Qontrast: Contrast Filters Mitigate Quantum Noise

Wladimir Silva, Frank Mueller
North Carolina State University
Raleigh NC, USA
wsilva@ncsu.edu, fmuelle@ncsu.edu

***Abstract*—We propose a novel method to mitigate measurement errors of quantum computations using image contrast filters, termed QONTRAST (Quantum Contrast filters). QONTRAST provides a general noise mitigation method at much lower cost and without requiring known expectation values than prior work. We experimentally evaluate QONTRAST for various benchmarks from prior work. Our results show fidelity improvements up to 18% for non-variational circuits like GHZ and Bernstein-Vazirani, and up to 5% for QAOA in SupermarQ benchmarks. Furthermore, hardware results in the state preparation of the Kagome lattice using VQE show a reduction of 22% in error rates. QONTRAST is platform agnostic across quantum device technology. Overall, results show that QONTRAST outperforms standard quantum resilience mechanisms of the Qiskit runtime, namely Twirled Readout Error eXtinction (TREX) and Zero Noise Extrapolation (ZNE). The source code of QONTRAST is available for download [1].**

I. INTRODUCTION

Quantum computing has the potential to break algorithmic complexity classes. A given NP-complete problem for classical computing, i.e., of exponential classical complexity, may have only polynomial complexity for a given quantum algorithm like Shor's factorization [23], [36], [12], but they require next-generation Fault-Tolerant Quantum Computing (FTQC) to eliminate noise in software with error correction techniques at the quantum level at 2-3 orders of magnitude more qubits than today's devices. Recent work has focused on simulation [24], [5], [8], [37], [3], compilation [22], [21], [7] and device execution [26], [27], [19].

Quantum noise mitigation (QNM) is an active area of research in the NISQ age that allows the error of measurements to be reduced on today's quantum devices. These techniques could also prove to pave a path on the way to FTQC with many recent result (e.g., [5], [10]). Commonly used QNM techniques include M3 [18] utilized by IBM and a number of techniques bundled in Mitiq, namely Zero Noise Extrapolation (ZNE) [6], [11], [33], Probabilistic Error Cancellation (PEC) [33], and Clifford Data Regression (CDR) [2] among others. Recent work also investigates machine learning as a basis for QNM [20]. The M3 method relies on a number of techniques ranging from a set of reduced calibration matrices to solvers for non-symmetric linear systems when calibration matrices become too expensive. Their twirled readout error extinction (T-Rex) technique adds an additional measure to its calibration circuits to estimate errors [35]. ZNE, on the other hand, is specialized to problems where the expectation outcome is know. Its backward-extrapolation of measurements from circuits with different noise levels allows a zero-noise projection onto the base plain. However, without known expectation values, ZNE cannot be applied. Even when these values are known will ZNE require a set of circuits with increasing noise to be synthesized and then run, which is more costly than running a single circuit.

This work introduces contrast filters as another method to reduce noise in quantum computing. An implementation of the technique, QONTRAST, is compared to the state-of-the techniques in both fidelity improvements and overhead for a variety of benchmarks. Results indicate that QONTRAST outperforms M3/T-Rex, ZNE and PEC while imposing the lowest computational overhead of them all.

QONTRAST makes the following contributions:

- Contrast filters are developed as a generalized technique to mitigate noise in quantum computing with low classical overhead compared to other QNM methods.
- Experiments on multiple quantum devices of different families compare QONTRAST to state-of-the-art QNM techniques. Benchmarks improve fidelity by up to 18% for non-variational circuits and up to 5% for QAOA.
- Error is reduced by 22% beyond the standard quantum resilience methods in a real hardware experiment (Kagome Lattice using VQE).

II. BACKGROUND AN RELATED WORK

The source of noise in quantum devices is due to an imperfect operational environment. Superconducting quantum devices do not operate at a temperature of absolute zero, ion traps hold isotopes in a chamber that does not provide a perfect vacuum, etc. These imperfections result in different error types in quantum computing, such as decoherence, depolarization, state preparation, measurement, cross-talk and thermal relaxation errors. To address these problems, a number of noise mitigation techniques have been developed in the NISQ age. This section briefly describes the most closely related work in this area.

