How Measurement Actually Works in a VQA

Variational Quantum Algorithms · Module 2 — Parameterized Circuits

Node 2.7

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Parameterized Circuits **Node:** 2.7

This node is for the reader who has seen $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ written down a hundred times and is finally ready to ask: **what physically happens, on a real device, in order to produce that scalar?**

The mechanics are not glamorous. They involve a lot of basis changes, a lot of computational-basis readouts, a lot of bit-counting, and a lot of statistical aggregation. But understanding them concretely is what separates a practitioner from a paper-reader, and it is the *why* behind several key facts that show up later in the course (variance scaling, measurement grouping, shot noise, hardware bias).

The Goal

We want to estimate

$$C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle, \quad H = \sum_{j=1}^{M} h_j P_j.$$

The strategy: estimate each Pauli expectation $\langle P_j\rangle$ separately, then take the linear combination. This works because expectation is linear: $\langle H\rangle = \sum_j h_j \langle P_j\rangle$.

So the entire problem reduces to: **how do you measure a Pauli string P on a state prepared by a circuit?**

Measuring a Single Pauli String: The Recipe

For a Pauli string like $P = X_0 \otimes Z_1 \otimes Y_2 \otimes I_3$ on 4 qubits, the recipe is:

Step 1: Prepare $|\psi\rangle$. Run the ansatz circuit $U(\boldsymbol{\theta})$ on the device, starting from $|0\rangle^{\otimes n}$.

Step 2: Rotate each non-Z, non-I qubit into the Z basis. - For each X entry in P: apply a Hadamard gate H to that qubit. - For each Y entry in P: apply $S^\dagger H$ to that qubit (equivalently $R_X(\pi/2)$). - For each Z entry: do nothing — it's already in the Z basis. - For each I entry: do nothing — measurement of identity is trivial (always $+1$).

After this rotation, the Pauli string has been mapped to a string of Z's and I's, and measuring it in the **computational basis** (which is the Z basis on every qubit) gives the eigenvalue of P.

Step 3: Read out every qubit in the computational basis. The hardware projects each qubit onto $|0\rangle$ or $|1\rangle$, giving a bitstring $b \in \{0,1\}^n$.

Step 4: Compute the eigenvalue from the bitstring. For each qubit position where P has a non-identity entry (X, Y, or Z after rotation, all now Z), the eigenvalue contribution is $(-1)^{b_i}$ ($+1$ for outcome 0, -1 for outcome 1). The full Pauli eigenvalue is the product over all non-identity positions:

$$\text{eigenvalue}(b) = \prod_{i \in \text{support}(P)} (-1)^{b_i}.$$

This is ± 1.

Step 5: Repeat N times and average. Run the circuit N times — each is a "shot" — and average the eigenvalues:

$$\widehat{\langle P \rangle} = \frac{1}{N} \sum_{k=1}^{N} \text{eigenvalue}(b^{(k)}).$$

This is your estimator. It is unbiased: $\mathbb{E}[\widehat{\langle P \rangle}] = \langle P \rangle$. Its variance is bounded by $1/N$ (since each sample is in $[-1, 1]$).

What This Looks Like For The Whole Hamiltonian

Multiplying the protocol across all M Pauli strings:

```
For each Pauli string P_j in H:
    For shot = 1 to N_j:
        Prepare |psi(theta)>
        Rotate to P_j's measurement basis
        Read out the bitstring
        Compute the eigenvalue
    Average to get hat<P_j>

Final cost estimate = sum_j h_j * hat<P_j>
```

Per cost evaluation: $M \times N$ total circuit runs, where M is the number of Pauli strings and N is the per-string shot count. For H_2 (5 strings after tapering), with $N = 1000$ shots each, that's 5000 circuit executions per cost evaluation.

Each evaluation takes wall-clock time on the order of seconds to minutes on a real device, depending on queue depth and circuit length.

Variance and the $1/\sqrt{N}$ Scaling

The estimator $\widehat{\langle P \rangle}$ is a sample mean of ± 1 random variables. By standard statistics:

$$\text{Var}[\widehat{\langle P \rangle}] = \frac{1 - \langle P \rangle^2}{N}, \quad \text{StdDev}[\widehat{\langle P \rangle}] = \sqrt{\frac{1 - \langle P \rangle^2}{N}} \leq \frac{1}{\sqrt{N}}.$$

The standard deviation shrinks as $1/\sqrt{N}$. **This is the fundamental scaling that bounds VQA cost evaluation accuracy.** To halve the error, quadruple the shots.

For the full Hamiltonian, the variance is

$$\mathrm{Var}[\widehat{\langle H\rangle}] = \sum_j \frac{h_j^2(1-\langle P_j\rangle^2)}{N_j} \le \sum_j \frac{h_j^2}{N_j}.$$

If you spend the same N shots on each term, the total cost-estimator variance is bounded by $\frac{1}{N}\sum_j h_j^2$. For molecules, $\sum_j h_j^2$ can be on the order of 1-100 hartree^2, which means even moderate accuracy (millihartree) requires hundreds of thousands of shots per evaluation.

This is a very tight constraint, and it is the reason **measurement grouping** (node 2.8) is so important — it is the only way to amortize measurements across multiple Pauli terms and bring the per-evaluation shot cost into the workable range.

What Hardware Adds On Top

The recipe above assumes ideal hardware. On a real device, several extra effects show up:

1. Gate errors during state preparation. Each gate in $U(\boldsymbol{\theta})$ has some error rate p_g. After 100 gates with $p_g = 10^{-3}$, the prepared state has fidelity $\approx e^{-0.1} \approx 0.9$ to the ideal state. The measured expectation values are biased toward those of the noisy state.

