Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Budget-Aware Neural Error Mitigation forVariational Quantum Algorithms

Guilin Zhang

guilin.zhang@workday.com

Workday (United States)

Kai Zhao

Workday (United States)

Xu Chu

Workday (United States)

Research Article

Keywords: quantum error mitigation, variational quantum algorithms, neural networks, NISQ, zero-noise extrapolation, Clifford data regression

Posted Date: August 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10093362/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Budget-Aware Neural Error Mitigation for Variational Quantum Algorithms

Guilin Zhang[1], Kai Zhao[1], and Xu Chu[1]

[1]Workday AI Research

Abstract

Variational quantum algorithms (VQAs) are limited by device noise, and error mitigation is the standard remedy on near-term hardware. Choosing a mitigation method for deployment requires comparing its true cost, yet the usual comparison hides it: it fixes the number of shots per circuit execution, while the methods differ sharply in how many executions they need per estimate. Zero-noise extrapolation (ZNE) needs three to five, Clifford data regression (CDR) more than thirty per target circuit, and a trained neural model exactly one; ZNE's accuracy further rests on a noise-scaling assumption that real devices violate. Inside a variational loop that evaluates thousands of parameter settings, it is this execution multiplicity, not the accuracy at unlimited shots, that decides which method is affordable. Our premise is that the comparison that matters is at *equal total shot budget per evaluation point*. We present budget-conditioned neural error mitigation (NEM): one network corrects a single shot-limited measurement from circuit and device features, conditioned on the shot-noise scale so that one model serves every budget. We evaluate every method—including each ZNE fold point and each CDR training circuit—under a protocol that spends the same total shots per evaluation point. Under it, NEM is the most accurate method at every budget under coherent miscalibration (reducing error by 72–81%), and the only method that beats unmitigated execution at budgets up to 2^{14} shots under stochastic device noise (11–20%); under miscalibration CDR is also positive but overtakes NEM only at 2^{16} shots, at 32× the per-evaluation cost. We further characterize when extrapolation fails: under control-offset miscalibration, folding ZNE returns a worse answer than no mitigation on 38–63% of instances in simulation, and on up to 88% on a real IBM Heron processor. On that processor, fine-tuning the model on a small on-device set—120 circuits with classically computed labels—narrows the simulation-to-hardware gap, recovering a 39% error reduction on held-out device instances from the same calibration epoch. Improvements hold from 4 to 20 qubits. Budget-aware accounting, we argue, is the comparison that matters for deployment.

Keywords: quantum error mitigation, variational quantum algorithms, neural networks, NISQ, zero-noise extrapolation, Clifford data regression

1 Introduction

Variational quantum algorithms such as the variational quantum eigensolver (VQE) and the quantum approximate optimization algorithm (QAOA) are leading candidates for near-term

Code and data: `https://github.com/GuilinDev/Quantum-Error-Mitigation`

quantum advantage, but their expectation-value estimates are biased by device noise (Preskill, 2018; Cerezo et al., 2021; Bharti et al., 2022; Tilly et al., 2022). Error mitigation aims to remove this bias without the qubit overhead of error correction (Temme et al., 2017; Cai et al., 2023). The most widely used technique, zero-noise extrapolation, amplifies noise by gate folding and extrapolates the measurements back to the zero-noise limit (Temme et al., 2017; Li and Benjamin, 2017; Giurgica-Tiron et al., 2020; Kim et al., 2023). Learning-based methods such as Clifford data regression instead fit a noisy-to-ideal map on classically simulable circuits run on the same device (Czarnik et al., 2021; Strikis et al., 2021).

These methods are usually compared at a fixed, generous number of shots *per circuit execution*. That comparison hides a first-order practical distinction: the methods need very different numbers of executions per estimate. ZNE needs three to five per mitigated value. CDR runs one execution per training circuit—more than thirty—for every new target circuit. A neural model trained offline needs exactly one. Inside a VQA loop that evaluates thousands of parameter settings under a per-iteration shot budget, these multiplicities decide which method is usable, not the accuracy at unlimited shots. Bultrini et al. (2023) made the budget axis explicit for classical methods and found the ranking is budget-dependent; we extend that analysis to learned mitigation and make the budget an input of the model itself.

This paper makes four contributions.

(i) Budget-conditioned neural mitigation. We train a neural error mitigation model that maps a single S-shot measured estimate, circuit-structure features, and device-characterization features to the ideal expectation value, and we condition it on the shot-noise scale $1/\sqrt{S}$ so that one network serves every budget (Section 3).

(ii) An equal-budget evaluation protocol and Pareto analysis. Every method receives the same total shot budget B per evaluation point, split across the executions it requires, with every execution—including each ZNE fold point and each CDR training circuit—drawn as measured shots under the same frozen noise realization (Section 4). Under this protocol NEM is the most accurate method at every budget under coherent miscalibration, and the only method that beats unmitigated execution at $B \leq 2^{14}$ under stochastic device noise; CDR overtakes it only at $B = 2^{16}$ (Section 5.1).

(iii) A characterization of extrapolation failure. Under physically motivated control-offset miscalibration (Krantz et al., 2019), unitary folding amplifies the coherent angle error while the observable responds non-monotonically. ZNE then underperforms unmitigated execution on 38–63% of instances, with unbounded single-instance error. We map this failure boundary against the miscalibration strength, quantify the tail risk (Section 5.2), and confirm the failure on a real IBM Heron processor (Section 5.3).

(iv) Scaling and controls. Mean error reductions of 61–85% under structured noise persist without degradation from 4 to 20 qubits, and a direct-prediction control—predicting $\langle H \rangle_{\text{ideal}}$ from features alone, without the noisy measurement—never beats unmitigated execution at any size, confirming that mitigation, not memorization, drives the result (Sections 5.4 and 5.5).

We are explicit about what NEM does not win: given 32× the per-evaluation quantum budget, CDR is the most accurate method in every regime we study. The case for learned mitigation is not accuracy supremacy but its position on the budget–accuracy Pareto frontier and its bounded failure behavior.

2 Related work

Classical mitigation. ZNE (Temme et al., 2017; Li and Benjamin, 2017) assumes the bias varies smoothly with a controllable noise-scale factor; digital implementations amplify noise by unitary folding (Giurgica-Tiron et al., 2020), and the technique has produced headline demonstrations on hardware at the 100-qubit scale (Kim et al., 2023). Its reliability, however, is contingent on the extrapolation ansatz matching the actual noise response, a sensitivity we quantify in Section 5.2. Probabilistic error cancellation (Temme et al., 2017; Van Den Berg et al., 2023) yields unbiased estimates at exponential sampling cost; virtual distillation (Huggins et al., 2021; Koczor, 2021) suppresses errors using multiple state copies at additional quantum cost. CDR (Czarnik et al., 2021) fits a linear noisy-to-ideal regression on near-Clifford circuits executed on the device, paying tens of executions per target circuit but inheriting robustness to noise structure because it calibrates on the device as it actually is; Strikis et al. (2021) extend the idea with learning-based protocols. Bultrini et al. (2023) benchmark these methods under a unified framework as a function of total shot budget and find that cheap methods win at small budgets while data-hungry methods win at large ones—the organizing observation our protocol builds on.