Mitiq: Mitiq [32] combines key noise mitigation techniques. Its principle technique is zero-noise extrapolation (ZNE) [6], [11], [33]. Given an expectation value for a given circuit, ZNE generates a variety of circuit candidates. Their results are backwards interpolated to a zero noise plain. Mitiq also features Probabilistic Error Cancellation (PEC) [33]. Given a user-defined executor function, a set of noisy results is obtained from a set of circuits. When composed with a linear combination, unbiased estimates of the ideal expectation values are obtained. Mitiq and IBM's Qiskit Runtime further

provide support for Pauli twirling [35]. Here, 2-qubit gates are transformed by independent single-qubit gates such that noise is increased to obtain multiple estimations of error-mitigated metrics.

IBM's M3 Library: M3 [18] mitigates errors for a subspace of the noisy input bit strings. It includes a matrix-free preconditioned iterative-solution to mitigate errors on numbers of qubits that would otherwise be impractical. The base technique performs subspace reduction. When noise is minimal, the measurement errors over N qubits satisfy $\vec{P}_{noisy} = A\vec{P}_{ideal}$, where p are the probability outcomes and A is a calibration matrix. A can be constructed from N calibration matrices $A^T = S_{N-1} \otimes \cdots S_1 \otimes S_0$, given S_k as the calibration matrix for the k-th qubit with a certain shape [18]. Alternatively, M3 provides a matrix free solution using two methods for the numerical solution of nonsymmetric linear systems:
1. Generalized minimal residual: An iterative method that uses Krylov subspaces to reduce a high-dimensional problem to a sequence of smaller dimensional problems.
2. Biconjugate-gradient-stabilized: An iterative method for the numerical solution of nonsymmetric linear systems with fast and smooth convergence.

This process results in a mitigated set of outcome probabilities. M3 is the default method for Qiskit Runtime resilience level one in the Sampler primitive.

Note that both M3 and T-REx are used in resilience level one of the Qiskit runtime. However, M3 is executed as part of the Sampler primitive whereas T-REx is invoked in the Estimator. Also, the Sampler produces quasi-probabilities whereas the Estimator produces estimation values.

Qiskit Runtime: Qiskit Runtime provides two features used in our experiments: Quantum resilience, the built-in framework for noise mitigation, and state preparation using VQE (Ising models). The resilience levels supported by the runtime's basic primitives, Estimator and Sampler, are shown in Table I. T-Rex, ZNE and PEC are quantum-based mitigation strategies; M3, on the other hand, mitigates result outcomes classically.

TABLE I
QISKIT ERROR MITIGATION, HIGH LEVELS INCUR MORE OVERHEAD [31].

Level	Definition	Estimator	Sampler
0	No mitigation.	None	None
1	Minimal: Readout error mitigation	Twirled Readout Error eXtinction (T-REx)	M3 library
2	Medium: Reduces bias (noise) in estimators, no guarantee for zero bias.	Zero Noise Extrapolation (ZNE)	None
3	Heavy: Theoretically results in zero bias estimators.	Probabilistic Error Cancellation (PEC)	None

III. METHOD DESCRIPTION AND DESIGN

Contrast filters are one of the common methods of image adjustment to improve their quality. Image contrast enhancement refers to accentuation or sharpening of image features so as to make them more useful for visualization or analysis. For a color image, the brightness of a pixel is the mean of its RGB values. $(r+g+b)/3$. This brightness can be increased by simply adding a delta value $\mu+\Delta\mu$ where the delta factor $\Delta\mu$ can be negative to darken the image or positive for lightening the image. A grayscale image, on the other hand, captures the intensity of light in pixels. Intensity values can be treated as floating point numbers ranging from 0-1. We propose a method to use this simple yet powerful technique to mitigate the noise of the histogram data of a multi-shot experiment (running the same circuit repeatedly and measuring each time, as is commonly done in quantum computing) by first mapping its values to probabilities. Next, these probabilities are treated as grayscale pixels. This allows for application of image processing analysis and display techniques with the ultimate goal of mitigating or eliminating the noise. We chose to focus on experimental outcomes at the classical level for two reasons: 1) Average Measurement (readout) noise is almost double the average Control-NOT (CX) gate noise (see Figure 1). 2) In current noisy hardware, mitigation usually takes the lion's share of the work (see metrics collected over 4 months at the bottom of Figure 1). A simple and effective method like QONTRAST drastically lowers this overhead, where 79% time is spent on mitigation, resulting in significant cost savings during the development cycle.

