# Noise Mitigation Techniques

Variational Quantum Algorithms · Module 9 — Hardware Noise Models

Node 9.4

## Position in Knowledge Graph

**System:** Variational Quantum Algorithms **Structural Domain:** Hardware Noise Models **Node:** 9.4

Noise mitigation is the collection of **post-processing techniques** that reduce the effect of noise on the *expectation value* you extract from a noisy quantum circuit — without doing full fault-tolerant quantum error correction. The distinction is critical:

- **Error correction** (QEC) fixes the underlying *quantum state* using redundant encoding and active syndrome measurement. It is the long-term answer. It requires thousands of physical qubits per logical qubit and is not yet practical.
- **Error mitigation** does not fix the state. It uses classical post-processing to produce a better *estimate* of the noiseless expectation value from noisy runs. It is the NISQ-era answer.

Mitigation techniques pay for their accuracy with **sample complexity** — more circuit runs, not more qubits. This matches the VQA resource profile well, where qubits are scarce but classical compute is cheap.

This node surveys the major mitigation techniques: zero-noise extrapolation (ZNE), probabilistic error cancellation (PEC), readout error correction, Clifford data regression, symmetry verification, dynamical decoupling, and virtual distillation. Node 9.5 goes deep on ZNE specifically. Node 9.6 covers noise-adaptive ansatz design, which is a structurally different mitigation strategy.

---

## What Mitigation Can And Cannot Do

**Can:**

1. **Estimate the noiseless expectation value** $\langle O \rangle_{\text{ideal}}$ with reduced bias, at the cost of increased variance.
2. **Remove systematic error** from specific known noise sources (e.g., readout errors, coherent calibration errors).
3. **Exploit symmetries** to reject errors that break a conserved quantity.
4. **Extend the usable depth** of a circuit by a factor of 2-5x under typical NISQ conditions.

**Cannot:**

1. **Recover the full quantum state** — you only get expectation values, not wavefunctions.
2. **Work past a fundamental sampling floor** — the sample complexity grows exponentially in the effective noise, so eventually you can't afford the shots.
3. **Fix errors that are indistinguishable from the signal** — e.g., if noise has the same symmetry structure as the observable.
4. **Replace error correction** for deep circuits — the overhead makes deep circuits unreachable.

Mitigation is the **right tool for the NISQ regime**: shallow-to-moderate circuits, moderate noise, tolerance for increased shot count. Outside that regime (either very shallow or very deep), different tools apply.

---

## The Sample-Complexity Cost

A universal fact about mitigation: the reduction in bias comes with an increase in variance. Formally, if a mitigation technique multiplies the estimator by a factor $\gamma > 1$ (to undo noise shrinkage), the estimator variance grows by $\gamma^2$, and the required shot count to reach fixed precision grows the same way.

For modest noise ($p \sim 10^{-3}$ per gate, depth $\sim 100$), typical $\gamma$ values are 2-10, meaning 4-100x more shots. Manageable.

For large noise ($p \sim 10^{-2}$, depth $\sim 500$), $\gamma$ can blow up to $10^3$-$10^4$, meaning $10^6$-$10^8$x more shots. Unreachable.

**The practical sweet spot:** NISQ circuits with gate fidelity 99.5%-99.9% and depth 50-200 gates. Mitigation works here; it fails outside.

This sample-complexity barrier is why mitigation is not a long-term answer. It's a bridge to fault-tolerance.

---

## Readout Error Correction

The simplest and most impactful mitigation. Measurement errors on current hardware are typically 1%-5% per qubit — much larger than gate errors. Luckily, they are also the easiest to characterize and correct.

**Procedure:**

1. **Characterize** the confusion matrix $M$, where $M_{ij}$ is the probability of observing outcome $i$ given the true outcome was $j$. This is done by preparing each computational basis state and measuring, typically with a few thousand shots.
2. **Invert** the confusion matrix: $M^{-1}$ maps observed probabilities back to ideal ones.
3. **Apply** $M^{-1}$ to the observed probability vector to recover the corrected distribution.