Learned mitigation. Neural networks have been used to reconstruct states and observables from noisy quantum data (Torlai et al., 2018) and to mitigate errors in quantum simulation (Bennewitz et al., 2022). Liao et al. (2024) benchmark random forests, multilayer perceptrons, and graph networks for expectation-value mitigation, including at the 100-qubit scale on hardware, and find learned models competitive with ZNE at lower quantum cost—an observation our protocol makes systematic. Closest to this work, Cantori et al. (2024) train convolutional networks to predict ideal expectation values of parameterized circuits from noisy values and circuit structure, including transfer to circuit sizes beyond the training set, and Liao et al. (2025) train noise-agnostic models with data-augmented training, outperforming ZNE and CDR on hardware at 4–6 qubits. In this journal, Adeniyi and Kumar (2025) adapt a neural mitigator to per-circuit error characteristics and validate it on IBM hardware. Neither line of work controls for quantum cost: comparisons grant every method full per-execution shot counts, and the per-evaluation execution multiplicity is not part of the analysis. Our contribution is orthogonal and composable with theirs: an equal-budget protocol, budget conditioning of the model itself, and an explicit failure-mode analysis of the classical alternative.

3 Budget-conditioned neural mitigation

A VQA prepares $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle^{\otimes n}$ and estimates the objective $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\,H\,|\psi(\boldsymbol{\theta})\rangle$. On hardware one observes an S-shot estimate of the noisy expectation,

$$\widehat{\langle H\rangle}_{\text{noisy}} \;=\; \langle H\rangle_{\text{ideal}} + \delta(\boldsymbol{\theta}, \mathcal{E}) + \varepsilon_S, \tag{1}$$

where δ is the bias induced by the noise process $\mathcal{E}$ and ε_S is measurement noise with scale proportional to $1/\sqrt{S}$. We learn a correction function

$$f_\phi\Big(\widehat{\langle H\rangle}_{\text{noisy}},\, \mathbf{x}_c,\, \mathbf{x}_n,\, 1/\sqrt{S}\Big) \;\approx\; \langle H\rangle_{\text{ideal}}, \tag{2}$$

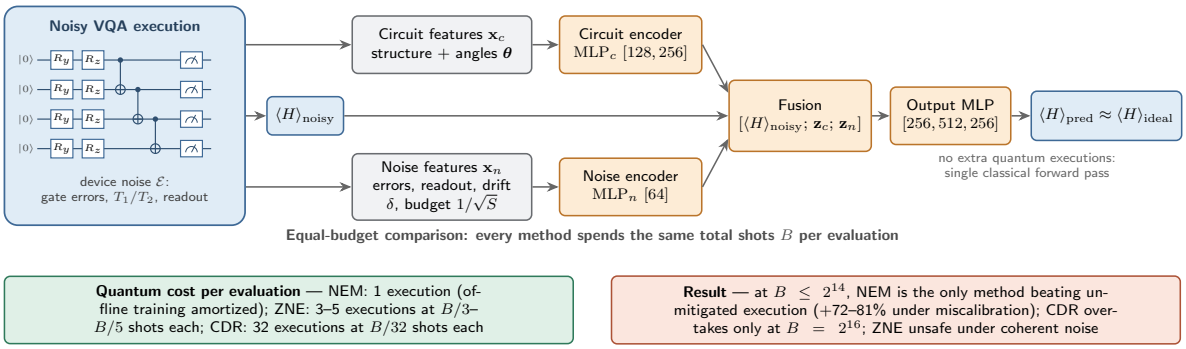


Figure 1: Budget-conditioned neural error mitigation. A single S-shot noisy VQA execution is combined with circuit-structure and device-characterization embeddings—including the shot-noise scale $1/\sqrt{S}$—to predict the ideal expectation value with one classical forward pass. Under an equal total budget B per evaluation point, competing methods must split B across their 3–32 required executions.

where $\mathbf{x}_c$ collects circuit features (qubit count, layer count, parameter count, and the variational angles themselves; dimension $3 + 4n$ for the ansatz family used here) and $\mathbf{x}_n$ collects device-characterization features (gate error rates, readout asymmetries, and—when characterized—the calibration-drift scale; Section 4).

Architecture. Figure 1 shows the model. Two multilayer-perceptron encoders embed the feature groups: the circuit encoder maps $\mathbf{x}_c$ through hidden widths $[128, 256]$ to a 128-dimensional embedding, and the noise encoder maps $\mathbf{x}_n$ through width 64 to a 64-dimensional embedding. The embeddings are concatenated with the measured value and passed through an output network of widths $[256, 512, 256]$ (LayerNorm, GELU, dropout 0.15) that predicts a residual correction added to the input estimate. The standard configuration has 428K parameters; Section 5.5 shows accuracy is insensitive to capacity over a $5\times$ range, so the capacity is generous rather than necessary.

Training. The model minimizes the Huber loss with AdamW (learning rate 10^{-3}, weight decay 10^{-4}, cosine annealing with warm restarts) for 75 epochs—about five minutes on a CPU workstation at the sizes studied. Training pairs $(\widehat{\langle H \rangle}_{\mathrm{noisy}}, \mathbf{x}_c, \mathbf{x}_n, \langle H \rangle_{\mathrm{ideal}})$ are generated in simulation: parameters and noise realizations are sampled per example, the noisy input is an actual S-shot measured estimate, and the ideal label is computed by exact noiseless simulation. Ideal labels therefore restrict *training* to classically simulable sizes, a limitation we return to in Section 6.

Budget conditioning. We condition training on the shot-noise scale so that one model serves every deployment budget. Each training example draws its shot count S log-uniformly from 2^8 to 2^{16}, samples the noisy input at that S, and appends $1/\sqrt{S}$ to $\mathbf{x}_n$. The network thereby learns how much of the input fluctuation is measurement noise versus learnable bias—at small S it shrinks its trust in the measurement, at large S it corrects fine structure—so no retraining is needed across budgets.

Table 1: Per-evaluation quantum cost at total budget B.

Method	Executions	Shots per execution	Offline cost
Unmitigated	1	B	—
NEM (ours)	1	B	4.4K executions + 5 min training
ZNE (exponential)	3	$B/3$	—
ZNE (adaptive)	4	$B/4$	—
CDR	32	$B/32$	repeated per target circuit

Table 2: Break-even: distinct evaluation points $M^\star$ beyond which NEM's total quantum cost (offline plus one execution per point) falls below the baseline's.

NEM vs.	Per-eval. executions	Break-even $M^\star$
ZNE (exponential)	3	2,200
ZNE (adaptive)	4	1,467
CDR	32	142

Costs. Data generation is a one-time cost of 4,400 circuit executions per device and noise regime, amortized over every subsequent evaluation; inference adds one classical forward pass (under one millisecond) to a single quantum execution. Table 1 summarizes per-evaluation costs against the baselines.

