Zero-Noise Extrapolation

Variational Quantum Algorithms · Module 9 — Hardware Noise Models

Node 9.5

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Hardware Noise Models **Node:** 9.5

Zero-noise extrapolation (ZNE) is the single most widely deployed error mitigation technique for VQAs. It is conceptually simple, needs no detailed noise model, works with any observable, and can be applied to almost any circuit with modest overhead. The basic idea:

1. Run the circuit at several **artificially amplified** noise levels.
2. Fit the expectation value as a function of noise strength.
3. Extrapolate back to zero noise.

The key insight is that you can deliberately make your circuits noisier in controlled amounts, and from the resulting data points you can infer where the line (or curve) would cross at zero noise. Under the right conditions, this gives a substantially better estimate than just running the circuit at nominal noise.

ZNE was introduced by Li-Benjamin (2017) and Temme-Bravyi-Gambetta (2017) more or less simultaneously. It has since become the default mitigation technique in Qiskit, Mitiq (a dedicated mitigation library), PennyLane, and most VQA benchmarking studies.

This node covers the details: noise amplification methods, fit choices, error analysis, practical implementation, and its integration with VQA optimization.

The Basic Idea, Formally

Let $\langle O\rangle(\lambda)$ be the expectation value of observable O when the circuit's noise is amplified by factor $\lambda \geq 1$. The nominal circuit corresponds to $\lambda = 1$. By construction, $\lambda = 0$ corresponds to *no noise*.

Assumption. The function $\lambda \mapsto \langle O\rangle(\lambda)$ is smooth and can be extrapolated from points $\lambda_1 = 1, \lambda_2, \lambda_3, \ldots$ back to $\lambda = 0$.

Procedure.

1. Choose a set of amplification levels $\{\lambda_i\}$ (e.g., $\{1, 1.5, 2, 2.5\}$).
2. For each λ_i, amplify the circuit's noise by that factor and run N_{shots} shots to get $\hat{O}(\lambda_i)$.
3. Fit a model $f(\lambda)$ to the data points $\{(\lambda_i, \hat{O}(\lambda_i))\}$.
4. Report the extrapolated value $\hat{O}_{\text{ZNE}} = f(0)$ as the mitigated estimate.

The critical choices are: **how to amplify the noise** (without changing the logic of the circuit) and **what functional form to fit**.

Noise Amplification Methods

Three main strategies exist for increasing the effective noise without changing the ideal operation of the circuit.

1. Unitary folding

Replace each gate U with $U \cdot U^\dagger \cdot U$, which is logically equivalent to U but has three times the gate count (and hence three times the noise).

More generally, folding by factor $\lambda = 2k + 1$ replaces U with $U \cdot (U^\dagger U)^k$. **Global folding** applies this to the entire circuit U_{circ}; **local folding** applies it to individual gates.

Pros: noise scaling is discrete ($\lambda = 1, 3, 5, \ldots$) but can be made continuous by folding a random subset of gates. Works in circuit-level simulators. Widely supported.

Cons: the assumption that noise scales linearly with gate count is only approximate (different gates have different noise, and noise doesn't always compose linearly).

Reference. Giurgica-Tiron et al. (2020) introduced the modern unitary folding framework.

2. Pulse stretching

At the pulse level, lengthen each gate's control pulses by a factor $\lambda \geq 1$. The gate does the same ideal operation but takes longer, during which more decoherence accumulates.

Pros: the noise amplification is *physical* rather than synthetic — you're actually running the hardware in a noisier regime. Fewer assumptions.

Cons: requires pulse-level hardware access (not available via simple circuit APIs), the stretching may not preserve gate fidelity exactly (coherent errors can get worse rather than just scaling), and λ is bounded by the coherence time (you can't stretch indefinitely).

Reference. Kandala et al. (2019) used pulse stretching for ZNE in the IBM hardware stack.

3. Identity insertion

Insert identity operations (physically realized as $UU^\dagger$ for some U) at various points in the circuit, adding noise without changing the ideal logic.

