs. **Cons:** state preparation is expensive — $O(2^n)$ gates in general. Often a bottleneck.

4. Re-uploading encoding

$$x \in \mathbb{R}^d \to U_L(\boldsymbol{\theta}_L)S(x)U_{L-1}(\boldsymbol{\theta}_{L-1})S(x)\cdots S(x)U_0(\boldsymbol{\theta}_0)|0\rangle,$$

where $S(x)$ is a data-encoding layer (e.g., a tensor product of $R_Y(x_i)$) repeated L times, interleaved with trainable layers $U_l(\boldsymbol{\theta}_l)$.

Pros: the function class grows polynomially with L — Fourier modes up to degree L become accessible. Universal function approximator in the limit $L \to \infty$. **Cons:** depth scales with desired Fourier degree.

5. Hamiltonian encoding

$$x \in \mathbb{R} \to e^{-ixH}|0\rangle,$$

where H is a problem-specific Hamiltonian. The data is encoded as evolution time.

Pros: physically natural, exploits quantum dynamics directly. **Cons:** Hamiltonian needs to be implementable; expressivity depends on H's spectrum.

The dominant choice in 2026 QML is **re-uploading**, because it gives the most controllable expressivity. Angle encoding is the simplest baseline.

The Schuld-Sweke-Meyer Theorem (Detailed)

The most important theoretical result about data encoding is the **Fourier characterization** of re-uploading models.

Setup. Consider a 1D input $x \in \mathbb{R}$ and a re-uploading model

$$f(x;\boldsymbol{\theta}) = \langle 0|U^\dagger(x,\boldsymbol{\theta})MU(x,\boldsymbol{\theta})|0\rangle,$$

where the data is encoded via e^{-ixH} gates with Hamiltonian H, and there are L such encoding gates interleaved with trainable layers.

Theorem (Schuld-Sweke-Meyer 2021). $f(x;\boldsymbol{\theta})$ is a *truncated Fourier series* in x:

$$f(x;\boldsymbol{\theta}) = \sum_{\omega\in\Omega} c_\omega(\boldsymbol{\theta}) e^{i\omega x},$$

where the **frequency set** Ω is determined by the spectrum of H and the depth L, and the **Fourier coefficients** $c_\omega(\boldsymbol{\theta})$ are determined by the trainable circuits.

Frequency set details: if H has eigenvalues $\{\lambda_i\}$, then Ω is the set of integer linear combinations of differences $\lambda_i - \lambda_j$ with at most L terms.

Implications:

1. **The encoding determines what frequencies are reachable.** Different H gives different Ω.
2. **The depth controls the Fourier degree.** L layers give frequencies up to roughly degree L.
3. **The trainable parameters control coefficients, not frequencies.** No amount of training will give you a frequency that isn't in Ω.
4. **The function class is fully characterized.** Two re-uploading models with the same encoding and depth have the same function class — only the coefficient flexibility may differ.

This is the cleanest theoretical result in QML expressivity, and it reframes data encoding as **frequency-set design**: pick the encoding that gives you the frequencies you need.

Multi-Dimensional Inputs

For $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, the Fourier characterization generalizes:

$$f(\mathbf{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{\omega}\in\Omega^d} c_{\boldsymbol{\omega}}(\boldsymbol{\theta}) e^{i\boldsymbol{\omega}\cdot\mathbf{x}},$$

where Ω^d is now a *set of frequency vectors* rather than scalars.

The structure of Ω^d depends on:

1. Whether the data components are encoded in **the same qubit** (in which case the frequencies couple) or **different qubits** (in which case the frequencies separate).
2. The **depth** of the re-uploading.
3. The **mixing** of data components in the trainable layers (a non-trivial entanglement layer between encoding gates can mix frequencies that would otherwise be separable).

This multi-dimensional case is more complex and less fully characterized than the 1D case, but the qualitative picture is the same: **encoding determines what's expressible; trainable parameters fine-tune within that space**.

The Encoding-Trainability Tradeoff

There is an analogue of the parameter-side tradeoff (8.5) for the data-encoding side.

Statement (informal). Increasing the data-encoding depth gives a larger Fourier set (more expressivity) but, all else equal, also brings the model closer to the barren plateau regime in the parameters. So there is a similar tradeoff: more data-encoding depth = more function expressivity = harder to train.

