Measuring Performance of Quantum Hardware and Algorithms, Error Correction and Error Mitigation

Rajesh Sathya Kumar, Andreas Spanias, Glen Uehara
SenSIP Center, School of ECEE, Arizona State University, Tempe, AZ 85287, USA

Abstract— **Quantum Computing is evolving at a pace that was never seen before. Corporation and Government realize the immense potential in the technology to radicalize the world we live in. This paper provides a brief survey on the various performance metrics available to measure performance of**

Quantum Hardware and Quantum Algorithms. Then, We will discuss the various error correction methods that provides a way of realizing coherent logical qubits at the hardware level and discuss the error mitigation techniques to mitigate errors at the Algorithmic Implementation Level. Finally, We showcase the performance metrics from different Quantum Hardwares using implementations of Basic Algorithms that have significant real world applications.

Keywords— ***Quantum Computing, Quantum Performance, Quantum Volume, Error Correction, Error Mitigation***

I. INTRODUCTION

Based on the experiments from the classical complexity theory, decision problems such as NP (Non-Deterministic Polynomial Time), NP-Hard and NP-Complete are believed impossible to completely solve in deterministic polynomial time for all possible inputs using classical computers. With the recent advances in Quantum Algorithms, a handful of them have been theoretically shown to solve such decision problems in polynomial time if we used a quantum computer. Such problems are called Bounded Error, Quantum Polynomial Time (BQP) Problems.

The holy grail of scientists and engineers working in this field, is to build a fault-tolerant and scalable Quantum Hardware that could show a ”Quantum Advantage” of a real world application. By ”Quantum Advantage”, we mean that, we have a Quantum Hardware that could solve a real world problem in polynomial-time which is impossible to solve in polynomial-time using our classical computers. Although, the Quantum Computing industry is getting a lot of attention in recent times from various entrepreneurs, governments, and tech leaders due to its immense potential to raise the bar in terms of computational power and scalability over its classical counterparts, the existing NISQ-era Quantum Computers are far from being fault-tolerant.

The theoretical Physicist David P. DiVincenzo laid out the necessary conditions to construct a Quantum Computer. The conditions are given below:

1) A scalable physical system with the well characterized elementary units of computation, Qubits. These qubits can be any physical two-level quantum system which can have the orthogonal states $|0\rangle$ and $|1\rangle$.
2) Ability to initialize the qubits in the simple, fiducial state
3) Long relevant decoherence times, which makes them useful to be operated or entangled without any loss of information
4) A Quantum single-qubit and two qubit gate-set must be universal (If the hardware is built on circuit model quantum computation) and have high fidelity.
5) The ability for qubit measurement or read-out.

II. FUNCTIONS OF A QUANTUM SIMULATOR

Quantum simulators [1] are software programs that run on classical computers and act as the target machine, making it possible to run and test quantum programs in an environment that predicts how qubits will react to different operations. The quantum simulator is responsible for providing implementations of quantum operations for an algorithm. This includes primitive operations such as H, CNOT, and Measure, as well as qubit management and tracking. The Quantum Development Kit includes different classes of quantum simulators representing different ways of simulating the same quantum algorithm. I have listed below some of the Quantum Simulators available in the industry and are accessible through cloud.

- IBM’s Qiskit
- Google’s Cirq
- Amazon’s AWS Bracket
- Microsoft’s Q# and Azure Quantum
- Rigetti’s Forest
- Xanadu’s Pennylane

III. QUANTUM COMPUTING ROADMAP

According to google, this is the roadmap of the Quantum Information research for this decade. Currently, most of the research is happening in the first 2 points

- Implement Error Correction
- Show Error Correction gets better with more qubits
- Make 1 logical qubit with endless coherency
- Make 2 logical qubits with 2-qubit operations
- Tile Thousands of logical qubits

IV. QUANTUM SIMULATOR PERFORMANCE METRIC TERMINOLOGIES

A. Circuit Depth

The length of the longest path from the input to output or the minimum amount of time taken to execute the circuit (assuming every gate operation are performed in same time-step) [2]