Similar to unitary folding in effect but more flexible about *where* the extra noise goes.

Pros: can target specific gates or qubits for amplification.

Cons: the specific choice of U matters (different U's have different noise profiles), so the amplification is not cleanly parameterized.

In practice. Unitary folding is the default for circuit-level ZNE; pulse stretching is the default when pulse-level access is available; identity insertion is used for specialized studies.

Extrapolation Fits

With data $\{(\lambda_i, \hat{O}_i)\}$, several fit choices exist.

Linear fit

$$f(\lambda) = a\lambda + b.$$

Extrapolated value: $f(0) = b$.

Simplest, robust, appropriate when the noise is small and the dependence is approximately linear.

Pros: stable, easy to implement, small variance.

Cons: can have significant bias when the true dependence is nonlinear.

Polynomial fit

$$f(\lambda) = \sum_{k=0}^{d} c_k \lambda^k.$$

Degree d fit; extrapolation is $f(0) = c_0$. Richardson extrapolation is a special case.

Pros: more flexible.

Cons: higher-degree polynomials overfit and amplify statistical noise in the extrapolation. Numerical instability for large d.

Typical choice: $d = 2$ or $d = 3$, no higher.

Exponential fit

$$f(\lambda) = Ae^{-\Gamma\lambda} + B.$$

Extrapolated value: $f(0) = A + B$.

The exponential form is motivated by the actual dependence of Pauli expectation values on depolarizing noise: $\langle P\rangle(\lambda) \approx \langle P\rangle_{\text{ideal}} \cdot (1 - 4p/3)^{L\lambda}$, which is exponential in λ.

Pros: matches the true depolarizing dependence.

Cons: the fit is nonlinear and can fail to converge with sparse data. Requires at least 3 data points and often 4+.

Typical choice: when $\geq$4 data points are available and noise is known to be approximately depolarizing.

Richardson extrapolation

A specific polynomial extrapolation that eliminates leading-order error terms one at a time. For example, with $\lambda = 1, 2$:

$$f(0) \approx 2\hat{O}(1) - \hat{O}(2).$$

With $\lambda = 1, 2, 3$: a weighted sum of the three values that cancels both linear and quadratic terms.

Pros: closed-form, known error bounds, well-studied in numerical analysis.

Cons: assumes specific polynomial structure, amplifies variance at higher orders.

Error Analysis

Two sources of error affect ZNE:

1. Statistical error. Each $\hat{O}(\lambda_i)$ has shot noise of order $1/\sqrt{N_{\text{shots}}}$. The extrapolation amplifies this noise. For linear extrapolation with two points $\lambda = 1, 2$, the extrapolated variance is:

$$\text{Var}[\hat{O}_{\text{ZNE}}] = \text{Var}[2\hat{O}(1) - \hat{O}(2)] = 4\sigma^2 + \sigma^2 = 5\sigma^2,$$

so the variance is 5x the single-run variance. For higher-order fits, the variance amplification is larger (up to $\sim 30\sigma^2$ for Richardson with $\lambda = 1, 2, 3$).

2. Extrapolation bias. If the true dependence is not the model you fit, extrapolation introduces a bias. For a linear fit applied to data that is actually exponential, the bias is $\sim$ (curvature) $\cdot \lambda_{\text{min}}^2$.

Tradeoff. Higher-order fits reduce bias but increase variance. The optimal choice depends on the circuit, noise level, and shot budget.

Rule of thumb. For moderate noise and shallow circuits, use linear fits. For moderate-to-deep circuits, use exponential. For high-precision work, use several fit types and compare — if they agree, trust them; if they disagree, the extrapolation is unreliable.

Pitfalls Of ZNE

Several situations where ZNE fails or misleads:

1. Non-monotonic dependence. If the noisy expectation value is not monotonic in λ (e.g., because of coherent errors that build up non-linearly), the extrapolation has no fixed point to converge to. Fit functions can give nonsense values.

