Benchmarking Quantum Error Mitigation in Large-Scale Circuits

STUDENTS: Aurelia Meiqi Brook, Roy Irving, Rebecca Munk

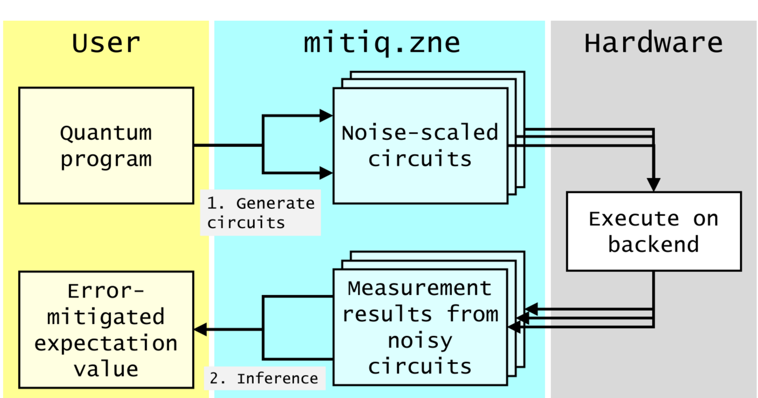
Quantum Error Mitigation (QEM)

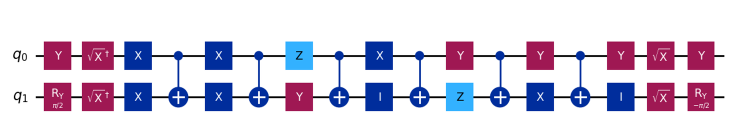
QEM techniques modify quantum input circuits and apply classical post-processing to mitigate errors in measurement outcomes. This allows for improvements in Noisy Intermediate-Scale Quantum (NISQ) devices without implementing full error correcting codes. QEM methods explored here include: Zero-Noise Extrapolation, Probabilistic Error Cancellation, and Layerwise Richardson Extrapolation. Our goal is to benchmark how well these techniques perform as we increase circuit width and depth.

Mitiq

Mitiq: an open-source Python library designed to implement various QEM techniques.

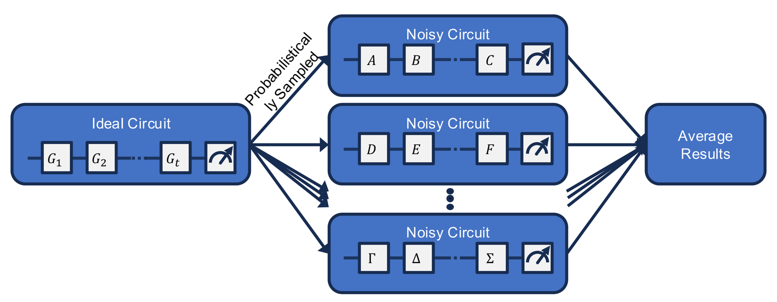
Given a function which inputs a quantum circuit and outputs an expectation value, Mitiq generates an array of circuits for a given QEM method, runs them, and then processes them, returning a mitigated result (upper left). It also has several benchmarking circuits built-in, such as the mirror circuit (lower left) [1].





Probabilistic Error Cancellation (PEC)

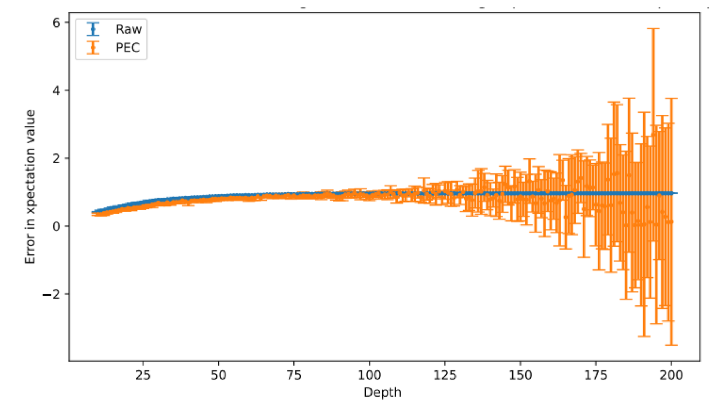
PEC: an error mitigation technique which reconstructs the ideal output of a circuit using a known noise model for the hardware.



PEC Algorithm

1. Represent each ideal operation as a combination of noisy operations with associated probabilities.
2. Use Monte-Carlo sampling to probabilistically select circuits made up of these noisy operations and determine the output of each.
3. Average the results of the previous step over many iterations to determine the optimal result.

