Shot Noise as Cost Stochasticity

Variational Quantum Algorithms · Module 3 — Cost Landscapes

Node 3.7

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Cost Landscapes **Node:** 3.7

So far in Module 3, the cost function has been treated as a *deterministic* scalar function on parameter space. This node introduces the **stochastic** view: in practice the optimizer never sees $C(\boldsymbol{\theta})$ — it sees a noisy estimate $\widehat{C}(\boldsymbol{\theta})$ corrupted by shot noise.

The relationship between C and $\widehat{C}$ is the relationship between **what the landscape really is** and **what the optimizer thinks it is**, and a lot of VQA practice consists of managing the gap between the two.

This node lays out the structure of that gap.

The Two Cost Functions

Let's be explicit.

True cost:

$$C(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle.$$

A deterministic function of $\boldsymbol{\theta}$. What you'd compute from a perfect statevector simulation.

Estimated cost:

$$\widehat{C}(\boldsymbol{\theta}) = \sum_j h_j \widehat{\langle P_j \rangle}, \quad \widehat{\langle P_j \rangle} = \frac{1}{N_j} \sum_{s=1}^{N_j} \hat{e}_j^{(s)}.$$

A random variable. What the optimizer actually receives. $\mathbb{E}[\widehat{C}(\boldsymbol{\theta})] = C(\boldsymbol{\theta})$ (unbiased estimator, in the absence of hardware noise).

The optimizer's job is to find $\arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$, but its only access to C is through repeated samples of $\widehat{C}$. This is a stochastic optimization problem in the textbook sense.

The Variance Structure

The variance of $\widehat{C}(\boldsymbol{\theta})$ is

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

This formula has several important features:

1. Inverse in shot count. Doubling N_j halves the variance contribution from term j.

2. Proportional to coefficient squared. Terms with large $|h_j|$ contribute more variance per shot.

3. State-dependent. The factor $(1 - \langle P_j \rangle^2)$ means that variance is *minimized* when $\langle P_j \rangle = \pm 1$ (the state is an eigenstate of P_j) and *maximized* when $\langle P_j \rangle = 0$. **The variance depends on $\boldsymbol{\theta}$.**

The third point is the most important: **the noise on $\widehat{C}$ is not uniform across parameter space**. Some regions are intrinsically noisier than others, and the optimizer experiences a *changing noise profile* as it moves around.

For chemistry Hamiltonians where the dominant Pauli terms have $h_j \sim 1$, the typical variance contribution per term is order 1 over N shots, and total variance is $\sum_j h_j^2/N$ — easily 0.01-1 for realistic settings. That's a per-evaluation standard deviation of 0.1-1 hartree, which is large compared to chemical accuracy (~ 0.001 hartree).

Where The Variance Hides The Structure

When the variance of $\widehat{C}$ is large compared to the *local variation* of C, the optimizer cannot distinguish "small motion in $\boldsymbol{\theta}$" from "no motion in $\boldsymbol{\theta}$."

Define the **signal-to-noise ratio** at a point:

$$\mathrm{SNR}(\boldsymbol{\theta}) = \frac{\|\nabla C(\boldsymbol{\theta})\|}{\sqrt{\mathrm{Var}[\widehat{C}(\boldsymbol{\theta})]}}.$$

When SNR is large, the optimizer sees the gradient clearly and can descend. When SNR is near 1, the optimizer's view is dominated by noise and progress is slow. When SNR is much less than 1, the optimizer is essentially doing a random walk.

The dangerous regime is **near critical points and on plateaus**, where the gradient is small but the variance is not. This is exactly where you most want to make small careful steps, and exactly where you can't tell whether your steps are doing anything.

This is the **shot noise floor** that limits VQA optimization in practice. It is not addressed by being smarter; it requires either more shots, or a smaller variance contribution at the relevant points, or better signal extraction.

How Shot Noise Differs From Other Optimization Noise

Shot noise has properties that make it different from generic stochastic optimization noise:

1. State-dependent variance. Unlike the constant Gaussian noise often assumed in stochastic optimization theory, shot noise variance depends on the parameter point. Standard convergence analyses (e.g., Robbins-Monro) assume bounded variance and may not directly apply.

2. Bounded estimator outputs. Each shot gives a value in $[-1, +1]$ for each Pauli, so $\widehat{C}$ has bounded support. This is mostly a feature — sub-Gaussian concentration applies — but it means certain heavy-tailed-noise techniques are not relevant.

3. Correlated across the gradient. When you compute $\nabla\widehat{C}$ by parameter-shift, the same shots are used for shifted evaluations within a single component but independent shots are used across components. This produces a specific (block-diagonal) covariance structure that some optimizers can exploit and others can't.

4. Adaptive in principle. You can choose to spend more shots in some regions of parameter space than others — variance is not a fixed property of the problem, it is a budget allocation. Module 5's "shot-adaptive weighting" technique exploits this.

5. Time-stationary. On simulated hardware, shot noise is i.i.d. across iterations. On real hardware, it can drift due to calibration changes. Most analysis assumes the i.i.d. case.

These features are why VQA optimization is its own subfield rather than a special case of generic stochastic optimization.

Cost-Side vs Optimizer-Side Mitigation

The shot noise problem can be attacked from either side:

Cost-side: reduce the variance of $\widehat{C}$ at fixed shot budget.

- Better measurement grouping (node 2.8) → lower per-shot variance.
- Variance-weighted shot allocation → put shots where they matter most.
- Local cost functions → replace high-variance global observables with low-variance local ones.
- Importance sampling — query terms with large $|h_j|$ more often.