B. Circuit Width

The circuit width is the total number of input qubits and bits used in the circuit. [2]

C. Quantum Volume

It is defined as the average performance on a set of random circuits. If a quantum computer can execute an algorithm successfully (error-free) with n qubits, its quantum volume is 2^n. [2]

D. Quantum Circuits and Programs

A quantum circuit is a computational routine consisting of an ordered sequence of quantum operations including gates, measurements, and resets on quantum data (qubits) and concurrent real-time classical computation. Data flows between the quantum operations and the real-time classical compute so that the classical compute can incorporate measurement results and the quantum operations may be conditioned upon or parameterized by data from the real-time classical compute. Here real-time means within the coherence time of the
qubits. [2]

E. Circuit Layer Operations per second (CLOPS)

Circuit Layer Operations per Second (CLOPS) is a measure correlated with how many QV circuits a QPU can execute per unit of time. That simple statement hides a wealth of choice about the possible circuit families and the execution context. Here, we pursue a holistic speed benchmark of a typical application. To faithfully model real-world use, we deem it essential to capture interaction time with the run-time environment that invokes the circuits. This attempts to avoid a pitfall seen in some synthetic benchmarks that characterize classical systems by their instruction clock rate without considering the effects of data transfers between CPU, cache, and main memory. [2]

V. Performance Metrics

Currently, the following performance metrics are used to evaluate a Quantum Computers [1]

1. Scale
2. Quality
3. Speed

- Scale: Measured by the number of qubits. Indicates the amount of information we can encode in a quantum system
- Quality: Measured by Quantum Volume which indicates quality of circuits and how faithfully circuits are implemented in hardware
- Speed: Measured by CLOPS (Circuit Layer Operations per Second) which indicates how many circuits can run on hardware in each time.

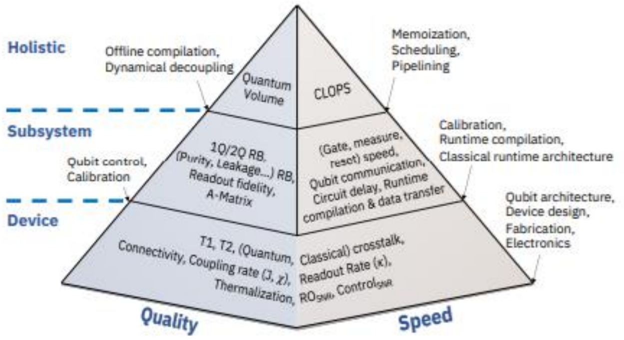

FIG. 1. Benchmarking pyramid showing how quality and speed can be benchmarked. Higher-level benchmarks capture more complexity but less specificity. There may be tradeoffs between the two faces of the pyramid [1].

VI. Quantum Noise and Decoherance

Noise describes all the things that can cause the quantum computers to malfunction. The source of quantum noise includes electromagnetic signals, earth's magnetic field, due to which the qubit states inherently change slightly from its original state [4]

This change in the qubit state due to environmental factors is called decoherence of qubits

VII. Quantum Error Correction

A. Barriers to Quantum Error Correction

1. Measurement of error destroys superpositions.
2. No-cloning theorem prevents repetition.
3. Must correct multiple types of errors (e.g., bit flip and phase errors).
4. How can we correct continuous errors and decoherence?

B. Types of Quantum Errors

There are the common types of errors that occur in quantum computers.

1. Bit Flip Errors
2. Phase Flip Errors
3. Complete Dephasing Errors
4. Rotation Errors

Bit Flip X: $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$
Phase Flip Z: $Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$
Complete dephasing: $\rho \rightarrow (\rho + Z\rho Z^{\dagger})/2$ (decoherence)
Rotation: $R_\theta|0\rangle = |0\rangle$, $R_\theta|1\rangle = e^{i\theta}|1\rangle$

A distance measure [7] quantifies the extent to which two quantum states behave in the same way. While these distance measures are usually given by certain mathematical expressions, they often possess a simple operational meaning, i.e., they are related to the problem of distinguishing two systems. These distance metrics are used to compare two qubit states to quantify the errors occurring in the system