For $n$ qubits, the full confusion matrix is $2^n \times 2^n$ — prohibitive beyond $\sim 10$ qubits. Practical variants:

- **Tensor product assumption:** treat each qubit's readout error independently. The confusion matrix factors as $M = M_1 \otimes M_2 \otimes \cdots \otimes M_n$. Characterization is $O(n)$ and inversion is $O(n)$.
- **Iterative Bayesian unfolding:** avoid explicit matrix inversion; iteratively refine the distribution using Bayes' rule. Handles correlated readout errors.
- **M3 (Matrix-free Measurement Mitigation):** solves a linear system to mitigate without forming the matrix. Scales to hundreds of qubits (Nation et al. 2021).

**Effectiveness:** removes most readout error bias at modest cost. Should be applied to *all* VQA experiments — it's essentially free relative to the underlying circuits.

---

## Zero-Noise Extrapolation (ZNE) — Brief

Core idea: run the circuit at several artificially amplified noise levels, fit the expectation value as a function of noise strength, and extrapolate to zero noise.

**Procedure:**

1. Define a noise-amplification method (unitary folding, pulse stretching, or identity insertion) that scales the effective noise by factor $\lambda \in \{1, \lambda_2, \lambda_3, \ldots\}$.
2. Run the circuit at each amplification level and estimate $\langle O \rangle(\lambda)$.
3. Fit a curve (linear, exponential, Richardson) to $\langle O \rangle$ vs $\lambda$.
4. Extrapolate to $\lambda = 0$ to get the estimated noiseless value.

**Pros:** requires no detailed noise model, easy to implement, works with any observable.

**Cons:** extrapolation is extrapolation — the fit can be wrong, especially with large noise or non-monotonic curves. Typical accuracy improvement is 3-10x for NISQ regimes.

Node 9.5 goes into full detail: folding methods, fit choices, error analysis, practical recipes.

---

## Probabilistic Error Cancellation (PEC)

A more ambitious technique: **invert the noise channel directly** via a quasi-probability decomposition.

**Idea.** Suppose the noise channel $\mathcal{N}$ is known and invertible. Then $\mathcal{N}^{-1}$ is a (quasi-)map: it can be written as a linear combination of physically implementable channels, possibly with negative coefficients:

$$\mathcal{N}^{-1} = \sum_k c_k \mathcal{E}_k, \quad \sum_k c_k = 1, \quad c_k \in \mathbb{R}.$$

**Procedure.** At each circuit run, sample a $k$ with probability $|c_k|/\gamma$ (where $\gamma = \sum_k |c_k| \geq 1$ is the one-norm), apply the $\mathcal{E}_k$ operation, and multiply the measurement outcome by $\mathrm{sign}(c_k) \cdot \gamma$.

Averaging many such runs produces an unbiased estimator of the noiseless expectation value.

**The cost.** The variance blowup factor is $\gamma^2$, and $\gamma$ grows exponentially in the circuit depth for most realistic noise ($\gamma \sim e^{c \cdot L \cdot p}$ for noise rate $p$ over depth $L$). So PEC works for shallow circuits but becomes prohibitively expensive for deep ones.

**Pros:** unbiased (not an extrapolation), gives mathematically rigorous guarantees.

**Cons:** requires precise knowledge of the noise channel (often unavailable), has exponential sample overhead, and the "quasi-probability" implementation is operationally awkward.

**Status in 2026:** PEC works on gate counts of 50-200 for superconducting hardware. Above that, sample complexity kills it. It is typically combined with ZNE (PEC for the short-range errors, ZNE for what remains).

**Reference:** Temme, Bravyi, Gambetta (2017) introduced PEC. Subsequent work by Takagi et al. (2022) gave sample complexity lower bounds showing the exponential is fundamental.

---

## Clifford Data Regression (CDR)

A clever mitigation strategy using the fact that Clifford circuits are **classically simulable**.

**Procedure.**

1. Take your VQA circuit $U(\boldsymbol{\theta})$.
2. Construct a set of "nearby" Clifford circuits by replacing non-Clifford gates with their closest Clifford approximants.
3. Classically simulate each Clifford circuit to get the ideal expectation values $\{C_i^{\text{ideal}}\}$.
4. Run each Clifford circuit on the noisy hardware to get $\{C_i^{\text{noisy}}\}$.
5. Fit a regression model $f : C_i^{\text{noisy}} \mapsto C_i^{\text{ideal}}$ — typically linear.
6. Apply $f$ to the noisy expectation value from the original (non-Clifford) circuit.

