Objective-Space Regularization Techniques

Variational Quantum Algorithms · Module 5 — Regularization

Node 5.7

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Regularization **Node:** 5.7

This node catalogs the **objective-space** family of regularization techniques: methods that modify the cost signal $C(\boldsymbol{\theta})$ **before** it reaches the optimizer, without touching the parameter update rule. It is the sibling of node 5.6 and the second operational half of the dynamical-modification view from node 5.5.

The Common Mechanism

The optimizer sees not the raw cost $C(\boldsymbol{\theta}_k)$ but a transformed signal

$$\widetilde{C}_k = \mathcal{F}\big(C(\boldsymbol{\theta}_k), C(\boldsymbol{\theta}_{k-1}), \ldots; \lambda\big).$$

The transformation $\mathcal{F}$ might smooth, weight, or augment the cost signal. The optimizer's update rule is **unchanged** — it still consumes a number called "the cost" — but the number it consumes has been filtered.

This contrast with node 5.6 is the entire point: same optimizer, different feedback.

Technique 1 — Temporal Smoothing

Replace the instantaneous cost with a moving average or exponentially-weighted average:

$$\widetilde{C}_k = (1-\beta)\widetilde{C}_{k-1} + \beta\, C(\boldsymbol{\theta}_k).$$

This introduces inertia into the feedback loop. Transient shot-noise spikes are damped before they can cause large parameter excursions. The price is delayed responsiveness: a genuine improvement in C takes a few iterations to show up in $\widetilde{C}$.

In VQA this is one of the cheapest stabilizers available — no extra circuit evaluations, just a running average — and it directly attacks the dominant noise source on NISQ hardware.

Technique 2 — Variance Penalization

Penalize cost evaluations with high statistical uncertainty:

$$\widetilde{C}(\boldsymbol{\theta}) = C(\boldsymbol{\theta}) + \mu\,\mathrm{Var}\big[\hat{C}(\boldsymbol{\theta})\big].$$

This biases the optimizer toward regions where the cost can be **trusted**, not just where it is small. In VQA the variance of a Pauli expectation estimate depends on $\langle\psi(\boldsymbol{\theta})|H^2|\psi(\boldsymbol{\theta})\rangle - \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle^2$, so this is a real, measurable quantity — not an abstract penalty.

As with all objective-space techniques in VQA, the penalty is typically scheduled to vanish near convergence so the location of the optimum is preserved.

Technique 3 — Shot-Adaptive Weighting

Cost evaluations carry different reliabilities depending on how many measurement shots were used to estimate them. Weight each evaluation by its precision:

$$\widetilde{C}_k = \frac{\sum_j w_j C_j}{\sum_j w_j}, \qquad w_j \propto \frac{1}{\sigma_j^2}.$$

This lets the optimizer use cheap, high-variance estimates exploratorily while still anchoring on rare high-shot evaluations near promising regions.

The resource-allocation question — *how to spend a finite shot budget across iterations* — is where this technique shades into the larger topic of measurement-optimizer co-design (see Module 10).

Technique 4 — Auxiliary Penalty on Cost-Change Magnitude

Penalize rapid swings in successive cost values:

$$\widetilde{C}_k = C(\boldsymbol{\theta}_k) + \nu\left(C(\boldsymbol{\theta}_k) - C(\boldsymbol{\theta}_{k-1})\right)^2.$$

This discourages oscillatory behavior driven by noise. Like temporal smoothing, it adds inertia, but here the inertia is on the **cost signal's volatility** rather than its level.

Structural Role

Objective-space regularization is the natural intervention point when **the optimizer is fine but the signal is noisy**. In VQA this is the typical failure mode: gradient methods, HOPSO, COBYLA — they all behave reasonably given a clean cost signal. The instability comes from shot noise corrupting the feedback.

This node, together with 5.6, gives the practitioner the full menu of dynamical-modification options. Node 5.8 then shows how to **schedule** these techniques over training so they vanish near convergence.

Relations to Other Concepts

- **Polyak averaging** — averaging iterates is a parameter-space analogue of cost averaging.
- **Kalman filtering** — temporal smoothing as a special case of state estimation under noise.
- **CMA-ES** — uses a form of variance-aware weighting in its sample selection.
- **Importance-weighted gradients** — the precision-weighting trick generalized.

Worked Example — Temporal smoothing as a low-pass filter

Treat the cost iterates $\{C(\boldsymbol{\theta}_k)\}$ as a discrete-time signal. The exponential moving average

$$\widetilde{C}_k = (1-\beta)\widetilde{C}_{k-1} + \beta C_k$$

has the transfer function

$$H(z) = \frac{\beta}{1-(1-\beta)z^{-1}},$$

a first-order low-pass filter with cutoff $\omega_c \approx \beta$ (in radians per iteration). Smaller β means heavier smoothing and slower response. The effective cost landscape seen by the optimizer is the original landscape convolved with this filter — high-frequency shot-noise components are suppressed; low-frequency descent directions pass through.

Visualization

Pending. Diagram: cost time series with high-frequency noise overlaid on a slow descent. Two traces: raw C_k (jagged) and smoothed $\widetilde{C}_k$ (clean). Annotate where the optimizer would have fired a destabilizing update on the raw signal.

Exercises

1. Show that exponential smoothing with parameter β is equivalent to a first-order IIR low-pass filter, and derive its 3 dB cutoff frequency.
2. Suppose the variance of your cost estimate scales as $1/N$ where N is the shot count. Design a shot-adaptive weighting scheme that minimizes the variance of $\widetilde{C}_k$ given a total shot budget $N_{\text{tot}} = \sum_j N_j$. (Hint: Lagrange multipliers.)
3. (VQA) Why does temporal smoothing interact badly with population-based optimizers like vanilla PSO? (Hint: think about how attractor positions get updated.)

Research Extensions

- **Adaptive smoothing** — adjusting β based on observed cost variance.
- **Bayesian optimization with noisy oracles** — explicit uncertainty modeling as objective-space regularization.
- **Joint shot-and-step adaptation** — co-designing the optimizer step size and the shot budget per iteration.

Cross-links to Other Courses

- **Signal Processing** — IIR filters, low-pass design, group delay.
- **Statistics Module 6** — Kalman filters and state estimation.
- **Optimization Dynamics (Module 6)** — the input side of the gradient-flow analysis.

Concepts: regularization, cost-function, stochastic-optimization
Prerequisites: 5.5
Enables: 5.8, 5.9

hbar.university — own.your.cognition