C. *Bit flip error correction / Error Correction via Repetition*

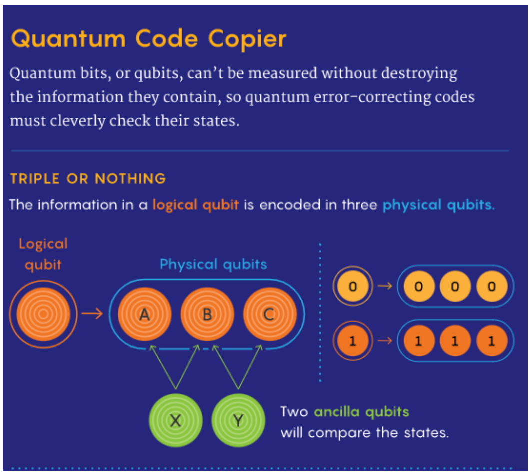

FIG. 2. Bit Flip Error Correction Algorithm

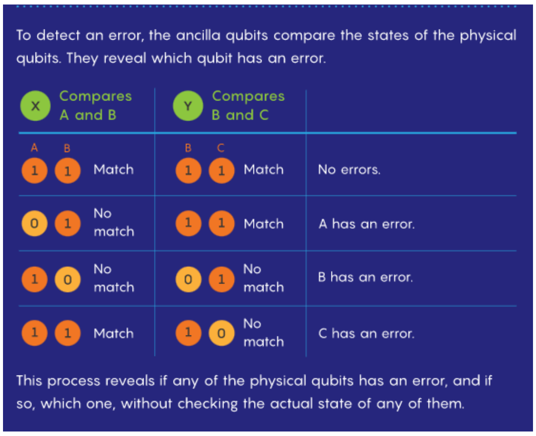

FIG. 3. Bit Flip Error Correction Qubit Comparison

Below is the circuit model for achieving the error corrected logical qubit via repetition

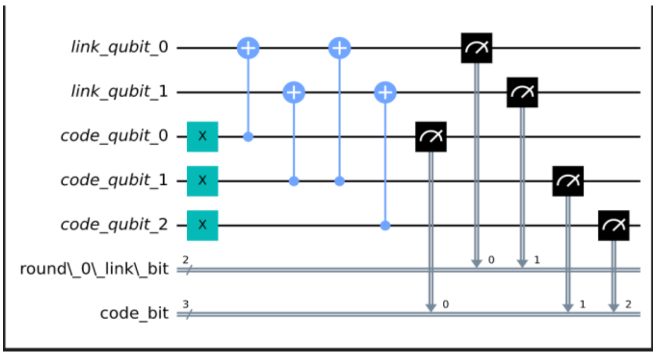

FIG. 4. Bit Flip Error Correction Qiskit Implementation

Code distance vs ln(Logical error probability) is given below

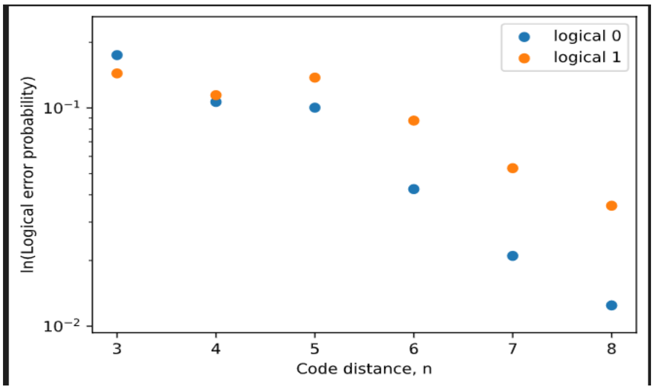

FIG. 5. Code distance vs ln(Logical error probability)

D. *Phase flip Errors*

Hadamard transform H exchanges bit flip and phase errors:

$H(\alpha|0\rangle + \beta|1\rangle) = \alpha|+\rangle + \beta|-\rangle$
$X|+\rangle = |+\rangle$, $X|-\rangle = -|-\rangle$ (acts like phase flip)
$Z|+\rangle = |-\rangle$, $Z|-\rangle = |+\rangle$ (acts like bit flip)

Repetition code in Hadamard basis corrects a phase error.