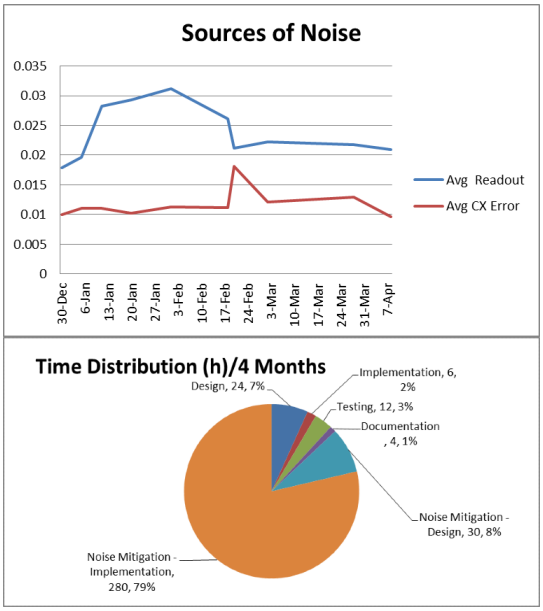

Fig. 1. Metrics over 4 months for VQE problem (Sec. V-E). Top: Avg. readout vs. CX noise levels for 16-qubit device (ibmq_guadalupe). Bottom: Time distribution per project phase, mitigation accounts for 79% of total work-hours.

Consider an arbitrary quantum circuit, e.g., Greenberger-Horne-Zeilinger (GHZ) with a 3-qubit entangled state $|\psi>= \frac{1}{\sqrt{2}}(|000> +|111>)$ shown in Figure 2.

Take the measurement results of the experiment and construct a probability distribution by dividing each result count

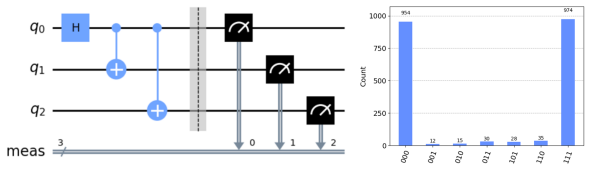

Fig. 2. Left: quantum circuit of a GHZ state; right: experimental result from a run on IBM's 16-qubit noisy simulator FakeGuadalupe.

by the number of "shots" or measurements in the device. Thus, let P_{noisy} be the probability distribution such that

$$P_{noisy} = \{p_1, p_2, ..., p_N\}, N = 2^n, \quad (1)$$

where (n) is the number of qubits in the circuit. We can map those probabilities to pixels in a grayscale image. This allows for image manipulation and display. Next, let $P_{filtered}$ be a new probability distribution such that

$$P_{filtered} = min(p_{max}, max(p_i, p_{min})), p_{min} < p_{max}, \quad (2)$$

where p_{min} is the minimum probability in the distribution, and p_{max} is the maximum. Eq. 2 represents the contrast filter that increases the separation between the darkest (low or noisy) and brightest areas of the image assuming the noisy probabilities are low. The effect can be visualized in Figure 3. Note that after the intensity re-scaling step, the results need to be re-normalized such that

$$P_{mitigated} = \frac{P_{filtered}(i)}{\sum_{i=0}^{N} P_{filtered}(i)}.$$

Finally, map the new distribution into a new set of measurements and compare against the original (right side of Figure 3).

Notice that the filter takes an optional input range argument to either stretch or shrink the intensity of the image. Given such an interval, values outside the interval are clipped to the interval edges. E.g., for an interval of [0, 1], values smaller than 0 become 0, and larger than 1 become 1.

Throughout this manuscript, we treat the input range as a contrast level percentage for simplicity purposes. For example, an input range of [0.1, 0.9] maps to a 10% filter contrast.

The results of the clipping effect are shown on the right side of Figure 3. We observe that the noise has vanished. Next, we test this simple technique against the more elaborate method implemented in the matrix-free measurement mitigation (M3) library by IBM research [18].