Quantitative version. Schuld and Killoran (2022) showed that for re-uploading models on n qubits with L encoding layers, the gradient variance scales as $O(2^{-nL})$ for generic encoding and trainable layers. The plateau is *both* in qubit count and encoding depth.

Mitigation: as in the parameter-side tradeoff, structured encodings (with restricted frequency sets) have better trainability. The right encoding for QML is the one that includes the frequencies needed for the data, but no more.

Encoding As Feature Engineering

A useful classical analogy: **data encoding in quantum ML is like feature engineering in classical ML**.

In classical ML, a kernel or feature map $\phi(x)$ lifts the data into a feature space where a linear (or simple) classifier can separate it. The expressive power of the model is largely determined by the feature map, not the classifier.

In quantum ML, the data encoding plays the same role: it lifts x into a quantum state $|\psi(x)\rangle$, where the trainable observable $\langle M(\boldsymbol{\theta})\rangle$ performs the "linear classification."

The implications:

1. **Expressivity comes mostly from the encoding.** If the encoding is linear in x, the model is at most a linear function of trigonometric features.
2. **Universal QML requires universal encoding.** If you want to learn any function, you need a re-uploading encoding with enough Fourier modes.
3. **Domain knowledge goes into the encoding.** Just as in classical ML, choosing the right features (encoding) for the problem is the most important design decision.

This analogy is not perfect — quantum encodings introduce specifically *trigonometric* nonlinearities that don't have a clean classical counterpart — but it's close enough to import most of the intuition from classical feature engineering.

The Quantum Kernel Perspective

A related view: the encoding defines a **quantum kernel**.

$$K(x, x') = |\langle\psi(x)|\psi(x')\rangle|^2.$$

This is an inner product between encoded data points, computable on a quantum computer. A kernel-based classifier (SVM, ridge regression) can be applied directly.

Properties of quantum kernels:

1. **They capture the encoding's similarity structure.** Two inputs that map to similar quantum states have high kernel value.
2. **They can be exponentially powerful.** For some encodings, the quantum kernel computes inner products in an exponentially-large feature space (related to the IQP encoding of Havlicek et al. 2019).
3. **They sidestep the trainability problem.** Kernel methods don't require gradient descent over a deep parameterized circuit — you compute the kernel matrix and run a classical SVM.

The downside: kernel methods scale as $O(N^2)$ with dataset size N, where N is the number of training examples. For large datasets, this is prohibitive.

The kernel view is the clean theoretical alternative to variational QML, and recent work (Schuld 2021, Liu et al. 2021) suggests it's often *better* than variational training when the right kernel is available.

Practical Encoding Recommendations

For practitioners building QML models in 2026:

For low-dimensional regression / classification ($d < 10$): - Re-uploading with angle encoding, $L = 2 - 5$ layers. - Frequency set design: pick H so the frequencies match the expected smoothness of the target function.

For high-dimensional data ($d > 100$): - Amplitude encoding (if state preparation is feasible). - Or aggressive feature reduction (PCA) before encoding.

For structured data (images, time series): - Hamiltonian encoding with a structured H (e.g., a translation-invariant one for images). - Or specialized encodings (data convolutional layers, quantum CNNs).

For exploratory work: - Start with simple angle encoding to establish a baseline. - Add re-uploading layers until performance plateaus or trainability degrades. - Switch to kernel methods if variational training stalls.

The dominant message: **think about encoding before trainable parameters**. The encoding sets the ceiling on what you can learn; the parameters only navigate within that ceiling.

Structural Role

This node is the **machine-learning extension** of Module 8. It generalizes the expressivity framework from parameters to data inputs, introduces the Schuld-Sweke-Meyer Fourier characterization, and establishes encoding choice as the primary design decision in QML.

It enables node 8.7 (Quantum Advantage Criteria) by clarifying what kinds of functions quantum models can express that classical models can't — the prerequisite for any advantage claim.

Relations to Other Concepts

- **Schuld-Sweke-Meyer (2021)** — the Fourier characterization theorem.
- **Havlicek et al. (2019)** — IQP-based quantum kernels.
- **Schuld (2021)** — quantum kernel methods.
- **Pérez-Salinas et al. (2020)** — original re-uploading paper.
- **Module 4 (Barren plateaus)** — the trainability cost of expressive encodings.