Break-even. The offline cost is recovered after a modest number of evaluations. Let M be the number of distinct evaluation points—parameter settings, or target circuits—that a deployed model serves before retraining. Counting every execution at equal per-execution shots, NEM's total quantum cost is $4400 + M$, against kM for ZNE ($k = 3$ or 4) and $32M$ for CDR. The baselines carry no offline cost, but they pay their multiplicity at *every* point: CDR, in particular, re-fits its near-Clifford regression for each target circuit. NEM is cheaper in total once M exceeds $M^\star = \lceil 4400/(k-1) \rceil$: 142 points relative to CDR, 1,467 relative to adaptive ZNE, and 2,200 relative to exponential ZNE (Table 2). A VQE optimization loop evaluates 10^3–10^4 parameter settings, so it clears the CDR break-even by one to two orders of magnitude within a single run. The bound is conservative: training shots are drawn from 2^8–2^{16}, so a deployment budget $B > 2^8$ per execution amortizes the offline cost faster still. This break-even counts executions at equal per-execution shots, whereas the equal-budget protocol of Section 4 fixes the total shots per evaluation; the two are complementary accountings of the same quantum cost—one asks how many evaluations repay the offline cost, the other which method is most accurate at a fixed per-evaluation budget.

4 Experimental protocol

Circuits and observables. We use hardware-efficient VQE ansatzes (alternating R_Y, R_Z rotation layers and CNOT ladders, two layers) at $n = 4$–16 qubits with the mean-magnetization observable $H = \frac{1}{n}\sum_i Z_i$. Simulations use Qiskit Aer (Qiskit contributors, 2024): exact density-matrix count sampling up to 10 qubits and statevector trajectory sampling above, so every reported measurement carries genuine sampling statistics. Classical baselines use mitiq 1.0 (LaRose et al., 2022).

Noise regimes. Three regimes span the noise-structure spectrum:

- *Stochastic device noise*: per-instance device profiles (depolarizing gate errors, thermal relaxation, asymmetric readout) sampled over realistic ranges—the regime where smooth noise-scaling assumptions are most plausible.
- *Stochastic-correlated noise*: adds per-instance gate-error fluctuations and correlated ZZ crosstalk.
- *Coherent miscalibration*: models control-offset calibration error (Krantz et al., 2019). Working in the native gate basis $\{R_Z, \sqrt{X}, X, \mathrm{CNOT}\}$, every physical pulse executes its nominal angle plus an offset drawn once per device snapshot from $\mathcal{N}(\delta, (\delta/2)^2)$ with $\delta \in [0.02, 0.06]$ rad, on top of two-qubit depolarizing noise and asymmetric readout—so the coherent offset overlays a stochastic background, which some figures label "stochastic + coherent." Virtual R_Z rotations, implemented as exact software frame updates, are exempt. Because the miscalibration lives in the executed unitary, folded ZNE circuits and CDR training circuits face *the same* quantum process. Under folding, offsets in the inverted segment do not cancel—$R_Y(\theta+\delta)\,R_Y(-\theta+\delta) = R_Y(2\delta)$—so the coherent error grows linearly with the scale factor while the observable responds non-monotonically: the mechanism by which coherent errors defeat extrapolation.

Fairness protocol. Each test instance freezes one noise realization (a device snapshot). The raw measurement, the NEM input, every ZNE fold point, and every CDR training circuit are S-shot measured estimates under that same realization; ideal values are exact noiseless simulations. Baseline failures abort the run; there are no silent fallbacks. ZNE uses mitiq's Richardson, exponential ($y(\lambda) = a\,e^{-b\lambda}$ with the depolarizing asymptote), and adaptive-exponential factories with global folding at scale factors $\{1, 2, 3\}$; CDR uses 31 near-Clifford training circuits with half the non-Clifford gates retained. We report 150 test instances per configuration (100 for budget sweeps, 80 for the δ-sweep of Figure 3) and paired t-tests against unmitigated execution, and because extrapolation has heavy-tailed failures we report median absolute errors alongside means throughout.

5 Results

5.1 The equal-budget Pareto frontier

Under equal-budget comparison, NEM dominates at low-to-moderate budgets and is overtaken only at the largest budget we test. Figure 2 shows this; Table 6 in the appendix lists every value. Under coherent-miscalibration noise, NEM reduces the mean error by 72%, 77%, 81%, and 81% at $B = 2^{10}, 2^{12}, 2^{14}, 2^{16}$ respectively—flat in B, because the learned correction targets the systematic bias and its training cost is already amortized. CDR, whose per-execution shot count is $B/32$, climbs from +23% at $B = 2^{10}$ to +84% at $B = 2^{16}$, overtaking NEM only at the largest budget. Both ZNE variants are *negative* at essentially every budget in this regime (-36% to -471% in the mean); the lone exception—adaptive ZNE at $B = 2^{14}$ (+7% mean)—is a heavy-tail artifact whose median still sits at the unmitigated level. Under stochastic device noise the picture compresses (the residual error is dominated by shot noise), but the ordering at low budget is the same: NEM is the only positive method at $B \le 2^{14}$ (+11 to +20%), and CDR turns positive only at $B = 2^{16}$ (+47%).

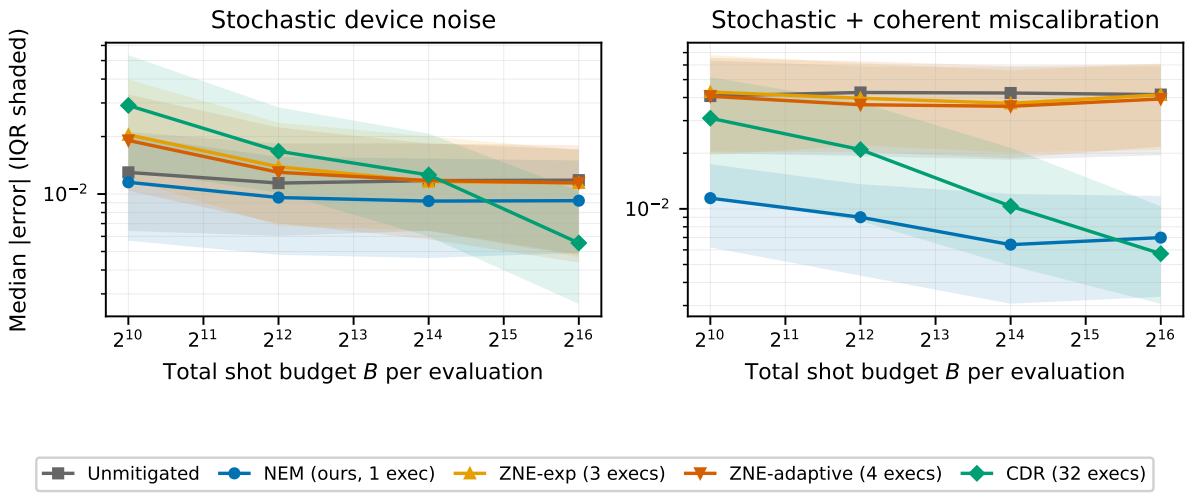


Figure 2: Equal-budget comparison at $n = 8$ (100 instances; median |error|, interquartile range shaded). Every method spends the same total budget B per evaluation point, split across its required executions (Table 1). NEM is the most accurate method at every budget under miscalibration and the only one below the unmitigated baseline at $B \leq 2^{14}$ under stochastic noise; CDR crosses below NEM only at the largest budget. ZNE medians track the unmitigated line: for the typical instance extrapolation removes nothing, while its mean error diverges (Section 5.2).