Optimizer-side: make the optimizer robust to whatever noise is there.

- Larger batch sizes (more shots per evaluation) → lower per-evaluation variance.
- Momentum / averaging methods (Adam, EMA gradients) → average over noise across iterations.
- Trust region methods — only commit to a step when the noise is small enough to be confident.
- Population methods (HOPSO) → collectively average over many points.

The thesis argues that *both* sides should be considered, and that the regularization techniques in Module 5 are best understood as the dynamics-side counterpart to the cost-side techniques here.

What Shot Noise Looks Like To The Optimizer

A toy mental model: imagine the true cost surface, then add a random Gaussian field of standard deviation $\sigma(\boldsymbol{\theta})$ on top of it. The Gaussian field is independent at each evaluation but has the same variance structure across evaluations (assuming i.i.d. noise).

What the optimizer sees at any single iteration is: - The true gradient direction, plus - A random gradient component proportional to the local noise level, plus - Anything from the optimizer's own update rule (momentum, etc.).

If the random component dominates the true gradient, the optimizer is essentially walking randomly. If the true gradient dominates, the optimizer is doing approximate gradient descent.

The transition between these regimes happens around SNR = 1, and the location of that transition in parameter space depends on the local landscape and the shot budget. **The optimizer crosses in and out of "useful" and "useless" regions during a single optimization run.**

Structural Role

This is the **stochasticity** node of Module 3 and the bridge to gradient computation (3.8) and to Module 5 (where the regularization framework is built around managing this exact problem).

It is also one of the cleanest places to see the "stochastic objective generator" abstraction (1.8) at work: the optimizer sees a noisy scalar oracle whose properties are determined by upstream physics. The shot noise is the dominant source of that noise on simulated hardware.

Relations to Other Concepts

- **Stochastic optimization theory** — Robbins-Monro, Kiefer-Wolfowitz, SGD, SPSA.
- **Sub-Gaussian random variables** — the formal class to which bounded shot noise belongs.
- **Variance reduction techniques** — control variates, stratified sampling, importance sampling.
- **Bayesian optimization** — explicitly models the stochastic oracle and uses Bayesian inference to decide where to query.
- **Robust optimization** — optimizing in the presence of bounded but unknown perturbations.

Worked Example — Variance At Three Parameter Points

For $|\psi(\theta)\rangle = R_Y(\theta)|0\rangle$ and $H = Z + 0.5X$:

$$C(\theta) = \cos(\theta) + 0.5\sin(\theta).$$

At $\theta = 0$ (state $|0\rangle$): $\langle Z\rangle = 1$, $\langle X\rangle = 0$. Cost = 1. Variance contributions: $(1-1) + 0.25(1-0) = 0.25$. Standard deviation $= 0.5/\sqrt{N}$.

At $\theta = \pi$ (state $|1\rangle$): $\langle Z\rangle = -1$, $\langle X\rangle = 0$. Cost = -1. Variance: $0 + 0.25 = 0.25$. Same as above.

At $\theta = \pi/2$ (state $|+\rangle$): $\langle Z\rangle = 0$, $\langle X\rangle = 1$. Cost = 0.5. Variance: $(1-0) + 0.25(1-1) = 1$. Standard deviation $= 1/\sqrt{N}$.

So the noise on $\widehat{C}$ varies by factor of 2 across parameter space. At eigenstate-like points (where one of the Pauli terms is saturated), variance is lower. At superposition points, variance is higher.

For an optimizer trying to detect a small gradient near $\theta = \pi/2$, the noise floor is *higher* there than near the minimum, even though the gradient is *also* higher. The SNR may or may not be favorable depending on the exact geometry.

This is the kind of effect that gets averaged away in textbook analyses but matters operationally.

Visualization

Pending. A 1D plot of $C(\theta)$ over $[-\pi, \pi]$, with a noise band of width $\sigma(\theta)$ around the curve, showing variable thickness. Annotation: "noise is wider at superposition points, narrower near eigenstates."

Exercises

1. For $H = Z_0Z_1$ on 2 qubits and a state $|\psi\rangle = \cos(\theta/2)|00\rangle + \sin(\theta/2)|11\rangle$, compute $\langle ZZ\rangle(\theta)$ and $\mathrm{Var}[\langle ZZ\rangle](\theta)$. Where is the variance maximized?
2. Show that if a cost function is the expectation value of a single observable bounded in $[-1, +1]$, then $\mathrm{Var}[\widehat{C}] \leq 1/N$ where N is the shot count, regardless of the state.
3. (VQA) For an optimizer running with adaptive shot allocation, design a heuristic that increases shot count when SNR is low. What metric would you use to decide?

Research Extensions

- **Variance-aware shot allocation** — dynamically increasing shots in high-noise regions.
- **Importance-weighted estimators** — biasing measurements toward terms with large coefficients.
- **Variance penalization (Module 5.7)** — regularizing the cost to prefer low-variance regions.

- **Bandit-style stopping rules** — terminating evaluation when confidence intervals are tight enough.

Cross-links to Other Courses

- **Statistics** — sample variance, central limit theorem, sub-Gaussian concentration.
- **Stochastic Optimization** — Robbins-Monro, SPSA, Polyak averaging.
- **Module 5 (Regularization)** — the dynamics-side approach to managing this noise.
- **Module 9 (Hardware Noise Models)** — noise sources beyond shot noise.

Variational Quantum Algorithms · Module 3 — Cost Landscapes · Node 3.7
Concepts: stochastic-optimization, variance-and-covariance, random-variables, expectation-value
Prerequisites: 2.7, 3.5
Enables: 3.8

hbar.university — own.your.cognition