$\alpha|+\rangle + \beta|-\rangle \rightarrow \alpha|000\rangle + \beta|000\rangle$

E. *Bit and Phase flip error correction / Nine Qubit Codes*

To correct both bit flips and phase flips, use both codes at once:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha(|000\rangle + |111\rangle)^{\otimes 3} + \beta(|000\rangle - |111\rangle)^{\otimes 3}$$

Repetition 000, 111 corrects a bit flip error, repetition of phase +++, --- corrects a phase error. This code corrects a bit flip and a phase, so it also corrects a Y error:

$Y = iXZ$: $Y|0\rangle = i|1\rangle$, $Y|1\rangle = -i|0\rangle$ (Global Phase irrelevant)

F. *Correcting Continuous Rotations*

How does error correction affect a state with a continuous rotation on it? [8]

$$R_\theta^{(k)}|\psi\rangle = \cos(\theta/2)|\psi\rangle - i\sin(\theta/2)Z^{(k)}|\psi\rangle$$
$$\cos(\theta/2)|\psi\rangle|I\rangle - i\sin(\theta/2)Z^{(k)}|\psi\rangle\ |Z^{(k)}\rangle \text{ (Error Syndrome)}$$

Measuring the error syndrome collapses the state:

Prob. $\cos^2(\theta/2)$: $|\psi\rangle$ (no correction needed)
Prob. $\sin^2(\theta/2)$: $Z^{(k)}|\psi\rangle$ (corrected with $Z^{(k)}$)

G. *Correcting All Single-Qubit Errors*

Theorem: If a quantum error-correcting code (QECC) corrects errors A and B, it also corrects $\alpha A + \beta B$. Any 2x2 matrix can be written as $\alpha I + \beta X + \gamma Y + \delta Z$. [8]

A general single-qubit error $\rho \to \Sigma\ A_k\ \rho\ A_k^\dagger$ acts like a mixture of $|\psi\rangle \to A_k\,|\psi\rangle$, and A_k is a 2x2 matrix.

Any QECC that corrects the single-qubit errors X, Y, and Z (plus I) corrects every single-qubit error.

Correcting all t-qubit X, Y, Z on t qubits (plus I) corrects all t-qubit errors.

H. *Shor Code Implementation*

Shor code is used in error correction of both the bit flip errors and phase flip errors. [10]

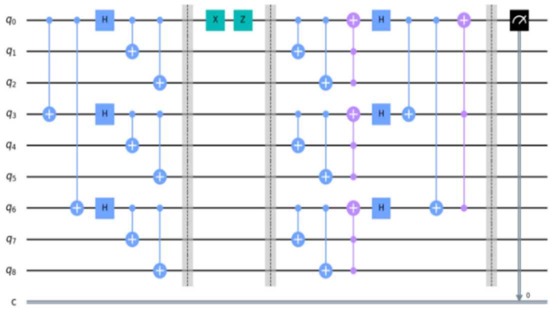

FIG. 6. Circuit Diagram of the Shor Code

VIII. Algorithmic performance Metrics

A. *Quantum Volume Simulation*

The following algorithms are simulated and measured for Quantum Volume. These are the results. [12]