The practical reading is direct. A VQE optimization loop evaluates 10^3–10^4 parameter settings at a per-iteration budget of 10^3–10^4 shots. That regime is exactly where single-execution learned mitigation is the most reliable option—the only one that helps under stochastic noise, and the most accurate under miscalibration; per-instance methods match it on accuracy only when the per-evaluation budget grows by more than an order of magnitude.

5.2 When extrapolation fails, and how badly

Figure 3 sweeps the coherent bias δ from 0 to 0.08 rad over a fixed stochastic background, and the methods separate cleanly. ZNE's *median* improvement never exceeds +2% anywhere on the sweep—including at $\delta = 0$, because the remaining bias there is dominated by readout error, which gate folding does not amplify; extrapolation removes the gate-error component and faithfully retains the rest.

Its *mean* error, however, explodes as δ grows (for exponential extrapolation, −646% at $\delta = 0.06$), driven by heavy tails. Table 3 shows the anatomy at one representative cell, and Table 7 in the appendix shows every structured-regime cell: ZNE returns a result worse than the unmitigated estimate on 38–63% of instances, with single-instance errors reaching three orders of magnitude above the unmitigated level when the extrapolation fit diverges. Extrapolation is an unbounded operation: at n=12 we observe a single instance where the adaptive variant's error reaches 371 in observable units, more than three orders of magnitude above the worst unmitigated error in that cell.

NEM, by contrast, tracks CDR at small δ (+82% versus +83% mean at $\delta = 0$) at 1/32 of the quantum cost. Its improvement then declines smoothly to +44% at $\delta = 0.08$—still inside its $[0, 0.09]$ training range, so this is genuine learning difficulty, not extrapolation—as the

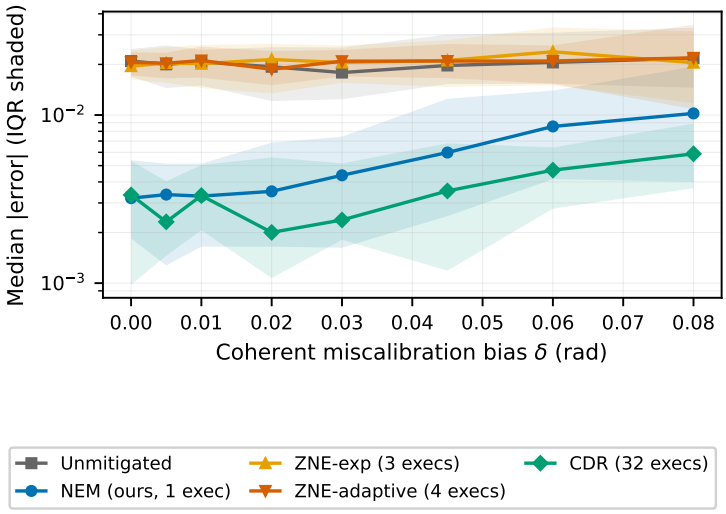


Figure 3: Median |error| versus miscalibration strength δ at $n = 8$ (80 instances per point, 8,192 shots per execution). ZNE medians coincide with the unmitigated baseline at every δ, including $\delta = 0$, where the residual bias is dominated by readout error that folding does not amplify. NEM matches CDR at small δ at 1/32 the quantum cost, and its improvement declines smoothly as the unlearnable coherent share grows. A single NEM is trained once on $\delta \sim \mathcal{U}(0, 0.09)$—wider than the $[0.02, 0.06]$ range used in the other experiments—and evaluated across the sweep, so every point is in-distribution and no model is retrained per δ.

high-dimensional oscillatory component of the coherent bias exceeds what the network can learn from 4,000 samples. CDR, which re-fits per instance, stays at 69–86%.

The asymmetry in failure behavior matters as much as the averages. The neural model's output is a bounded residual correction: across all 2,000 structured-regime test instances (4–20 qubits) its worst-case error stays *below* the worst unmitigated error, in every cell.

This bounded behavior can be tightened further with a safeguard. We train three seeds and fall back to the raw value when they disagree beyond a threshold calibrated on unlabeled data. This drops the worse-than-raw rate from 7% to 5% at a cost of three points of mean improvement, and in the low-noise stochastic regime it pins the model's worst case exactly at the unmitigated level. We view the safeguard as cheap insurance rather than a core mechanism.

5.3 Real-hardware validation

The failure analysis above is the paper's most consequential claim, so we test it on a real device. We run the hardware-efficient VQE on `ibm_marrakesh`, a 156-qubit Heron r2 superconducting processor, using four qubits at two circuit depths (2 and 5 ansatz layers). We compare unmitigated execution against folding ZNE under the equal-budget protocol, with ten and eight test instances respectively, 8,192 shots, and all jobs grouped in a single batch. Real devices carry intrinsic control-offset miscalibration, so no noise is injected—the device *is* the noise source.

Figure 4 shows the result: folding ZNE fails on real silicon exactly as predicted. Its mean error is 3.0× the unmitigated error at two layers and 4.1× at five layers, and it returns a worse answer than no mitigation on 80% and 88% of instances respectively. Unitary folding triples the circuit and amplifies the device's coherent and readout errors, which do not extrapolate

Table 3: Robustness statistics, coherent-miscalibration regime, n=10 (150 instances, 8,192 shots per execution). "Worse" is the fraction of instances on which the method's error exceeds the unmitigated error.

Method	Mean	Median	p95	Max	Worse
Unmitigated	.0412	.0385	.080	0.11	—
NEM (ours)	**.0061**	**.0044**	**.015**	**0.03**	7%
ZNE-Richardson	.0460	.0432	.099	0.15	63%
ZNE-exponential	.1569	.0385	.081	17.1	49%
ZNE-adaptive	.1220	.0394	.082	12.1	53%
CDR (32× cost)	.0035	.0025	.009	0.01	6%

to the zero-noise limit; the deeper circuit, with more gates to fold, fails harder. This is the same mechanism our simulations isolate (Section 5.2), now confirmed on hardware without any modelling assumptions. The result establishes that the unsafe regime we characterize is not a simulation artefact but a property of real devices.