2. Noise that's not a "factor". ZNE assumes noise can be amplified *uniformly*. For heterogeneous noise (different gates have very different error rates), this assumption breaks down. The effective λ is ambiguous.

3. Saturation at small λ. If the nominal noise is already close to saturation (noise level so high that expectation values are near zero), you cannot extrapolate linearly — the curve is in its flat region and the fit is ill-conditioned.

4. Cost landscape distortion. Applying ZNE to every cost evaluation in a VQA optimization slows convergence (because shots are 3-10x more expensive) *and* introduces correlated errors across iterations. The optimizer sees a slightly noisier and slightly warped landscape.

5. The expensive folded circuits. Folded circuits have more gates, so they take longer to run *and* they're more error-prone. At high λ, the amplified circuit may be so noisy that the data point is useless.

Mitigating the pitfalls. Use multiple λ values (at least 3), try multiple fit functions, check consistency, and validate on benchmark problems where the ideal answer is known (Clifford circuits work well for this).

ZNE In VQA Optimization

A subtle question: when should ZNE be applied *during* VQA optimization?

Option A — Every iteration. Apply ZNE at every cost evaluation. Guarantees the optimizer sees the mitigated cost, but the shot overhead is 3-10x per iteration.

Option B — Only at the final evaluation. Optimize with raw (unmitigated) cost, then apply ZNE once at the end to estimate the true cost at the converged point. Saves shots during training but relies on the raw cost having the same minimum as the mitigated cost.

Option C — Periodic mitigation. Apply ZNE every k iterations to check convergence, use raw cost otherwise. Compromise between A and B.

The tradeoff. Option B is correct *if* noise is depolarizing (since depolarizing preserves the minimum location — see node 9.2). For more structured noise, the raw minimum may differ from the mitigated minimum, and Option A is needed.

Current practice. Most 2026 VQA experiments use Option B or C for compute efficiency, with Option A reserved for small-scale precision studies.

ZNE As A Basic Building Block

ZNE is the "default" mitigation in most VQA libraries and is the first technique to reach for. Its main virtues:

1. **Model-free.** Doesn't require knowledge of the noise channel.
2. **Universal.** Works for any observable, any circuit, any noise type.
3. **Cheap.** Overhead is 3-10x in shots, which is manageable.
4. **Stackable.** Can be combined with readout mitigation, symmetry verification, DD without conflict.
5. **Transparent.** The extrapolation curve gives a visual diagnostic — if the curve looks wrong, you know ZNE is unreliable.

The main virtues of *alternatives* (CDR, PEC, virtual distillation) are corrections to specific ZNE failure modes. ZNE is the baseline; other techniques are improvements for specific conditions.

Structural Role

This node is the **detailed treatment** of the most widely used mitigation technique, expanding on the overview in node 9.4. It covers noise amplification methods, fit choices, error analysis, pitfalls, and VQA integration.

It is the reference node for ZNE in the course, and the practical companion to node 9.4's survey view.

Relations to Other Concepts

- **Li, Benjamin (2017)** — ZNE origin.
- **Temme, Bravyi, Gambetta (2017)** — ZNE + PEC origin.
- **Kandala et al. (2019)** — hardware ZNE demonstration with pulse stretching.
- **Giurgica-Tiron et al. (2020)** — unitary folding framework and Mitiq library.
- **LaRose et al. (2022)** — Mitiq: software package for error mitigation.
- **Richardson extrapolation (numerical analysis classic)** — the parent technique.
- **Module 9.2 (Depolarizing noise)** — why the minimum is preserved under depolarizing.
- **Module 9.4 (Mitigation techniques)** — the broader context.

Worked Example — ZNE On A 10-Qubit QAOA

Setup. A 10-qubit QAOA at depth $p = 3$, two-qubit gate fidelity 99.5%, shot budget 10^5 per cost evaluation. The cost is $\langle H_C \rangle = -0.53$ (noiseless, known from small simulation) and -0.32 (noisy, directly measured).