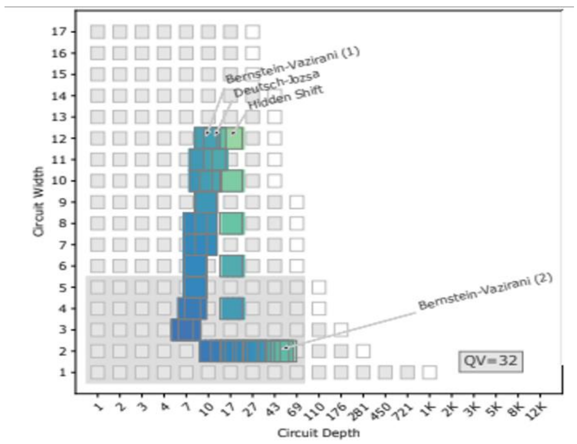

FIG. 7. Quantum Volume Simulation 1

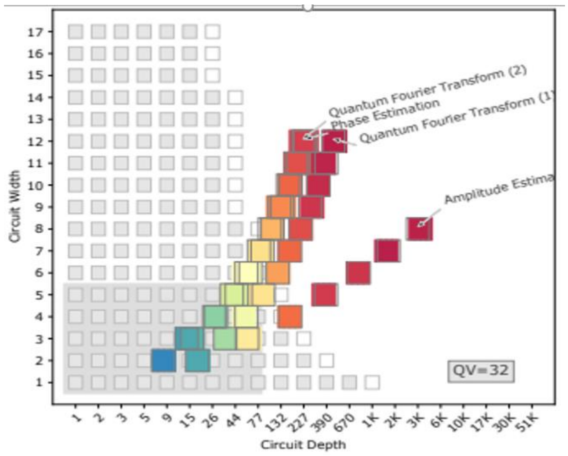

FIG. 8. Quantum Volume Simulation 2

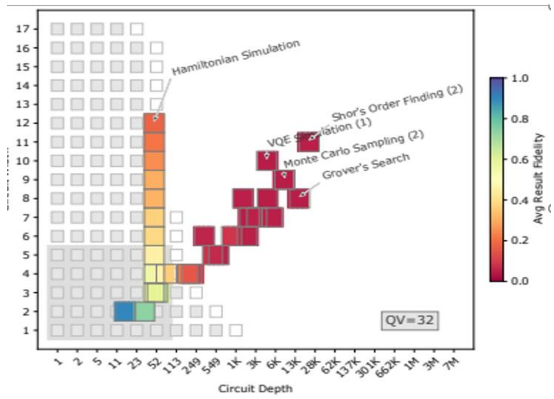

FIG. 9. Quantum Volume Simulation 3

B. *QFT Simulations and evaluating Performance metrics – Results*

Parameters

With Noise model:

Single Qubit Depolarization Error: 0.3%
Two Qubit Error: 3%
Shots = 100

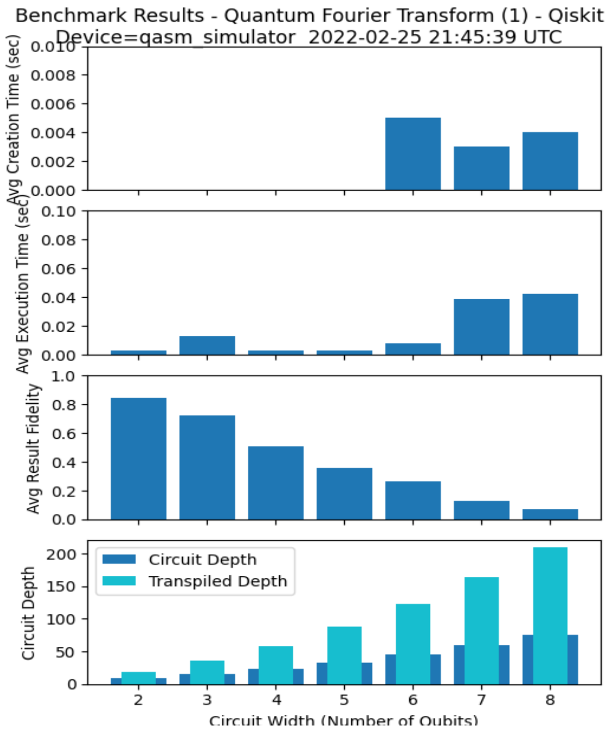

FIG 10. QFT Simulations and evaluating Performance metrics – Results

IX. RANDOMIZED BENCHMARKING PROTOCOL

In quantum computing, the Clifford Group is a set of gates defined by the property that they always transform Pauli's to Pauli's. Randomized benchmarking is a method for assessing the capabilities of quantum computing hardware platforms through estimating the average error rates that are measured under the implementation of long sequences of random quantum gate operations. It is the standard used by quantum hardware developers such as IBM and Google to test the validity of quantum operations, which in turn is used to improve the functionality of the hardware.