**Intuition.** The noise process is (approximately) consistent across Clifford and non-Clifford circuits — if you know how noise distorts the Cliffords, you can invert it for the non-Cliffords.

**Pros:** doesn't require a noise model, handles *systematic* bias in a principled way, often works when simple ZNE fails.

**Cons:** needs classical simulation of Clifford circuits (fine for moderate qubit counts; up to $\sim 100$ qubits for $\mathcal{O}(100)$-depth Cliffords), the "nearby Clifford" construction is heuristic, the regression can overfit.

**Reference:** Czarnik, Arrasmith, Coles, Cincio (2020) introduced CDR. Later extended to include polynomial regression and neural network models.

**Practical note:** CDR is the technique that most often outperforms ZNE on realistic VQA problems in the 50-qubit range. It should be considered a first-line mitigation alongside ZNE.

---

## Symmetry Verification

Many physical Hamiltonians have **conserved quantities** — particle number, spin parity, total charge. If the ideal quantum circuit preserves these symmetries, then measurement outcomes that violate them are *known to be wrong* and can be discarded.

**Procedure.**

1. Identify a conserved quantity $Q$ (typically a Pauli sum that commutes with the cost Hamiltonian).
2. Measure $Q$ (either as part of the cost estimation or via a separate ancilla-based check).
3. Discard any shot where $Q$ does not match the expected value.
4. Use only the remaining shots to estimate the cost.

**Example (chemistry VQE).** For a molecule with $N$ electrons, the particle number $\hat{N} = \sum_i \hat{n}_i$ is conserved. Discard any Fock-space outcome with $\hat{N} \neq N$. This removes errors that change particle number (which amplitude damping does).

**Pros:** cheap (no extra circuits), principled (error rejection is unbiased), removes a physically meaningful class of errors.

**Cons:** only works if the symmetry is actually preserved by the circuit (not all ansätze preserve particle number), reduces the *effective* shot count (shots violating the symmetry are discarded), doesn't help with errors that preserve the symmetry.

**Reference:** McArdle, Yuan, Benjamin (2019) developed symmetry verification for VQE. It's now standard in chemistry VQA pipelines.

---

## Dynamical Decoupling (DD)

A mitigation technique at the **pulse level** rather than the circuit level. It reduces coherent and low-frequency noise by rapid unitary "refocusing" pulses.

**Intuition.** If you're sitting idle on a qubit for time $\tau$ and there's a slowly varying environment field, the qubit acquires a phase error. But if you apply an $X$ pulse halfway through, the phase error accumulated in the first half gets reversed in the second half — the net error is small.

**Standard DD sequences.**

- **Hahn echo:** a single $\pi$ pulse in the middle. Cancels static inhomogeneous dephasing.
- **CPMG:** a train of $\pi$ pulses at regular intervals. Cancels frequency noise up to a higher frequency.
- **XY-4, XY-8:** pulse sequences that protect against more general noise (not just dephasing).
- **Uhrig DD:** non-uniform pulse spacing optimized for specific noise spectra.

**Application to VQAs.** In a VQA, during the "dead time" when qubits are idle (waiting for other gates to finish, or during measurement of other qubits), DD pulses can be inserted to refocus errors. This is especially useful for superconducting qubits with long dead times.

**Pros:** reduces coherent and low-frequency noise at pulse level, no computational overhead, transparent to the circuit (the DD pulses are inserted into the idle periods).

**Cons:** doesn't help with high-frequency (Markovian) noise — and most gate errors *are* Markovian. Primarily helps with idle errors.

**Reference:** Viola, Lloyd (1998) introduced DD theoretically. Modern usage in VQAs: Pokharel et al. (2018) showed 3-5x fidelity improvement on IBM hardware via DD insertion.

---

## Virtual Distillation

A more recent (2020+) technique that uses **multiple copies** of the noisy state to purify it.