This run validates the failure analysis; the neural mitigator's headline improvements are established in simulation, where training and test share the noise process. On hardware, a model trained purely on a digital twin shows a simulation-to-hardware gap (Appendix B). We attribute this gap primarily to the twin's mismatch rather than to learned mitigation itself—consistent with the affine fit below recovering a shrinkage factor below one (the twin overcorrects)—though with a single device we cannot fully separate the two. Fine-tuning the twin-trained model on a small on-device set—120 four-qubit circuits measured in a single batched job, labelled by exact classical simulation—raises its held-out error reduction on `ibm_marrakesh` from +16% (twin-only) to +39% on instances from the same calibration epoch, a gain stable across hyperparameters and seeds, and a deterministic affine recalibration of the twin's correction reaches +45% on this epoch (Appendix B) (Liao et al., 2025; Adeniyi and Kumar, 2025). Unlike ZNE, whose folding amplifies coherent error without bound, the learned correction is bounded and beneficial in aggregate once a small amount of same-epoch device data is supplied—though still worse than unmitigated on about a quarter of individual instances, and calibration-specific, as Appendix B details.

5.4 Scaling

NEM's improvement does not degrade as circuits grow from 4 to 20 qubits. Figure 5 shows this; Table 5 in the appendix lists every cell. Mean error reduction under coherent miscalibration is 72%, 77%, 80%, 85%, 76%, 80%, 77% at $n = 4, 6, 8, 10, 12, 16, 20$, and the stochastic-correlated regime is similarly flat (61–68%); in both structured regimes every NEM improvement is significant ($p < 10^{-4}$, paired t-test), with no degradation up to the largest size simulated. At n=16 under miscalibration, NEM's +80% matches CDR's +81%, and at n=20 its +77% matches CDR's +79%, each at 1/32 of the quantum cost. The per-size models use identical architecture and training budgets; only the input dimension grows, linearly in n.

Under purely stochastic noise the headroom itself shrinks: the unmitigated error approaches the shot-noise floor of the per-execution budget (S=8,192 up to n=10; 1,024 at n=12, 16; 512 at n=20, chosen for simulation cost). NEM's improvement then tracks that ceiling: +10 to +22% ($p < 0.05$) through $n = 12$, +7% ($p = 0.12$, not significant) at $n = 16$, and +11% ($p = 0.026$) at $n = 20$. More than half of the residual error there is irreducible measurement noise, so little bias is left for any single-execution method to remove.

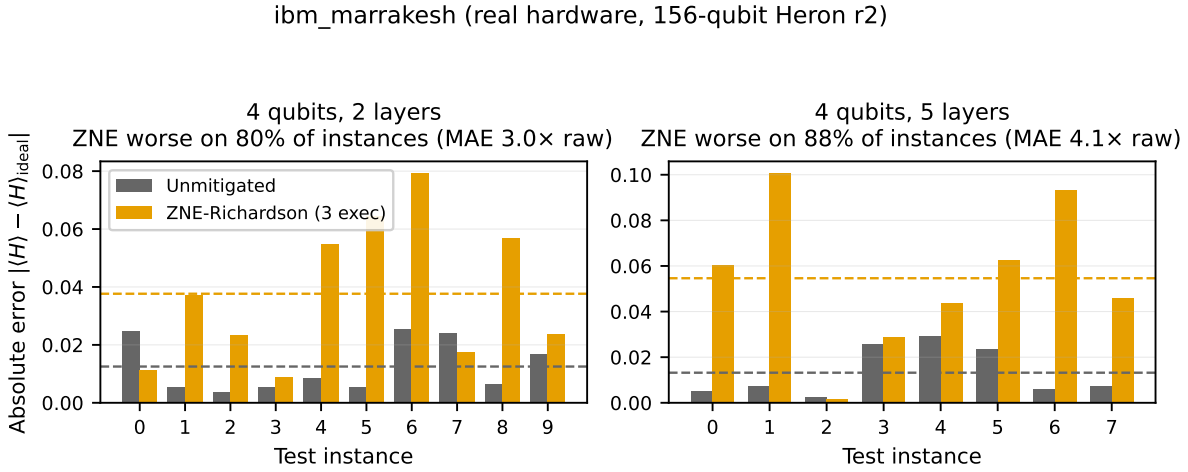


Figure 4: Folding ZNE on real hardware (`ibm_marrakesh`, 4 qubits). Per-instance absolute error of unmitigated execution (grey) versus ZNE-Richardson (orange); dashed lines are the means. ZNE's error exceeds the unmitigated error on 80% (2 layers) and 88% (5 layers) of instances, with mean error 3.0× and 4.1× the unmitigated level—the failure mode of Section 5.2, confirmed on a physical device.

5.5 Controls and ablations

Direct prediction. A skeptical reader should ask whether the network is mitigating or merely memorizing the expectation-value landscape. A control model with identical architecture but no access to the noisy measurement answers this: it is worse than unmitigated execution in every cell—never beating it at any size or regime, by as much as 927% (Table 5). The measurement is indispensable; the network's role is correction, not regression of $\langle H \rangle_{\text{ideal}}$ from circuit parameters.

Features. Table 4 isolates what the network uses, at $n = 8$ under miscalibration. Device-characterization features carry the conditioning signal: removing them triples the error (+191% MAE), while removing circuit features costs almost nothing (+0.3%) in this regime. The interpretation is physical: the learnable bias components—stochastic shrinkage, readout offset, and the mean effect of the drift δ—are functions of the device state, whereas the angle-dependent fine structure of the coherent bias is precisely the part the network cannot extract from 4,000 samples. Even with both feature groups removed, the measurement-conditioned model retains +46%; the direct-prediction control shows the converse, that features without the measurement are worthless.

Capacity. The concern that a 400K-parameter network is overparameterized for a roughly 40-dimensional input is correct—and harmless: an 88K-parameter model matches the 428K standard exactly (+81.4% versus +81.7% mean error reduction on identical data), so capacity can be cut 5× without loss.

