The Cosine Schedule and Two-Stage Protocol

Variational Quantum Algorithms · Module 5 — Regularization

Node 5.8

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Regularization **Node:** 5.8

This node introduces the **two-stage protocol** that operationalizes the dynamical-modification view (node 5.5) using a cosine-annealed regularization schedule. It is the most concrete recommendation in Module 5: a specific recipe with specific hyperparameters that has been used in our own experiments. It depends on 5.5 (the framing), 5.6 (parameter-space techniques), and 5.7 (objective-space techniques).

The Idea in One Sentence

> Apply regularization aggressively early to stabilize trajectories, then anneal it to zero so the final iterates converge to the original (unregularized) optimum.

This is the obvious move once you accept the dynamical-modification view of node 5.5: regularization is for the path, not for the destination. If you keep regularization on at convergence you have changed the problem; if you anneal it away you have only changed the route.

The Schedule

Let T be the total iteration budget. Define a cosine schedule on the regularization weight:

$$\lambda_k = \frac{\lambda_0}{2}\left(1 + \cos\frac{\pi k}{T}\right), \qquad k = 0, 1, \ldots, T.$$

At $k = 0$, $\lambda_k = \lambda_0$. At $k = T$, $\lambda_k = 0$. The decay is smooth and monotone with a flat shoulder at the start (so early training experiences nearly the full regularization) and a flat tail at the end (so the optimizer has time to settle into the unregularized optimum).

Cosine schedules are popular in deep learning for the **learning rate**. Here we apply the same shape to the **regularization weight**, for analogous reasons: early phase wants exploration with stabilization, late phase wants fine-tuning without distortion.

The Two-Stage Protocol

The protocol splits the optimizer's run into two phases:

Stage A — Stabilized exploration. For $k \in [0, T_A]$, run the optimizer with the regularization schedule active. Use a relatively small per-iteration budget (`maxiter`$_A$ inner steps per outer iteration if the optimizer is hierarchical). The goal here is **not** to reach the minimum. The goal is to reach a stable region of parameter space that is robust to shot noise.

Stage B — Unregularized refinement. For $k \in [T_A, T]$, the schedule has effectively decayed; run the optimizer with regularization disabled (or near-zero). Use a longer per-iteration budget ($\texttt{maxiter}_B$). The goal here **is** to reach the minimum. Because Stage A delivered a good initialization, the unregularized phase no longer suffers the instabilities that motivated regularization in the first place.

A Concrete Configuration (Chernyak D30)

The configuration used in our own VQE experiments on a depth-30 problem instance:

Component	Setting
Ansatz	`TwoLocal(rotation_blocks=['rz','ry'], reps=4)`
Parameters	40
Optimizer	HOPSO (see Module 6)
Stage A iterations T_A	15
Stage A inner budget $\texttt{maxiter}_A$	15
Stage B inner budget $\texttt{maxiter}_B$	10
Schedule	cosine on regularization weight
λ_0	tuned per instance, typically small

These numbers are not magic. They were chosen empirically on a specific class of Hamiltonians and they reflect a tradeoff between *time spent stabilizing* and *time spent converging*. They are reported here as a worked example, not as a universal recommendation. Node 5.9 (empirical behavior) discusses how performance depends on these knobs.

Why Two Stages and Not One?

A natural alternative is to keep regularization on continuously, anneal slowly, and run a single-phase optimizer. The two-stage version is operationally cleaner for three reasons:

1. **Hyperparameter independence.** The Stage A budget can be tuned for stabilization, the Stage B budget for refinement. They optimize different quantities and you do not have to find a single setting that does both jobs at once.
2. **Logging clarity.** It is easier to read a training curve when the role of each iteration is explicit. "Why did the loss spike?" is answered differently in Stage A (stabilization phase, expected) and Stage B (refinement phase, suspicious).
3. **Optimizer compatibility.** Some optimizers (HOPSO included) have internal state — particle velocities, momentum buffers — that benefit from a clean reset between phases. The two-stage protocol gives you a natural moment to reset.

Structural Role

This node is the **operational climax** of Module 5. Everything in 5.1–5.7 leads here: - 5.1–5.4 establish the standard view and its limits. - 5.5 reframes regularization as dynamical modification. - 5.6 and 5.7 enumerate the levers. - 5.8 picks one specific lever (annealed parameter-space penalty) and shows how to schedule it.

Node 5.9 (empirical behavior under shot noise) then asks the question this node provokes: *does the protocol actually work?*

Relations to Other Concepts

- **Cosine annealing for learning rate** — same shape, different quantity. Loshchilov-Hutter, SGDR.
- **Curriculum learning** — staging the optimization problem from easier to harder.
- **Trust-region adaptation** — Stage B is essentially a relaxed trust region after Stage A has positioned the iterate.
- **Simulated annealing** — temperature schedule analogy; here λ plays the role of an inverse temperature.

Visualization

Pending. Diagram: a training curve with two color-coded segments. Left half: Stage A, jagged but bounded, with the cosine λ_k curve overlaid. Right half: Stage B, smooth descent to a lower asymptote. Annotate T_A, $\lambda_0 \to 0$, and the final unregularized minimum.

Exercises

1. Plot the cosine schedule $\lambda_k = \frac{\lambda_0}{2}(1 + \cos(\pi k/T))$ and verify that $\lambda_0 = \lambda_0$ and $\lambda_T = 0$. What is the iteration at which $\lambda_k = \lambda_0/2$?
2. Suppose Stage A is too short. Predict qualitatively what the training curve looks like and which failure mode dominates.
3. (VQA) Why is it inappropriate to evaluate the *final* energy of a Stage A run as if it were the answer? What metric should you report instead from Stage A?

Research Extensions

- **Adaptive T_A** — switching from Stage A to Stage B when a stability criterion is met instead of at a fixed iteration.
- **Schedule shapes beyond cosine** — linear, polynomial, step. Empirically, the differences are small once the total integral is matched.
- **Multi-stage protocols** — three or more phases for hierarchical optimization problems.

Cross-links to Other Courses

- **Optimization Dynamics (Module 6)** — interaction with HOPSO's intrinsic damping.
- **Hardware Noise Models (Module 9)** — Stage A's stabilization is most beneficial under high noise.
- **Math-Intuition** — the cosine schedule as a smooth interpolant between two regimes.

Variational Quantum Algorithms · Module 5 — Regularization · Node 5.8
Concepts: regularization, convergence, stochastic-optimization
Prerequisites: 5.5, 5.6, 5.7
Enables: 5.9

hbar.university — own.your.cognition