**Idea.** If the noisy state is $\rho = (1-\epsilon)|\psi\rangle\langle\psi| + \epsilon\sigma$ with $\epsilon \ll 1$, then $\rho^2/\mathrm{Tr}(\rho^2)$ is closer to $|\psi\rangle\langle\psi|$ than $\rho$ is. Higher powers $\rho^M$ are closer still.

**Procedure.**

1. Prepare $M$ copies of the noisy state.
2. Use a circuit that implements the "derangement" operation on the copies.
3. Measure an observable $O$ along with the derangement.
4. The combined measurement gives an unbiased estimator of $\mathrm{Tr}(\rho^M O)/\mathrm{Tr}(\rho^M)$ — which is close to $\langle\psi|O|\psi\rangle$.

**Pros:** no noise model required, exponential suppression in $M$ for the coherent error part.

**Cons:** requires $M$ copies of the state (so $M\times$ qubit count), the derangement circuit introduces its own noise, the method is biased in a specific way that needs correction.

**Reference:** Huggins et al. (2021), Koczor (2021) independently introduced virtual distillation. Still somewhat research-stage; not yet standard in VQA pipelines but is the main 2024-2026 research direction for error mitigation.

---

## Mitigation In Practice: A Decision Tree

**For VQA practitioners**, the practical question is: *which* mitigation technique should I apply?

1. **Always apply readout error correction.** Cost is negligible, benefit is significant. Non-negotiable.

2. **If your Hamiltonian has a conserved symmetry, apply symmetry verification.** Also cheap and principled.

3. **If your hardware has long idle times, apply dynamical decoupling.** Costs nothing in shots; reduces idle errors.

4. **For the "main" mitigation**, choose between ZNE and CDR based on depth:

   - Shallow ($L < 100$): ZNE with linear or exponential fit. Simple and effective.
   - Moderate depth ($100 < L < 300$): CDR often outperforms ZNE. Use if you can classically simulate the Cliffords.
   - Deep ($L > 300$): Neither works well. Consider reducing depth or switching to a shallower ansatz.

5. **For research-grade VQAs**, consider PEC (if noise model is known) or virtual distillation (if qubit count allows). Higher effort, can give best-in-class results.

**Combining techniques.** Readout correction, symmetry verification, and DD can be stacked with ZNE or CDR. Be careful about combining PEC with anything else — the quasi-probability math gets messy.

---

## Mitigation And The Thesis

The thesis view of mitigation:

1. **Mitigation is a form of classical side-computation applied to the noisy optimizer.** It is not part of the quantum dynamics — it is outer-loop post-processing.

2. **HOPSO's structure makes mitigation cleaner.** Because HOPSO separates the dynamical update from the objective generator, mitigation can be applied to the objective generator (which produces $C_{\text{noisy}}$) without touching the optimizer dynamics. This is architecturally clean.
3. **Mitigation reduces bias but not the fundamental signal-to-noise tradeoff.** The usable depth is set by the fidelity budget plus the shot budget, and mitigation just shuffles the budget between the two. No free lunch — the question is whether your VQA can afford the tradeoff.
4. **The future of mitigation is integration with optimization.** Mitigation techniques applied in *series* (observe, correct, optimize) are one architecture; applied in *parallel* (the optimizer is aware of the mitigation cost and adjusts accordingly) is another. Module 10 takes up the latter.

---

## Structural Role

This node is the **survey of the mitigation toolkit** for Module 9. It introduces each major mitigation technique, describes what it does and its cost, and gives a practical decision tree for when to use each.

It is the parent node for 9.5 (ZNE in depth) and 9.6 (noise-adaptive ansatz design). Together, these three nodes cover the main lines of attack against NISQ noise for VQAs.

---

## Relations to Other Concepts

- **Temme, Bravyi, Gambetta (2017)** — PEC original.
- **Li, Benjamin (2017)** — ZNE original.
- **Czarnik et al. (2020)** — CDR original.
- **McArdle et al. (2019)** — symmetry verification for VQE.
- **Huggins et al. (2021), Koczor (2021)** — virtual distillation.
- **Viola, Lloyd (1998)** — DD theory.
- **Nation et al. (2021)** — M3 readout mitigation.
- **Cai et al. (2023)** — review of quantum error mitigation.