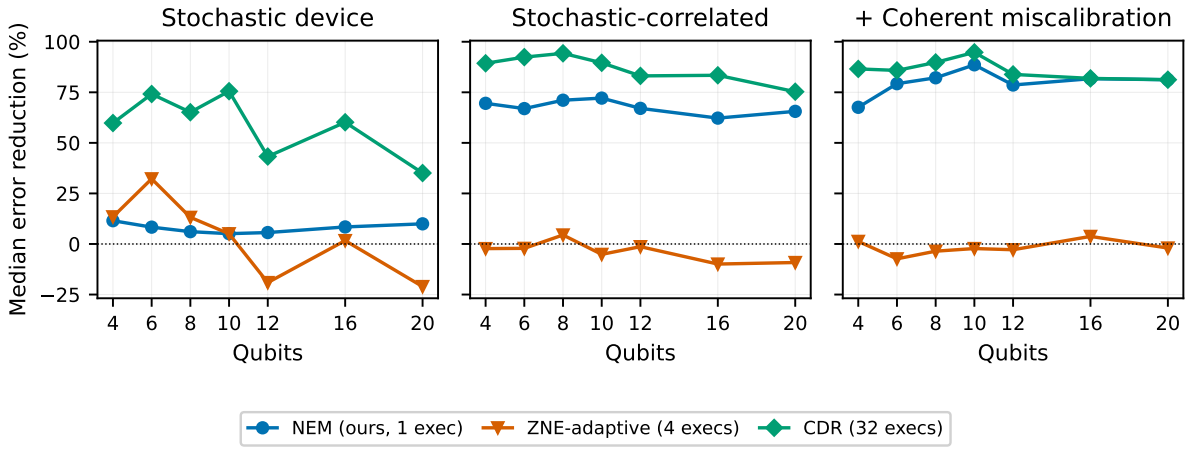


Figure 5: Median error reduction versus qubit count (150 instances per cell, 100 at n=20; S=8,192 shots per execution up to n=10; 1,024 at n=12, 16; 512 at n=20). NEM's improvement does not degrade from 4 to 20 qubits in the structured regimes; the direct-prediction control (omitted for scale) is negative at every size.

Table 4: Feature ablation, coherent-miscalibration regime, n=8 (150 instances; unmitigated MAE .0437).

Variant	MAE	vs. raw	vs. full
Full model	.0081	+81%	—
No circuit features	.0081	+81%	+0.3%
No noise features	.0236	+46%	+191%
Noisy value only	.0237	+46%	+192%
Direct prediction (no measurement)	.0995	−129%	—

6 Discussion and limitations

What we claim. At equal per-evaluation quantum budget, single-execution learned mitigation occupies the low-budget end of the accuracy–cost Pareto frontier, is the only method studied whose worst case stays bounded, and characterizes—rather than asserts—the regime where extrapolation-based mitigation is unsafe.

What we do not claim. CDR is more accurate wherever its 32× per-evaluation cost is affordable; our results quantify, not contest, that trade. Nor do we claim architectural novelty: the network is a plain residual MLP, and the contribution lies in the budget conditioning, the evaluation protocol, and the failure analysis.

Limitations. Several limitations bound these results and mark the directions we see as most promising. Our evaluation is primarily in simulation; we validate the ZNE failure mode on real hardware (Section 5.3), and the neural mitigator's headline improvements are established in simulation. A model trained purely on a digital twin shows a simulation-to-hardware gap, which a small on-device fine-tuning set closes (Appendix B); the fitted correction is specific to a

calibration epoch and would be refreshed periodically in deployment, as device-data methods such as CDR already require. The noise models, while physically motivated (calibration offsets, crosstalk, readout asymmetry), are otherwise not device-fitted. Training labels require exact classical simulation, restricting training to classically simulable sizes; transfer from small training circuits to larger deployment circuits (Cantori et al., 2024; Strikis et al., 2021) and label-free data augmentation (Liao et al., 2025) are the natural extensions for further reducing the device data the on-device fine-tuning above requires, and both compose with our budget conditioning. Our observables are diagonal (Z-type); non-diagonal Hamiltonians require basis-rotated measurement and may change feature design. Each model is trained per device family and noise regime; drift beyond the characterized feature ranges requires retraining, though the success of shot-budget conditioning suggests broader conditioning is feasible. Finally, our ansatz family is fixed; generalization across circuit families is untested here and is where graph- or sequence-structured encoders would plausibly earn their complexity.

7 Conclusion

Mitigation methods should be compared the way they are deployed: under a fixed quantum budget per evaluation point. Under that accounting, a budget-conditioned neural mitigator—one measured estimate in, one corrected estimate out—dominates the low-budget frontier that VQA optimization loops actually occupy, keeps its worst case bounded where extrapolation diverges, and concedes the high-budget regime to Clifford data regression honestly and quantifiably. We believe this budget-aware framing, more than any single architecture, is the durable contribution: it turns "which mitigation method is best?" into the answerable question "best at what price?".

Declarations

Funding. No funding was received for this work.

Competing interests. The authors declare no competing interests.

Data and code availability. All simulation code, experiment configurations, and per-instance result data are available at `https://github.com/GuilinDev/Quantum-Error-Mitigation`. Trained model checkpoints are available from the authors on request.

Author contributions. G.Z. designed the framework, implemented the experiments, and drafted the manuscript. K.Z. and X.C. contributed to the methodology and analysis and revised the manuscript. All authors reviewed and approved the final version.

References

Temitope Bolaji Adeniyi and Sathish A. P. Kumar. Adaptive neural network for quantum error mitigation. *Quantum Machine Intelligence*, 7(1):13, 2025. doi: 10.1007/s42484-024-00234-4.

Elizabeth R Bennewitz, Florian Hopfmueller, Bohdan Kulchytskyy, Juan Carrasquilla, and Pooya Ronagh. Neural error mitigation of near-term quantum simulations. *Nature Machine Intelligence*, 4(7):618–624, 2022. doi: 10.1038/s42256-022-00509-0.

Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022. doi: 10.1103/RevModPhys.94.015004.

Daniel Bultrini, Max Hunter Gordon, Piotr Czarnik, Andrew Arrasmith, M. Cerezo, Patrick J. Coles, and Lukasz Cincio. Unifying and benchmarking state-of-the-art quantum error mitigation techniques. *Quantum*, 7:1034, 2023. doi: 10.22331/q-2023-06-06-1034.

Zhenyu Cai, Ryan Babbush, Simon C Benjamin, Suguru Endo, William J Huggins, Ying Li, Jarrod R McClean, and Thomas E O'Brien. Quantum error mitigation. *Reviews of Modern Physics*, 95(4):045005, 2023. doi: 10.1103/RevModPhys.95.045005.

Simone Cantori, Andrea Mari, David Vitali, and Sebastiano Pilati. Synergy between noisy quantum computers and scalable classical deep learning for quantum error mitigation. *EPJ Quantum Technology*, 11:45, 2024. doi: 10.1140/epjqt/s40507-024-00256-8.

Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021. doi: 10.1038/s42254-021-00348-9.

Piotr Czarnik, Andrew Arrasmith, Patrick J Coles, and Lukasz Cincio. Error mitigation with clifford quantum-circuit data. *Quantum*, 5:592, 2021. doi: 10.22331/q-2021-11-26-592.

Tudor Giurgica-Tiron, Yousef Hindy, Ryan LaRose, Andrea Mari, and William J Zeng. Digital zero noise extrapolation for quantum error mitigation. In *IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 306–316, 2020. doi: 10.1109/QCE49297.2020.00045.

William J Huggins, Sam McArdle, Thomas E O'Brien, Joonho Lee, Nicholas C Rubin, Sergio Boixo, K Birgitta Whaley, Ryan Babbush, and Jarrod R McClean. Virtual distillation for quantum error mitigation. *Physical Review X*, 11(4):041036, 2021. doi: 10.1103/PhysRevX.11.041036.

Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023. doi: 10.1038/s41586-023-06096-3.