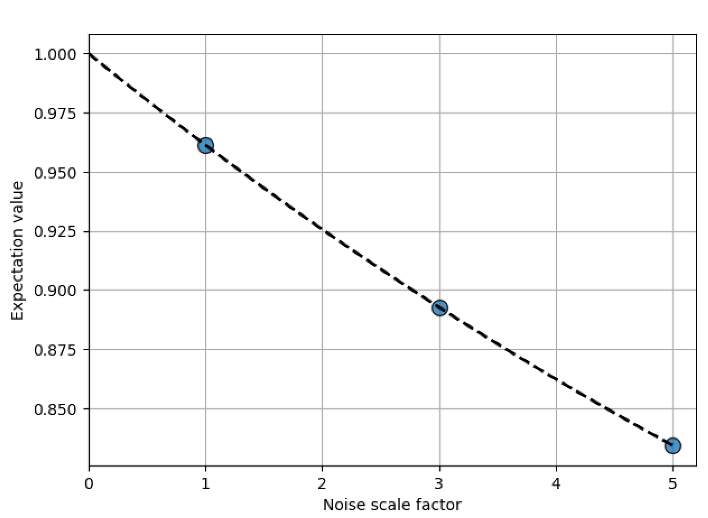
Define "**usefulness cutoff**" point when mitigated value error is larger than raw value error five times in a row. Early cutoffs due to "rough patches" or late cutoffs due to "lucky spots" make this definition imprecise.



"Lucky spot" example: PEC error as a function of depth for a five-qubit mirror circuit on FakeSherbrooke where cutoff reached maximum allotted depth value.

Deeper or wider circuits mean more gates and more possibilities for the sampler to choose from, leading to a greater variance in output values for each run. With a fixed sample number, this leads to larger error bars and a greater change to average to a poor result.

Zero Noise Extrapolation (ZNE)

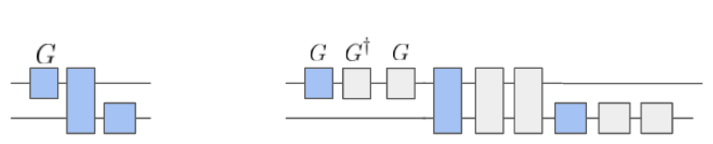


ZNE: an error mitigation technique where noise in the form of additional gates are systematically increased without changing the ideal outcome of a measured observable, allowing users to utilize simple statistical methods to find the "zero-noise" limit.

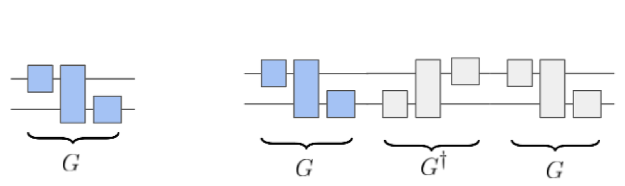
Additional gates are added on via "folding":

Since all quantum gates are unitary matrices, we can add on an arbitrary set of gates $G^\dagger G = I$ many times over without changing the noiseless resulting bitstring. However, on an actual QPU, the application of more gates generates more noise. In this study, we set the noise scale factor $\lambda = 3$.

Local Folding: Each gate is considered individually



Global Folding: All gates are considered collectively before repeating



Layered Richardson Extrapolation (LRE)

Overview:
Layerwise Richardson Extrapolation (LRE) mitigates quantum noise by constructing a multivariate extrapolation from circuits that are folded differently per layer. This allows for richer structure than traditional ZNE, which applies the same scale globally.

Multivariate Basis Construction:
Given a circuit with d layers and extrapolation degree k, the number of monomial terms is:

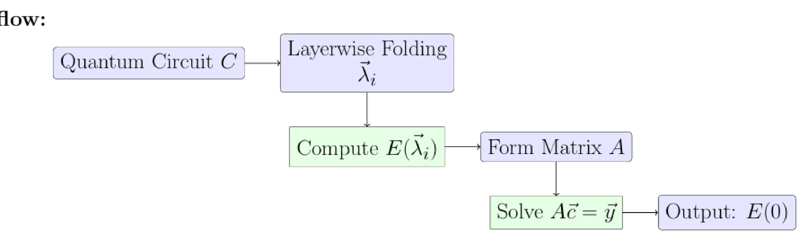
$$\#\text{terms} = \binom{d+k}{k}$$

Each noise-scaled circuit is generated by a fold vector $\vec{\lambda}_i$, and we compute an observable $E(\vec{\lambda}_i)$. These populate:

$$A\vec{c} = \vec{y}, \quad \text{and then} \quad E(0) = \sum_i c_i E(\vec{\lambda}_i)$$

where A is the matrix of basis monomials, $\vec{y}$ is the vector of noisy observables, and $\vec{c}$ are the Lagrange coefficients.