Steps for Randomized Benchmarking Protocol:

1. Generate RB sequences
2. Execute the RB sequences (with some noise)
3. Get statistics about the survival probabilities
4. Find the averaged sequence fidelity
5. Fit the results

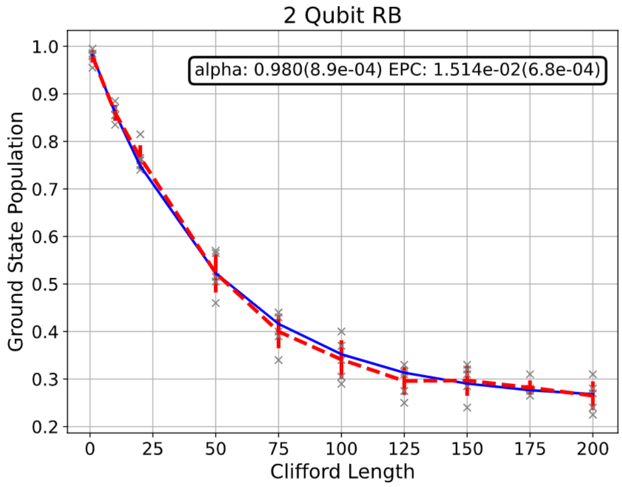

FIG 11. Sample Noise Modelling from randomized benchmarking protocol

X. QUANTUM ERROR MITIGATION

Quantum error correction and mitigation are two approaches to tackle the impact of noise. Whereas error correction requires an enormous qubit overhead, error mitigation is a set of protocols using minimal qubit resources. Quantum error mitigation refers to a series of modern techniques aimed at reducing (mitigating) the errors that occur in quantum computing algorithms. Unlike software bugs affecting code in usual computers, the errors which we attempt to reduce with mitigation are due to the hardware. Below are some the error mitigation techniques.

A. Zero-noise extrapolation

The ZNE method works by assuming that the amount of noise present when a circuit is run on a noisy device is enumerated by a parameter g. Suppose we have an input circuit that experiences an amount of noise g = g0 when executed. Ideally, we would like to evaluate the result of the circuit in the g = 0 noise-free setting.
To do this, we create a family of equivalent circuits whose ideal noise-free value is the same as our input circuit. However, when run on a noisy device, each circuit experiences an amount of noise g = s * g0 for some scale factor s⩾1. By evaluating the noisy outputs of each circuit, we can extrapolate to s=0 to estimate the result of running a noise-free circuit.

The Algorithm achives scaling using a technique called unitary folding. When the scale factor is s=1, the resulting circuit is

$$V=U^\dagger U=I.$$

Hence, the s=1 setting gives us the original unfolded circuit. When s=3, the resulting circuit is

$$VV^\dagger V=U^\dagger UUU^\dagger U^\dagger U=I.$$

In other words, we fold the whole circuit once when =3. Generally, whenever ss is an odd integer, we fold (s−1)/2 times.

The s=2 setting is a bit more subtle. Now we apply folding only to the second half of the circuit, which is in our case given by U†. The resulting partially folded circuit is (U†UU†) U=I