Worked Example — Frequency Set For A Simple Re-Uploading Model

Consider a 1-qubit re-uploading model:

$$U(x, \boldsymbol{\theta}) = R_Y(\theta_4) e^{-ixZ} R_Y(\theta_3) e^{-ixZ} R_Y(\theta_2) e^{-ixZ} R_Y(\theta_1).$$

Three data encoding gates with $H = Z$, so the eigenvalues are $\{+1, -1\}$ and the differences are $\{2, 0, -2\}$.

Compute the frequency set. With $L = 3$ encoding layers and frequency differences $\{0, \pm 2\}$, the reachable frequencies are integer combinations of $\{2, 0, -2\}$ with at most 3 terms. This gives $\{-6, -4, -2, 0, 2, 4, 6\}$ — degrees up to 6.

Function class. $f(x; \boldsymbol{\theta})$ is a Fourier series with frequencies in $\{-6, -4, -2, 0, 2, 4, 6\}$ — i.e., a degree-6 trigonometric polynomial in $2x$.

Function expressibility. Can this model express $\sin(3x)$? The frequencies in $\sin(3x)$ are ± 3, which are *not* in the model's frequency set (which only has even frequencies). So no — the model cannot exactly express $\sin(3x)$.

To get $\sin(3x)$, you'd need $H = 3Z/2$ so the frequency differences include ± 3, or a different encoding that produces odd frequencies.

This is the kind of analysis you do *before* training: check that your encoding can represent the function family you care about.

Visualization

Pending. A diagram showing two re-uploading models: (a) with $L = 1$ encoding layer, frequency set $\{0, \pm 2\}$, expressing only low-frequency trigonometric functions; (b) with $L = 4$ encoding layers, frequency set $\{0, \pm 2, \pm 4, \pm 6, \pm 8\}$, expressing higher-frequency functions. Side by side with target function plots that the models can/can't reach. Caption: "Encoding sets the frequency budget. Training spends it."

Exercises

1. For a 1D re-uploading model with $H = X$ and $L = 2$ encoding layers, compute the frequency set. Which simple functions $\sin(\omega x), \cos(\omega x)$ can it represent for integer ω?
2. Can amplitude encoding represent the function $f(x) = x^2$? If yes, how? If no, what would you change?
3. (VQA) For a 2D classification task with circular decision boundary, design a re-uploading encoding that has the right frequency content. Which value of L is sufficient?

Research Extensions

- **Frequency-set design** — algorithmic methods for picking H and L to match a target function class.
- **Encoding-trainability optimization** — joint optimization of encoding and parameters.
- **Quantum kernels with provable advantage** — encodings that produce kernels not classically computable.
- **Multi-dimensional Fourier characterization** — extending Schuld-Sweke-Meyer to high-dimensional inputs.

Cross-links to Other Courses

- **Fourier Analysis** — trigonometric polynomials, Fourier series, Stone-Weierstrass.
- **Kernel Methods (classical ML)** — kernel SVMs, RKHS, feature maps.
- **Module 4 (Barren plateaus)** — trainability cost of deep encodings.
- **Module 8 Section 8.7 (Quantum Advantage)** — what encodings enable provable quantum advantage.

Quantum Advantage Criteria

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Expressivity vs Trainability **Node:** 8.7

This is the **closing node** of Module 8 — and arguably the most important one for anyone wondering whether all of this VQA machinery actually *matters*.

The question is simple: **when does a VQA actually beat a classical algorithm at the same task?**

The answer, as of 2026, is honest and uncomfortable: **almost never, with current hardware**. There are theoretical regimes where quantum advantage is provable. There are heuristic regimes where it's plausible. There are application-specific regimes where it's claimed. But for the vast majority of VQA experiments published in the past decade, the honest assessment is that a classical algorithm of comparable effort would do at least as well.

This node walks through the criteria for quantum advantage, the regimes where it's been claimed, the regimes where it's been *disproved* (a sobering category), and the open questions about where future advantage might come from. It is the essential reality check for the rest of Module 8 and indeed for the entire VQA literature.

Three Notions Of Quantum Advantage

The phrase "quantum advantage" means different things in different contexts. The three main notions:

1. Asymptotic advantage (provable). A quantum algorithm has an asymptotically better runtime (poly vs super-poly, or poly vs exp) than the best classical algorithm for the same task. Examples: Shor's algorithm, Grover's algorithm. Holds for all sufficiently large inputs.

2. Empirical advantage (heuristic). On a specific benchmark or instance, the quantum algorithm produces better results (lower cost, faster) than a specific classical baseline. Holds only for the specific benchmark; may not generalize.