Bálint Koczor. Exponential error suppression for near-term quantum devices. *Physical Review X*, 11(3):031057, 2021. doi: 10.1103/PhysRevX.11.031057.

Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019. doi: 10.1063/1.5089550.

Ryan LaRose, Andrea Mari, Sarah Kaiser, Peter J. Karalekas, Andre A. Alves, Piotr Czarnik, Mohamed El Mandouh, Max H. Gordon, Yousef Hindy, Aaron Robertson, Purva Thakre, Misty Wahl, Danny Samuel, Rahul Mistri, Maxime Tremblay, Nick Gardner, Nathaniel T. Stemen, Nathan Shammah, and William J. Zeng. Mitiq: A software package for error mitigation on noisy quantum computers. *Quantum*, 6:774, 2022. doi: 10.22331/q-2022-08-11-774.

Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):021050, 2017. doi: 10.1103/PhysRevX.7.021050.

Haoran Liao, Derek S Wang, Iskandar Sitdikov, Ciro Salcedo, Alireza Seif, and Zlatko K Minev. Machine learning for practical quantum error mitigation. *Nature Machine Intelligence*, 6: 1478–1486, 2024. doi: 10.1038/s42256-024-00927-2.

Manwen Liao, Yan Zhu, Giulio Chiribella, and Yuxiang Yang. Noise-agnostic quantum error mitigation with data augmented neural models. *npj Quantum Information*, 11:8, 2025. doi: 10.1038/s41534-025-00960-y.

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018. doi: 10.22331/q-2018-08-06-79.

Qiskit contributors. Qiskit: An open-source framework for quantum computing. Zenodo, 2024.

Armands Strikis, Dayue Qin, Yanzhu Chen, Simon C Benjamin, and Ying Li. Learning-based quantum error mitigation. *PRX Quantum*, 2(4):040330, 2021. doi: 10.1103/PRXQuantum.2.040330.

Kristan Temme, Sergey Bravyi, and Jay M Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18):180509, 2017. doi: 10.1103/PhysRevLett.119.180509.

Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022. doi: 10.1016/j.physrep.2022.08.003.

Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018. doi: 10.1038/s41567-018-0048-5.

Ewout Van Den Berg, Zlatko K Minev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse pauli-lindblad models on noisy quantum processors. *Nature Physics*, 19(7):1116–1121, 2023. doi: 10.1038/s41567-023-02042-2.

A Complete result tables

B Hardware sim-to-real study

Section 5.3 validates the ZNE failure mode on `ibm_marrakesh` and notes that a purely twin-trained neural mitigator shows a simulation-to-hardware gap. We give the full picture here—first the gap under the hardest, no-device-data setting, then the small amount of on-device data that closes it—because both halves are informative about when and how learned mitigation can be deployed.

Our hardware protocol is sim-to-real by construction: we read the device calibration through the runtime `Target` (gate errors, T_1/T_2, readout), build an Aer noise model from it with `NoiseModel.from_backend`, train the budget-conditioned model on that digital twin, and test on

Table 5: Mean error reduction vs. unmitigated execution (%) for every cell of the scaling study. 150 test instances per cell (100 at n=20); per-execution shots S=8192 ($n \leq 10$), 1024 (n=12, 16), 512 (n=20). Asterisks mark cells where the NEM improvement is not significant at $p < 0.05$ (paired t-test).

Regime	Method	n=4	n=6	n=8	n=10	n=12	n=16	n=20
Stochastic device	NEM (ours)	+22	+15	+10	+13	+11	+7*	+11
	Direct prediction	−927	−631	−605	−395	−365	−226	−180
	ZNE-Richardson	−32	−34	−46	−29	−160	−102	−144
	ZNE-exponential	+21	−87	−35	+7	−894	−206	−1152
	ZNE-adaptive	+20	+21	+15	$< -10^4$	−9970	−6546	$< -10^4$
	CDR	+65	+73	+74	+78	+44	+55	+46
Stochastic-correlated	NEM (ours)	+68	+66	+66	+67	+64	+62	+61
	Direct prediction	−302	−186	−156	−120	−74	−57	−7
	ZNE-Richardson	−8	−8	+6	−3	−38	−33	−42
	ZNE-exponential	−188	+3	−15	−27	−38	−120	−166
	ZNE-adaptive	$< -10^4$	+8	−839	−198	$< -10^4$	−2672	−8
	CDR	+90	+90	+93	+88	+78	+81	+69
Coherent miscalibration	NEM (ours)	+72	+77	+80	+85	+76	+80	+77
	Direct prediction	−244	−188	−129	−57	−69	−3	−0
	ZNE-Richardson	−29	−11	−10	−12	−17	−6	−29
	ZNE-exponential	−104	−24	−70	−281	−87	−464	−32
	ZNE-adaptive	+12	$< -10^4$	−6204	−196	−6250	$< -10^4$	−6604
	CDR	+85	+83	+88	+92	+79	+81	+79

the physical processor. This is the hardest deployment setting—no device data enters training—and it exposes a clean distribution shift. Table 8 reports the outcome. The twin's depolarizing channels overstate the device's true bias, so the model learns a correction that is too aggressive: on the near-shot-noise four-qubit circuits it overshoots the (already small) error, scoring −3% at two layers and −94% at five, where the deeper circuit's larger learned correction overshoots more.

Two considerations qualify the negative result. The gap is a property of the *twin*, not the method: in simulation, where training and test share the noise process (the deployment setting CDR and on-device methods also assume), the model is strongly positive (Sections 5.1–5.4). The hardware experiment swaps a matched noise model for a mismatched one, and the model degrades accordingly—the expected behaviour of a supervised corrector under covariate shift. The failure is also bounded in principle: the model emits a residual correction, and the ensemble safeguard of Section 5.2 gates large-disagreement corrections back to the raw value, so a safeguarded deployment would track the unmitigated baseline rather than ZNE's unbounded amplification.

Closing the gap with on-device fine-tuning. The gap closes with a small amount of device data. We measure a fine-tuning set of 120 random four-qubit, five-layer circuits on `ibm_marrakesh` in a single batched job, compute their exact ideal labels by classical statevector simulation, and fine-tune the twin-trained model on these real (noisy, ideal) pairs. We evaluate on 16 held-out instances measured in the same job, so training and test share one calibration epoch—the setting a deployment actually faces. Table 9 reports the outcome. On this draw the twin-only model already improves by +16% over unmitigated execution; fine-tuning lifts it to +39%, and halves the worse-than-unmitigated rate from 50% to 25%. The single best-seed run reaches +56%; we report the more conservative mean over all settings. The gain is stable,

Table 6: Equal-budget sweep at n=8 (100 instances): mean error reduction vs. unmitigated execution (%) when every method spends the same total budget B per evaluation point, split across its required executions.