2. Readout error. The "measure in the Z basis" step itself has an error rate, typically 1-5%. A qubit that is actually in $|0\rangle$ is reported as 1 with probability p_{readout}, and vice versa. This adds bias to all expectation values.

3. Crosstalk. Measuring one qubit can subtly disturb the others, creating correlations between supposedly independent measurements.

4. Calibration drift. Gate parameters and readout fidelities change over time. A measurement made at the start of a queue may have slightly different bias than one made an hour later.

All of these turn the ideal recipe into a noisy approximation. The key consequence: **the estimator is biased, not just noisy.** The bias depends on the hardware and changes over time. Unbiased estimation via post-processing (readout error mitigation, zero-noise extrapolation, probabilistic error cancellation) is an active research area.

What The Optimizer Sees

After all of this, the optimizer is handed a single number per evaluation: $\widehat{C}(\boldsymbol{\theta}_k)$. This number is

- biased (by hardware noise and possibly by mitigation procedures),
- noisy (by shot statistics and time-correlated drift),
- delayed (by queue and execution time),
- expensive (each evaluation consumed $M \times N$ circuit runs).

This is the operational meaning of the "stochastic objective generator" abstraction from node 1.8. The abstraction is correct, but **the numbers behind it come from a long pipeline that you have to design to control variance and bias**.

The thesis's experimental setup on simulated hardware fixes the shot count at $N = 1024$ per term and runs for tens of thousands of total circuit executions per optimization run. On real hardware, the equivalent can take hours to a day.

Structural Role

This is the **operational unmasking** node of Module 2. It exposes what $\widehat{C}(\boldsymbol{\theta})$ actually means in terms of physical operations, and connects the abstract Pauli decomposition (2.5) to the noisy scalar the optimizer sees.

Forwards to 2.8 (measurement grouping), which is the optimization step that comes immediately after this raw recipe.

Relations to Other Concepts

- **Quantum measurement (Born rule)** — the underlying physics. $P(b|\psi) = |\langle b|\psi\rangle|^2$.
- **POVMs (positive operator-valued measures)** — the most general measurement formalism; the protocol above is a special case.
- **Direct fidelity estimation** — protocols for estimating state fidelity using random Pauli measurements.
- **Classical shadows** — efficient protocols that estimate many observables from a single measurement set; an active alternative to per-Pauli measurement.
- **Readout error mitigation** — post-processing to remove bias from readout errors.

Worked Example — Measuring $\langle X_0 Z_1\rangle$ on a Bell State

Suppose $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and we want to estimate $\langle X_0 Z_1\rangle$.

Step 1: prepare the Bell state via $H_0, \mathrm{CNOT}_{0,1}$ on $|00\rangle$.

Step 2: rotate qubit 0 (X position) into Z basis by applying H. Qubit 1 (Z position) needs no rotation.

The state after rotation is:

$$H_0|\psi\rangle = H_0 \cdot \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|+0\rangle + |-1\rangle) = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle - |11\rangle).$$

Step 3: measure both qubits in computational basis. Each outcome (b_0, b_1) occurs with probability $|\text{amplitude}|^2 = 1/4$.

Step 4: the eigenvalue is $(-1)^{b_0}(-1)^{b_1}$. - 00 $\to +1$ (prob 1/4) - 10 $\to -1$ (prob 1/4) - 01 $\to -1$ (prob 1/4) - 11 $\to +1$ (prob 1/4)

Expectation: $\frac{1}{4}(+1 - 1 - 1 + 1) = 0$.

So $\langle X_0 Z_1\rangle = 0$ for the Bell state. (Sanity check: this is correct — the Bell state has zero expectation for any Pauli string except the four stabilizers $II, ZZ, XX, -YY$.)

Step 5: repeat N times. The sample mean converges to 0 with variance $1/N$.

Visualization

Pending. A circuit diagram with three layers labeled: "(1) state prep — $U(\boldsymbol{\theta})$", "(2) basis change — H, S†H per Pauli letter", "(3) Z-basis readout". A bracket below indicating "this entire circuit runs N times". A separate annotation on the right showing the bitstring → eigenvalue computation.

Exercises

1. For the state $|\psi\rangle = R_Y(\theta)|0\rangle$, describe the measurement protocol for estimating $\langle X\rangle$ and $\langle Z\rangle$. Compute the variance of each estimator at $\theta = \pi/2$ for $N = 100$ shots.
2. Show that $\mathrm{Var}[\widehat{\langle P\rangle}] = (1 - \langle P\rangle^2)/N$. Where in the derivation does the boundedness of the eigenvalues to ± 1 enter?
3. (VQA) The naive cost-estimator variance bound is $\sum_j h_j^2/N$. For a 4-qubit H_2 Hamiltonian with 15 Pauli terms, estimate the shot count per term needed to achieve millihartree accuracy in the cost estimator.

Research Extensions

- **Classical shadows** — measure random unitaries, estimate many observables at once; logarithmic shot scaling in some regimes.
- **Adaptive shot allocation** — dynamically reallocate shots between Pauli terms based on observed variance.
- **Importance-weighted sampling** — query terms with larger $|h_j|$ more often.
- **Bayesian estimators** — use prior knowledge about $\langle P_j\rangle$ to improve early-iteration accuracy.

Cross-links to Other Courses

- **Quantum Measurement** — Born rule, projective measurements, POVMs.
- **Statistics Module 3** — sample mean, variance of estimators, central limit theorem.
- **Module 8** (measurement grouping) — the immediate next step after this node.
- **Module 9** (hardware noise models) — what gets layered on top in practice.

Variational Quantum Algorithms · Module 2 — Parameterized Circuits · Node 2.7
Concepts: born-rule, expectation-value, tensor-product, variance-and-covariance
Prerequisites: 2.5, 2.6
Enables: 2.8

hbar.university — own.your.cognition