3. Sample complexity advantage (practical). The quantum algorithm uses fewer samples (less data) to achieve a given accuracy than any classical algorithm. Especially relevant for QML, where data is expensive.

VQAs are usually assessed in the second sense (empirical) or, more recently, the third (sample). The first (asymptotic) is rarely on offer for VQAs, because the variational structure makes asymptotic analysis hard.

What Provable Asymptotic Advantage Requires

For a VQA to give provable asymptotic advantage, several conditions must hold:

1. The classical baseline is well-characterized. You must know the best classical algorithm for the same task, or have a complexity-theoretic lower bound. For most problems VQAs target (chemistry ground states, optimization), the best classical algorithms are heuristics — there's no clean lower bound.

2. The quantum algorithm has a provable runtime. VQAs have variational structure with classical optimization in the loop, and the convergence of the classical optimizer is itself unprovable in general. So the *whole* VQA runtime is unprovable.

3. The quantum part actually does something exponentially hard for classical methods. Many quantum-inspired classical algorithms (DMRG, neural network states, Monte Carlo) can do what shallow VQAs do, sometimes faster.

Net result: no VQA has provable asymptotic advantage as of 2026. Several have *plausible* asymptotic advantage based on conjectured complexity classes (e.g., assuming $P \neq NP$), but no rigorous proof.

Empirical Advantage And The Dequantization Problem

The empirical advantage story is complicated by **dequantization** — the discovery that many quantum algorithms thought to give exponential speedup actually have classical algorithms with similar performance.

Notable dequantizations:

- **Tang's quantum-inspired classical recommendation algorithm (2018):** showed that the quantum recommendation algorithm of Kerenidis and Prakash, which appeared to give exponential speedup, has a classical analogue with polynomial-time runtime.
- **Quantum-inspired classical PCA, regression, etc.** Many "quantum machine learning" speedups have classical analogues.
- **Tensor network simulations** of VQAs that claimed empirical advantage have repeatedly shown that DMRG or MPS methods reach the same accuracy with less compute.

The pattern: every time a VQA claims empirical advantage, the classical ML/numerical community responds with a quantum-inspired algorithm that closes the gap.

This is sobering. It does *not* mean quantum advantage is impossible — it means **the bar is high**. To claim a quantum advantage, you need to show your problem instance is hard for *all* known classical methods (DMRG, NN-states, Monte Carlo, etc.), not just one specific baseline.

When Quantum Advantage Is Plausible For VQAs

Despite the dequantization caution, there are regimes where VQA quantum advantage remains plausible:

1. Strongly correlated electron systems beyond DMRG range. DMRG works well for 1D and quasi-1D systems but degrades in 2D and 3D. For 2D Hubbard model at intermediate doping, no classical method scales well. A VQE on a 2D Hubbard ansatz may be the right tool — *if* the VQA can be made to converge.

2. Quantum chemistry with strong static correlation. Coupled-cluster methods (CCSD, CCSD(T)) fail for systems with strong static correlation (transition metals, bond-breaking). UCCSD via VQE might handle these — but the trainability and noise problems make it hard to demonstrate.

3. Sampling problems related to BosonSampling/IQP. Some sampling tasks are *provably* classically hard (Aaronson-Arkhipov for BosonSampling, similar for IQP circuits). These are not VQAs in the strict sense, but they share architectural features. A VQA whose output distribution is in this hardness class would inherit the advantage.

4. Quantum simulation of dynamics. Time evolution under a Hamiltonian is a natural quantum task. If the VQA prepares states involved in a hard simulation problem, the broader simulation pipeline can give advantage. The VQA part is typically state preparation, not the simulation itself.

5. Tasks with quantum-data inputs. If the data x is itself quantum (e.g., outputs of a quantum sensor), classical methods need to first measure and reconstruct, which is exponentially expensive. A quantum ML model can process quantum data directly, with provable sample complexity advantage (Huang et al. 2022).

6. Specific structured problems with hidden symmetries. Some VQAs exploit problem-specific structure (e.g., translation invariance, particle conservation) in ways classical algorithms don't naturally exploit. For these problems, a custom-designed VQA may beat generic classical methods.

These six are the **realistic** regimes for VQA advantage. None has been definitively demonstrated as of 2026, but all are active research areas.

The Sample Complexity Frontier

A more recent and more promising frontier: **sample complexity advantages**.