---

## Worked Example — Mitigation Stack For A 10-Qubit VQE

Consider a 10-qubit VQE for a chemistry problem, running on superconducting hardware with: - Two-qubit gate fidelity 99.5%. - Readout fidelity 97%. - Circuit depth $L = 80$ (including 40 two-qubit gates). - Particle number conservation (electrons fixed at 5). - $10^5$ shots budget per cost evaluation.

**Without mitigation.** - Circuit fidelity: $(0.995)^{40} \approx 0.818$. Then readout on 10 qubits: $(0.97)^{10} \approx 0.737$. Combined: 0.60. - Cost estimate is biased — off by roughly 40% relative to true value.

**With readout mitigation.** - Readout bias corrected by tensor-product M matrix inversion. Residual readout error: $\sim 1\%$. Combined fidelity: $0.818 \cdot 0.99 \approx 0.81$. - Cost estimate bias: $\sim 19\%$.

**Plus symmetry verification (particle number).** - Discards shots that violate $\hat{N} = 5$. Roughly 15% of shots are discarded. - Effective shots: $0.85 \cdot 10^5 = 8.5 \times 10^4$. - Bias from remaining errors (those preserving particle number): $\sim 10\%$.

**Plus ZNE (linear fit, $\lambda = 1, 1.5, 2$).** - Uses 3x the shot budget ($3 \times 8.5 \times 10^4 = 2.55 \times 10^5$) to get the extrapolation points. - After linear fit, residual bias: $\sim 2\%$. - Variance roughly doubled compared to single $\lambda = 1$ run.

**Final result:** cost estimate with $\sim 2\%$ bias, requires $\sim 3\times$ shot count relative to raw. Acceptable for VQA optimization.

**Observation.** Each mitigation layer halves (or better) the residual bias. The stack is additive. Getting below 2% bias requires either CDR or PEC, both of which are significant investments.

---

## Visualization

*Pending. A bar chart showing bias and variance for each mitigation layer applied sequentially: (no mitigation) → +readout → +symmetry → +ZNE. Bias decreases; variance increases. Caption: "The bias-variance tradeoff of mitigation stacks."*

---

## Exercises

1. For a depolarizing channel at rate $p = 0.01$ and a single-qubit observable $X$, compute the PEC quasi-probability decomposition and the resulting $\gamma$.
2. Show that symmetry verification is *unbiased* — the estimator using only symmetric shots has the same expectation as the ideal estimator, provided the symmetry is exactly preserved by the ideal circuit.
3. (VQA) For a 20-qubit VQA with depth 100, gate fidelity 99.5%, readout fidelity 97%, and shot budget $10^6$, design a mitigation stack and estimate the final cost bias and effective shot count.

---

## Research Extensions

- **Adaptive mitigation** — selecting the mitigation technique based on the current circuit properties.
- **Learning-based mitigation** — neural networks trained to correct noisy expectation values given side information about the circuit.
- **Hybrid PEC/ZNE** — combining the two techniques to trade off bias and variance optimally.
- **Mitigation-aware optimizer design** — optimizers that account for the mitigation cost in their loss function.
- **Scalable virtual distillation** — reducing the qubit overhead of copy-based techniques.

---

## Cross-links to Other Courses

- **Quantum Error Correction** — the long-term answer that mitigation approximates.
- **Statistical Estimation Theory** — bias-variance tradeoff as a fundamental constraint.
- **Module 9.5 (Zero-Noise Extrapolation)** — the detailed treatment of the most widely used technique.
- **Module 9.6 (Noise-Adaptive Ansatz Design)** — a structurally different mitigation strategy.
- **Module 10 (Adaptive Control Systems)** — mitigation-aware optimizer design.

---

*Variational Quantum Algorithms · Module 9 — Hardware Noise Models · Node 9.4*
**Concepts:** quantum-noise, expectation-value, quantum-channels
**Prerequisites:** 9.1, 9.2, 9.3
**Enables:** 9.5, 9.6, 9.8

*hbar.university — own.your.cognition*