Workflow:



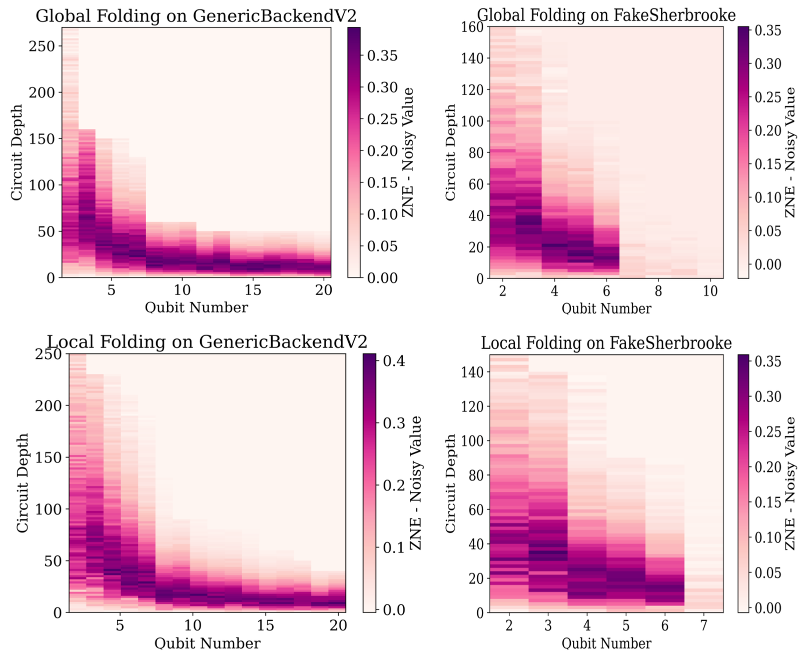
Comparison of Error Mitigation Methods:

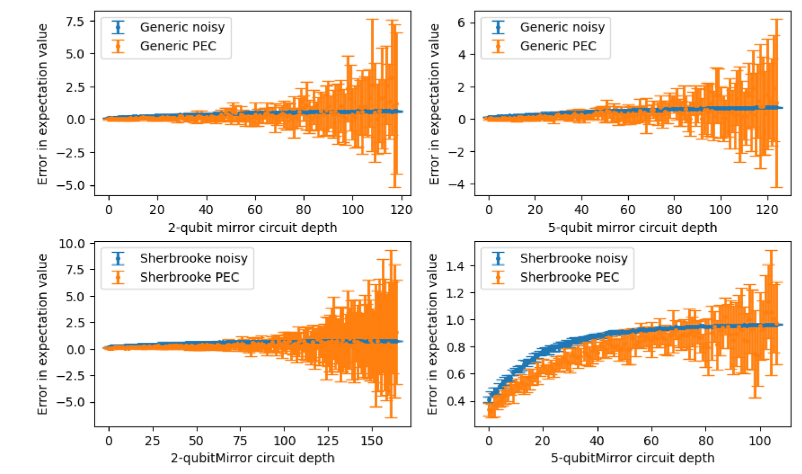
Feature	ZNE (Global)	PEC	LRE (Layerwise)
Noise Scaling Type	Global fold factor	Gate-level inversion	Layerwise fold vectors
Fitting Method	Univariate poly fit	Monte Carlo averaging	Multivariate poly extrapolation
Main Tunable Params	- `scale_factors` - `folding_method` - `degree`	- `noise_model` - `samples_per_gate` - `repetitions`	- `fold_vectors` $\vec{\lambda}_i$ - `num_chunks` - `poly_degree` k
Circuit Overhead	Moderate	High	Moderate–High
Backend Requirements	Minimal	Requires known noise	Minimal (structured folding)
Strengths	Simple, interpretable	Accurate for known noise	Captures detailed structure
Limitations	Ignores layer-specific errors	Sampling-intensive	Risk of overfitting