Regime	Method	B=2^{10}	B=2^{12}	B=2^{14}	B=2^{16}
Stochastic device	NEM (ours)	+11	+17	+20	+19
	ZNE-exponential	−194	−216	−6	+8
	ZNE-adaptive	$< -10^4$	$< -10^4$	$< -10^4$	+10
	CDR	−132	−54	−9	+47
Coherent miscalibration	NEM (ours)	+72	+77	+81	+81
	ZNE-exponential	−471	−36	−93	−129
	ZNE-adaptive	$< -10^4$	−183	+7	−76
	CDR	+23	+42	+68	+84

Table 7: Tail statistics across structured-regime cells: median and maximum absolute error, and the fraction of instances on which each method returns a worse answer than unmitigated execution.

		NEM (ours)			ZNE (best of three)			CDR		
Cell	Raw MAE	med	max	worse	med	max	worse	med	max	worse
Coherent miscal., n=10	0.0412	0.0044	0.03	7%	0.0432	0.15	63%	0.0025	0.01	6%
Coherent miscal., n=12	0.0397	0.0080	0.03	9%	0.0399	0.17	59%	0.0073	0.02	6%
Coherent miscal., n=16	0.0442	0.0079	0.03	11%	0.0419	0.14	49%	0.0078	0.02	10%
Coherent miscal., n=20	0.0436	0.0082	0.03	12%	0.0504	0.19	61%	0.0087	0.03	7%
Coherent miscal., n=4	0.0436	0.0115	0.04	14%	0.0349	0.16	38%	0.0046	0.02	4%
Coherent miscal., n=6	0.0390	0.0072	0.04	12%	0.0388	0.12	57%	0.0047	0.02	16%
Coherent miscal., n=8	0.0437	0.0074	0.03	10%	0.0430	0.14	54%	0.0043	0.01	4%
Stochastic-corr., n=10	0.0345	0.0090	0.04	11%	0.0346	0.11	50%	0.0037	0.01	4%
Stochastic-corr., n=12	0.0348	0.0105	0.07	13%	0.0350	1.82	49%	0.0055	0.02	10%
Stochastic-corr., n=16	0.0349	0.0128	0.06	17%	0.0401	0.15	61%	0.0059	0.02	6%
Stochastic-corr., n=20	0.0360	0.0119	0.07	19%	0.0377	0.10	52%	0.0092	0.04	10%
Stochastic-corr., n=4	0.0395	0.0108	0.04	15%	0.0360	0.11	55%	0.0045	0.01	8%
Stochastic-corr., n=6	0.0386	0.0115	0.04	15%	0.0355	0.08	44%	0.0030	0.01	0%
Stochastic-corr., n=8	0.0389	0.0109	0.06	16%	0.0340	0.17	46%	0.0023	0.01	2%

not a lucky configuration: across 18 learning-rate/epoch/encoder-freezing settings, each run over five seeds (90 runs), every configuration mean is positive (range +27% to +41%) and the worst individual seed still reduces error by +20%. A two-parameter affine recalibration of the twin's correction, $\langle H\rangle_{\text{pred}} = \langle H\rangle_{\text{noisy}} + a\,(\langle H\rangle_{\text{twin}} - \langle H\rangle_{\text{noisy}}) + b$ fitted on the same device set, reaches +45% with $a = 0.77 < 1$—a direct measurement that the twin overcorrects, exactly as the negative result diagnosed.

We are explicit about the scope of this correction: it is specific to a calibration epoch. The twin-only and fine-tuned models above are compared *within* one epoch—training and held-out circuits measured in a single job—which is the comparison a deployment faces. Across epochs the fit does not transfer: applying this epoch's fine-tuned model to the earlier five-layer instances of Table 8 (recomputed from their recorded measurements; the script is in the repository) yields a smaller improvement than the current twin, because the device has drifted and those near-shot-noise circuits leave little headroom. On those same instances the *current* twin checkpoint scores +39.5%, so Table 8's −94% reflects an earlier twin and calibration rather than intrinsic instability

Table 8: Hardware results on `ibm_marrakesh` (4 qubits, equal-budget, 8,192 shots). Improvement is mean error reduction vs. unmitigated; "worse" is the fraction of instances exceeding the unmitigated error. ZNE's failure transfers to hardware; the twin-trained neural model's correction does not.

Depth	Method	Improvement	Worse	MAE
2 layers	ZNE-Richardson	−201%	80%	.038
	NEM (twin-trained)	−3%	60%	.013
5 layers	ZNE-Richardson	−314%	88%	.055
	NEM (twin-trained)	−94%	88%	.026

Table 9: On-device fine-tuning on `ibm_marrakesh` (4 qubits, 5 layers, 8,192 shots, 16 held-out instances; unmitigated MAE .018). The twin-only model is trained on the digital twin alone; the fine-tuned model adds 120 device-measured circuits with exact classical labels, and all three of its cells (improvement, worse-rate, MAE) are means over 18 settings × 5 seeds, every configuration positive; the affine row recalibrates the twin's correction on the same device set. All rows are evaluated within one calibration epoch (Appendix text). "Worse" is the fraction of instances exceeding the unmitigated error.

Method	Improvement	Worse	MAE
NEM (twin-only)	+16%	50%	.015
NEM (on-device fine-tuned)	+39%	25%	.011
NEM (affine recalibration)	+45%	19%	.010

of the method; we therefore report the within-epoch gain only. This epoch dependence is shared with every device-data method: CDR re-fits on freshly drawn device circuits for each target circuit, paying the cost per evaluation, whereas a fine-tuned model amortizes one device-data fit across an entire evaluation loop and is refreshed only when calibration drifts. A small, classically labelled on-device set is thus enough to turn the learned mitigator from a marginal corrector into a clearly beneficial one within an epoch—the route the negative result pointed to (Liao et al., 2025; Adeniyi and Kumar, 2025).

C Reproducibility details

Data generation. Each scaling cell uses 4,000 training and 400 validation pairs (2,500/250 at n=20), with circuit parameters drawn uniformly from $[0, 2\pi)$ and noise parameters drawn per sample from the regime ranges of Section 4. Noisy inputs are measured estimates at the per-size shot budget; ideal labels are exact statevector expectation values.

Training. AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 64, Huber loss, cosine annealing with warm restarts (T_0=25), 75 epochs, gradient clipping at norm 1. The best-validation checkpoint is selected. Three seeds are trained per cell; the best-validation seed is reported and the seed ensemble feeds the safeguard analysis.

Baselines. mitiq 1.0 with global unitary folding at scale factors $\{1, 2, 3\}$ (adaptive: four steps, scale factor 2, depolarizing asymptote). CDR: 31 near-Clifford training circuits, fraction of non-

Clifford gates retained 0.5, target circuits transpiled to the $\{R_Z, \sqrt{X}, X, \mathrm{CNOT}\}$ basis with an exact expectation-equivalence check. Every baseline execution passes through the same measured-shot estimator and, in the miscalibration regime, the same per-snapshot circuit transform as the raw measurement.

Statistics. Improvements are reported against the unmitigated estimate on the same instances; significance uses two-sided paired t-tests on absolute errors. Medians and interquartile ranges are reported wherever extrapolation tails dominate means.