Setup. Suppose you want to learn some property of an unknown quantum state or process. A classical algorithm needs many samples (measurements) to reconstruct the state and then compute the property. A quantum algorithm can sometimes compute the property directly, with exponentially fewer samples.

Results (Huang-Kueng-Preskill 2020, Aharonov et al. 2022):

- **Shadow tomography:** $O(\log d)$ samples suffice for many properties of a d-dimensional state, vs $O(d)$ classically.
- **Quantum learning advantage:** for certain learning tasks (e.g., learning quantum process tomography), a quantum learner uses exponentially fewer samples than any classical learner, and this is *provable*.
- **Quantum data analysis:** the same task can require exponentially less data with quantum access.

Connection to VQAs: if a VQA is part of a sample-efficient protocol (e.g., learning a property of a state that the VQA prepared), the sample complexity advantage carries over. This is one of the cleanest theoretical justifications for VQA-based ML in 2026.

Caveat: sample complexity is not the same as runtime complexity. A protocol with low sample complexity may still have high computational complexity. The advantage is real but limited.

What Disproves Quantum Advantage

A useful counter-frame: what kinds of evidence *would* disprove a VQA quantum advantage claim?

1. **A classical algorithm matches the quantum algorithm's performance.** (Dequantization.)
2. **The classical baseline was incorrectly chosen.** Many "advantage" claims compare against weak baselines.
3. **The quantum algorithm doesn't actually scale.** Many VQAs are demonstrated at small n and don't extrapolate.
4. **The quantum algorithm is dominated by classical processing.** Some VQAs do most of their work classically (in the optimizer) and the quantum part is incidental.
5. **The benchmark doesn't reflect a real problem.** Synthetic benchmarks designed to be quantum-friendly don't transfer to real applications.

A *credible* quantum advantage claim must address all five of these objections. Most published VQA-advantage claims address one or two and ignore the rest.

This is the standard for the field in 2026, and it has gotten significantly stricter over the past five years as dequantization has matured.

The Honest 2026 Assessment

Putting it all together, here is the honest 2026 picture for VQA quantum advantage:

1. No proven asymptotic advantage for any VQA on any natural problem.

2. Several plausible empirical advantages in narrow regimes: - Strongly correlated chemistry / condensed matter beyond classical reach. - Sample complexity advantages for quantum data. - Specific sampling problems inherited from quantum supremacy circuits.

3. Many *disproven* advantages in broad regimes: - Most QML claims for classical data have been dequantized or matched by classical baselines. - Most VQA chemistry below 30 qubits has classical analogues that perform as well or better. - Most QAOA claims for combinatorial optimization don't hold up against modern classical heuristics.

4. The hardware bottleneck. Even if a VQA has theoretical advantage, current hardware (NISQ, ~100 qubits, ~1% noise per gate) cannot demonstrate it. The advantage *might* appear at fault-tolerant scales, but that is years away.

5. The most active research frontier: finding provable sample complexity advantages for VQAs on quantum data, where classical methods are exponentially worse by structural necessity.

This is uncomfortable but it is the field's current state. Anyone working on VQAs should know it.

What Would Change The Picture

For VQA quantum advantage to become *demonstrated* (not just plausible), several things would need to happen:

1. Better hardware. Lower noise ($p < 10^{-4}$ per gate), more qubits ($n > 200$ high-fidelity), and longer coherence times. This is the most important factor and the hardest to control.

2. Better ansätze. Ansätze designed using the expressivity-trainability tradeoff (Module 8.5), with explicit symmetry restrictions matched to the problem. This is a research area that this thesis contributes to.

3. Better optimizers. Hybrid optimizers (HOPSO, Module 6.8) and dynamics-side regularization (Module 5) that can navigate the difficult landscapes. Also a thesis contribution area.

4. Honest classical baselines. The community needs to stop comparing VQAs against weak classical baselines (random search, untuned SGD) and start comparing against state-of-the-art (DMRG, neural quantum states, professional optimization solvers).

5. Focus on quantum-data tasks. Where the input is naturally quantum, the comparison is more honest because classical methods lose information at the input boundary.

If all five happen, VQAs may demonstrate genuine advantage in the late 2020s or 2030s. If only some happen, VQA advantage will remain a theoretical possibility for longer.

Closing Module 8

Module 8 has been a long arc:

- **8.1-8.3:** what expressivity means (definitions, metrics, universal approximation).
- **8.4:** what trainability means (gradient variance bounds, hierarchy of theorems).
- **8.5:** the explicit tradeoff curve relating them (Ragone Lie-algebraic theory).
- **8.6:** the data-encoding extension for quantum machine learning.
- **8.7 (this node):** the honest reality check on whether any of this gives quantum advantage.