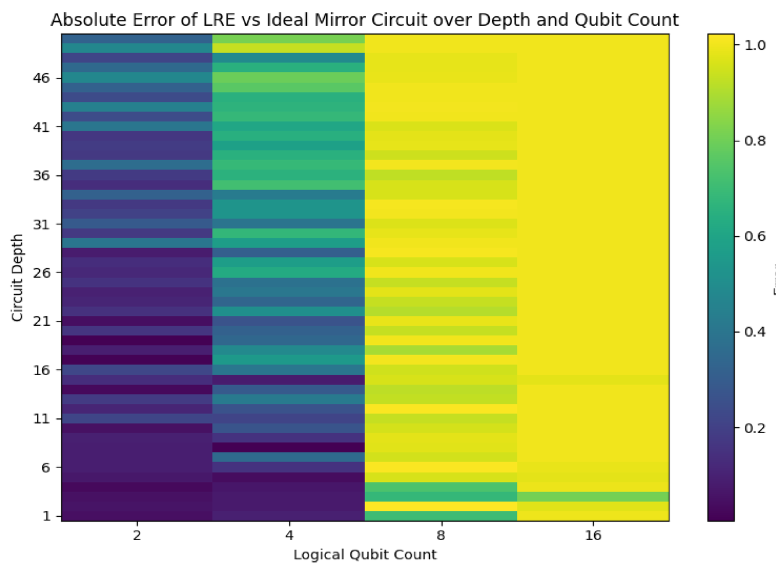
Comparison of ZNE, PEC, and LRE including configurable benchmarking parameters.

Results

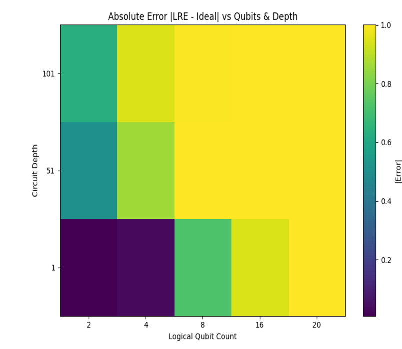
- ZNE: Performance of ZNE mitigated result minus unmitigated result
  - Global Folding:
    - Using GenericBackendV2: max width of 20 (on CPU), max depth of 270
    - Using FakeSherbrooke: max width of 9 qubits, max depth of 160
  - Local Folding:
    - Using GenericBackendV2: max width of 20 (on CPU), max depth of 250
    - Using FakeSherbrooke: max width of 7 qubits, max depth of 140
- PEC: Difference between ideal and calculated expectation value:
  - Using GenericBackendV2, depth cutoff:
    - 2 qubits: ~ 120
    - 5 qubits: ~ 130
  - Using FakeSherbrooke, depth cutoff:
    - 2 qubits: ~ 175
    - 5 qubits: ~ 115
  - Usefulness cutoff when variance gets so large that the average doesn't converge due to fixed sample number. Can vary greatly due to "rough patches" and "lucky spots."



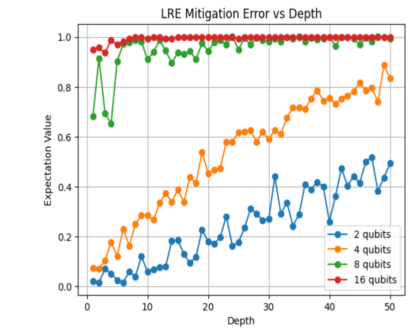




Heatmap (Depth × Qubits, Low-Shot Scan)
Heatmap of absolute LRE error (|LRE–ideal|) across circuit depths (1–150) and logical qubit counts (2, 4, 8, 16), generated with minimal shots (< 500 shots per point). This rapid breadth-first scan on a standard CPU (under 30 mins) identifies where mitigation delivers the greatest benefit.



Heatmap (Depth × Qubits, Low-Shot
Heatmap of |LRE – ideal| error versus LRE chunk count (1 through 20) and circuit depth/qubit combinations, using minimal shots (<500). This scans up to 150 depth and 20 qubits, more shots would be ideal but this was performed, in order to assess where mitigation and measurement doesn't fail completely with minimal local cpu effort.



Depth Curves (Shot-Budget Tradeoff)
Layerwise Richardson extrapolation preserves expectation values for 2, 4, 8, and 16 qubits, but as depth and width grow, the low-shot regime (< 1000 shots) limits initial accuracy. Deeper/wider circuits therefore require higher shot budgets to sustain reliable values.

Future Work and References

- Compare performance on real quantum hardware.
- Fully streamline notebooks for future use.
- Test other benchmarking circuits.

[1] R. LaRose et. al. Mitiq: A software package for error mitigation on noisy quantum computers. Quantum, 6:774, Aug. 2022.
[2] Z. Cai et. al. Quantum error mitigation. Reviews of Modern Physics, 95(4), Dec. 2023.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: Nate Stemen, Sara Mouradian
SPONSOR: Unitary Foundation