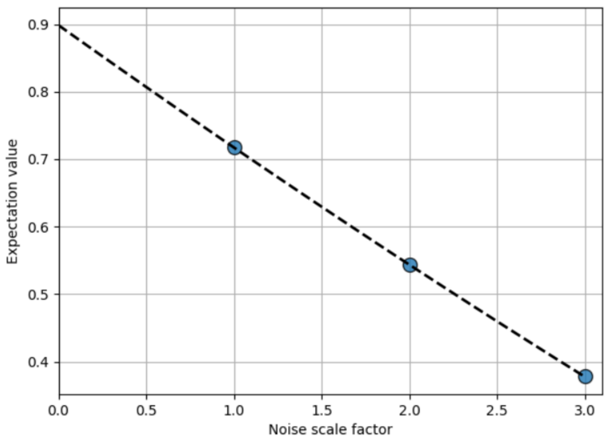

FIG 12. Noise Scaling Factor vs Expectation Value

B. *Clifford Data Regression*

Clifford data regression (CDR) is a learning-based quantum error mitigation technique in which an error mitigation model is trained with quantum circuits that resemble the circuit of interest, but which are easier to classically simulate.

Steps for using Clifford Data Regression:

1. Define a Quantum Circuit
2. Define an Executor
3. Define an Observable
4. Define a Near-Clifford Simulator

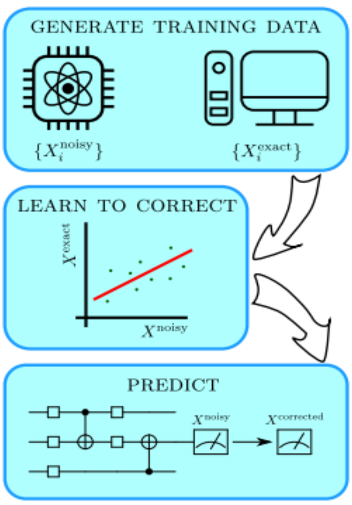

FIG 13. Proposed error mitigation method for Clifford Data Regression

C. *Measurement Error Mitigation*

A simpler form of noise is that occurring during final measurement. At this point, the only job remaining in the circuit is to extract a bit string as an output. For an n qubit final measurement, this means extracting one of the 2^n possible n-bit strings. As a simple model of the noise in this process, we can imagine that the measurement first selects one of these outputs in a perfect and noiseless manner, and then noise subsequently causes this perfect output to be randomly perturbed before it is returned to the user.

Given this model, it is very easy to determine exactly what the effects of measurement errors are. We can simply prepare each of the 2^n possible basis states, immediately measure them, and see what probability exists for each outcome.

Experimental Parameters:

Execution Specs:
QASM Simulator
Noise Model: 1% single qubit error
n = 2 Qubits

Qiskit API:
complete_meas_cal, CompleteMeasFitter

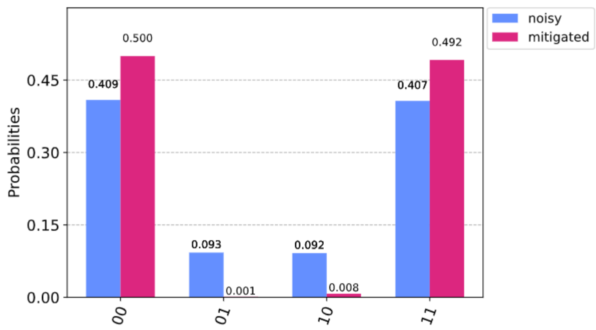

FIG 14. Measurement Error Mitigation – Qiskit with 2 qubits

REFERENCES

[1] T. Proctor, K. Rudinger, K. Young, E. Nielsen, R. Blume-Kohout "Measuring the Capabilities of Quantum Computers", arXiv:2008.11294 (2020), https://arxiv.org/abs/2008.11294

[2] K. Temme, S. Bravyi, J. M. Gambetta "Error Mitigation for Short-Depth Quantum Circuits", Phys. Rev. Lett. 119, 180509 (2017), https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.119.180509

[3] Y. Li, S. C. Benjamin "Efficient Variational Quantum Simulator Incorporating Active Error Minimization", Phys. Rev. X7, 021050 (2017), https://journals.aps.org/prx/abstract/10.1103/PhysRevX.7.021050

[4] Giurgica-Tiron, Tudor, Yousef Hindy, Ryan LaRose, Andrea Mari, and William J. Zeng. "Digital zero noise extrapolation for quantum error mitigation." In 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 306-316. IEEE, 2020.