The narrative thread: **the expressivity-trainability tradeoff is the central design constraint for VQAs**, and within that constraint, only narrow regimes plausibly give quantum advantage. The thesis's contribution is to provide tools (regularization, hybrid optimizers, dynamics-side framing) for navigating the constraint better — to find more of those narrow regimes, and to make them work in practice.

Module 9 (Hardware Noise Models) and Module 10 (Adaptive Control Systems) take this theoretical foundation and ask what it looks like in real hardware deployment.

Structural Role

This node is the **reality check** of Module 8 and of the VQA section of the course as a whole. It states the criteria for quantum advantage, walks through what's been claimed and disproven, and gives an honest 2026

assessment of the field's state.

It is the natural endpoint of the theoretical arc: having built up the machinery for understanding VQAs in detail, it asks the meta-question of whether VQAs are *worth* understanding in detail, and answers "yes, but with caveats."

Relations to Other Concepts

- **Tang (2018)** — quantum-inspired classical recommendation algorithm, the start of dequantization.
- **Aaronson-Arkhipov BosonSampling** — sampling-based quantum supremacy.
- **Huang-Kueng-Preskill (2020)** — classical shadows and learning quantum data.
- **Aharonov et al. (2022)** — provable sample complexity advantages.
- **Module 4 (Barren Plateaus)** — the trainability obstacle that constrains advantage.

Worked Example — Comparing VQE Against DMRG On A Spin Chain

Consider a 20-site Heisenberg spin chain — a benchmark VQA target.

Classical baseline: DMRG. - Finds the ground state to chemical accuracy in seconds on a laptop. - Scales to ~100 sites without difficulty. - Has been the gold standard for 1D spin systems for 30 years.

VQE attempt (2026 hardware, ~100 qubits, 1% noise): - Hardware-Efficient Ansatz, depth ~10. - 1000-shot gradient estimates, Adam optimizer. - Converges (after careful initialization) to ~1% accuracy. - Total runtime: hours of quantum compute time.

Comparison: - DMRG: seconds, much higher accuracy. - VQE: hours, lower accuracy. - DMRG wins decisively.

For 2D Heisenberg (20×20 lattice): - DMRG accuracy degrades sharply with width — limited to ~10 wide. - VQE on 400 qubits: not yet achievable on real hardware. - *In principle*, this is where VQE could win, if hardware were available.

The honest assessment: for benchmarks where we *can* compare, classical methods win. For benchmarks where we *can't* compare yet (very large 2D lattices), nobody knows. Hardware progress will determine the outcome.

Visualization

Pending. A 2D plot with axes "system size" and "accuracy" showing the regimes where DMRG, neural network states, Monte Carlo, and VQE are competitive. DMRG dominates 1D, NN-states dominate moderate dimensions, Monte Carlo dominates fermionic systems, and a small region in the upper-right is "where VQE might eventually win." Caption: "Quantum advantage is a small slice of problem space, currently empty."

Exercises

1. List three problems where, in principle, a VQA could beat the best known classical algorithm. For each, identify the classical baseline you're comparing against.
2. The dequantization argument says that many quantum ML speedups have classical analogues. Can amplitude encoding be dequantized? Explain.
3. (VQA) For a 30-qubit VQE on a chemistry molecule (e.g., LiH), estimate the total quantum compute time needed to reach chemical accuracy at current noise levels. Compare to DMRG's runtime on the same problem.

Research Extensions

- **Provable quantum advantage on quantum data** — extending the Huang et al. results to broader classes of problems.
- **VQA + classical preprocessing pipelines** — using VQAs as one component of a larger workflow, where the advantage may be in the integration.
- **Honest benchmarking standards** — methodology for VQA comparisons that resists motivated reasoning.
- **Quantum-classical hybrid advantage** — regimes where the joint quantum + classical algorithm beats either alone.

Cross-links to Other Courses

- **Computational Complexity** — BQP, NP, polynomial hierarchy, complexity-class separations.
- **Classical Numerical Methods** — DMRG, neural quantum states, Monte Carlo (the competitors).
- **Module 9 (Hardware Noise Models)** — the hardware limits on advantage.
- **Module 4 (Barren Plateaus)** — the trainability obstacle that constrains advantage.