Step 1 — Choose amplification levels. $\lambda \in \{1, 2, 3\}$ via unitary folding.

Step 2 — Run circuits. - $\lambda = 1$: fold factor 1 (nominal), $\hat{O}_1 = -0.32$. - $\lambda = 2$: fold each gate to $UU^\dagger U$, 3 gate count, $\hat{O}_2 = -0.19$. - $\lambda = 3$: fold each gate twice, 5 gate count, $\hat{O}_3 = -0.11$.

Each shot budget 10^5, so total 3×10^5 shots.

Step 3 — Fit.

Linear fit: $f(\lambda) = -0.41 + 0.105\lambda$. $f(0) = -0.41$.

Exponential fit: $f(\lambda) = -0.55 \cdot e^{-0.42\lambda} + 0.01$. $f(0) = -0.54$.

Richardson (with $\lambda = 1, 2$*):* $f(0) = 2 \cdot (-0.32) - (-0.19) = -0.45$.

Step 4 — Compare to the truth. The noiseless value is -0.53. - Linear: -0.41, error 0.12. - Exponential: -0.54, error 0.01. **Best.** - Richardson: -0.45, error 0.08. - Unmitigated: -0.32, error 0.21.

Observation. Exponential is the best fit for this depolarizing-dominated regime. Linear is significantly biased because the underlying decay is not linear. Richardson is between the two.

Shot cost. 3x (three amplification levels $\times$ 10^5 shots each).

Variance amplification. Roughly 5x for linear, more for Richardson, less for exponential (because the fit is "softer").

Conclusion. Exponential ZNE gives a 20x bias reduction at 3x shot cost. Good tradeoff.

Visualization

Pending. A plot of $\langle H_C \rangle$ vs λ showing three data points (at $\lambda = 1, 2, 3$), three fit curves (linear, exponential, Richardson), and the true value at $\lambda = 0$. The exponential curve should hit the true value closely; the linear should miss. Caption: "ZNE in action: fit choice matters."

Exercises

1. For a single-qubit depolarizing channel at rate p, show that the expectation value of X under λ-folded noise is $(1 - 4p/3)^{2\lambda-1} \cdot \langle X \rangle_{\text{ideal}}$. Use this to derive the exact exponential fit formula.
2. Prove that linear ZNE is unbiased if the true $\langle O \rangle(\lambda)$ is linear in λ, and biased by $\mathcal{O}(\lambda^2 \cdot |O''|)$ if it is quadratic.
3. (VQA) For a 12-qubit VQE with depth 60 and two-qubit gate fidelity 99.3%, compare linear, exponential, and Richardson ZNE using simulated data. Which gives the smallest total error (bias + statistical)?

Research Extensions

- **Adaptive λ-selection** — automatically choosing the amplification levels based on the observed curvature.
- **Learning-based extrapolation** — using neural networks to extrapolate rather than parametric fits.
- **Joint cost/gradient ZNE** — mitigating both the cost and its gradient simultaneously with a shared shot budget.
- **Exponential ZNE with variance reduction** — variance reduction techniques specific to the exponential fit.
- **ZNE in the deep-circuit regime** — adapting ZNE for circuits that are past the saturation depth.

Cross-links to Other Courses

- **Numerical Analysis** — Richardson extrapolation, polynomial interpolation.
- **Statistical Estimation** — bias-variance tradeoff, extrapolation risk.
- **Module 9.4 (Mitigation techniques)** — overview context.
- **Module 9.8 (Taxonomy of Noise)** — where ZNE fits in the mitigation landscape.
- **Module 10.5 (Closing the Loop)** — integrating mitigation with the optimizer.

Variational Quantum Algorithms · Module 9 — Hardware Noise Models · Node 9.5
Concepts: quantum-noise, expectation-value, stochastic-optimization

Prerequisites: 9.2, 9.4
Enables: 9.8

hbar.university — own.your.cognition