[5] Zhang, Shi-Xin, Zhou-Quan Wan, Chang-Yu Hsieh, Hong Yao, and Shengyu Zhang. "Variational Quantum-Neural

Hybrid Error Mitigation." arXiv preprint arXiv:2112.10380 (2021).

[6] T. Proctor, K. Rudinger, K. Young, E. Nielsen, R. Blume-Kohout “Measuring the Capabilities of Quantum Computers”, arXiv:2008.11294 (2020).

[7] K. Temme, S. Bravyi, J. M. Gambetta “Error Mitigation for Short-Depth Quantum Circuits”, Phys. Rev. Lett. 119, 180509 (2017).

[8] Y. Li, S. C. Benjamin “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”, Phys. Rev. X 7, 021050 (2017).

[9] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, W. J. Zeng “Digital zero noise extrapolation for quantum error mitigation”, IEEE International Conference on Quantum Computing and Engineering (2020).

[10] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R.

[11] Ozeri, and D. J. Wineland, “Realization of quantum error correction,” Nature, vol. 432, no. 7017, pp. 602–605, 2004.

[12] J. ROFFE, “The Coherent Parity Check Framework for Quantum Error Correction,” Durham University, 2019.

[13] 1. A. M. Steane, “A tutorial on quantum error correction,” in Proceedings of the International School of Physics “Enrico Fermi,” 2006, vol. 162, pp. 1–32.

[14] C. H. Bennett, “Quantum Information: Qubits and Quantum Error Correction,” in International Journal of Theoretical Physics, Feb. 2003, vol. 42, no. 2, pp. 153–176.

[15] R. Vathsan, *Quantum Error Correction*. Cambridge university press, 2020.

[16] Roffe, Joschka. "Quantum error correction: an introductory guide." Contemporary Physics 60, no. 3 (2019): 226-245.

[17] T. Proctor, K. Rudinger, K. Young, E. Nielsen, R. Blume-Kohout “Measuring the Capabilities of Quantum Computers”, arXiv:2008.11294 (2020).

[18] K. Temme, S. Bravyi, J. M. Gambetta “Error Mitigation for Short-Depth Quantum Circuits”, Phys. Rev. Lett. 119, 180509 (2017).

[19] Y. Li, S. C. Benjamin “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”, Phys. Rev. X 7, 021050 (2017).

[20] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, W. J. Zeng “Digital zero noise extrapolation for quantum error mitigation”, IEEE International Conference on Quantum Computing and Engineering (2020).

[21] Easwar Magesan, J. M. Gambetta, and Joseph Emerson, Robust randomized benchmarking of quantum processes, https://arxiv.org/pdf/1009.3639

[22] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson, Characterizing Quantum Gates via Randomized Benchmarking, https://arxiv.org/pdf/1109.6887

[23] A. D. C'orcoles, Jay M. Gambetta, Jerry M. Chow, John A. Smolin, Matthew Ware, J. D. Strand, B. L. T. Plourde, and M. Steffen, Process verification of two-qubit quantum gates by randomized benchmarking, https://arxiv.org/pdf/1210.7011

[24] Jay M. Gambetta, A. D. C´orcoles, S. T. Merkel, B. R. Johnson, John A. Smolin, Jerry M. Chow, Colm A. Ryan, Chad Rigetti, S. Poletto, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen, Characterization of addressability by simultaneous randomized benchmarking, https://arxiv.org/pdf/1204.6308

[25] David C. McKay, Sarah Sheldon, John A. Smolin, Jerry M. Chow, and Jay M. Gambetta, Three Qubit Randomized Benchmarking, https://arxiv.org/pdf/1712.06550

[26] Czarnik, P., Arrasmith, A., Coles, P. J. & Cincio, L. Error mitigation with clifford quantum-circuit data. Preprint at https://arxiv.org/abs/ arXiv:2005.10189 (2020).

[27] Benchmarking neural networks for quantum computation, N.H. Nguyen, E.C. Behrman, M.A. Moustafa, J.E. Steck, https://arxiv.org/abs/1807.03253