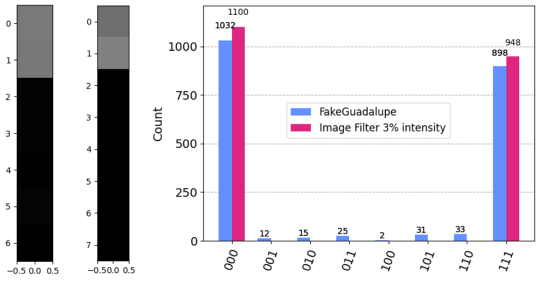

Fig. 3. Visualization of brightness intensity rescaling and the final result for the GHZ state circuit in Figure 2.

IV. Experimental Framework

Experiments are conducted on hardware for a variety of IBM Q devices and in simulation for IBM Q and IonQ noise models specific to their respective devices. Simulations are performed on an Intel-Linux x86/64 (4 vCPUs, 32 GB RAM) with GPU-NVIDIA L4-24GB. We utilize a variety of devices to illustrate the wide applicability of QONTRAST across device families and quantum technologies (see Table II). For each experiment, calibration data specific to runs on hardware devices are reported as they vary day-by-day.

TABLE II
UTILIZED QUANTUM DEVICES: NUMBER OF QUBITS, DEVICE FAMILY AND EXPERIMENTS IN HARDWARE / VIA SIMULATION

Name	Qubits	Family	h/w	sim
IBM Lagos	7	Falcon r5.11H	✓	
IBM Melbourne	14	Falcon r5.11		✓
IBM Guadalupe	16	Falcon r5.11H	✓	✓
IBM Kolkata	27	Falcon r5.11		✓
IBM Cairo	27	Falcon r5.11	✓	✓
IBM Hanoi	27	Falcon r5.11	✓	
IBM Auckland	27	Falcon r5.11H	✓	
IonQ aria-1	25	Ion Trap		✓

We compare results from QONTRAST to M3's T-Rex, ZNE and PEC (see Section II) in terms of probability distributions. The experiments assess results for a number of benchmarks ranging from sampling circuits over quantum resilience and a variational quantum Eigensolver (VQE) to Ising models. The set of benchmarks originates from three sources: The M3 documentation, the Qiskit runtime, and the SupermarQ benchmarks. SupermarQ [34] is a scalable, hardware-agnostic quantum benchmark suite from which we report 8 experiments of different types. Metrics for the initial experiments include:

- T1 (relaxation time): A constant of spontaneous relaxation from state |1> to state |0> in microseconds.
- T2 (dephasing time): A constant of spontaneous change in phase from |+> to |-> in microseconds.
- Readout error: The number of erroneous measurement operations and the significant measurement times, which can lead to the decoherence of quantum states.
- CNOT error rate: A two-qubit randomized benchmark where sequences of CNOT gates are applied to take the qubit on a random walk among certain points on the Bloch sphere and returning it to the initial state. As the number of gates increases, the chance of returning to the initial state drops exponentially and eventually saturates near 50%. The gate error rate is extracted from the fit to this exponential decay.
- Queue wait time: The time the experiment resides on the fair use queue of the respective platform.
- Execution time: Time reported for executing a circuit by Qiskit runtime.

For SupermarQ, we experiment with all of its benchmarks, which are described as follows:

- GHZ: This benchmark represents the GHZ state preparation parameterized by the number of qubits. Device performance is based on the Hellinger fidelity between the experimental and ideal probability distributions.

- Mermin-Bell: Test of a quantum computer's ability to exploit purely quantum phenomena such as superposition and entanglement based on Bell-inequality tests of locality. Performance is based on a quantum device ability to prepare a GHZ state and measure the Mermin operator.
- Bit-Code: Syndrome measurement in a *bit-flip* error correcting code. Device performance is given by the Hellinger fidelity between the experimental results and the ideal distribution. The ideal is known based on the bit state parameter.
- Phase-Code: Syndrome measurement in a *phase-flip* error correcting code. Device performance and the ideal distribution are as defined in the bit-code benchmark above.
- VQE Proxy: Full VQE application that targets a single iteration of the variational optimization. The score is based on the averaged values of the measured energies from these experiments based on how closely the experimental results are to the noiseless values.
- Hamiltonian Simulation: Focuses on the ability to simulate 1D Transverse Field Ising Models (TFIM) of variable length. Device performance is based on how closely the experimentally obtained average magnetization matches the noiseless value.
- QAOA Fermionic (ZZ) Swap Proxy: Implements MaxCut on a Sherrington-Kirkpatrick model for fermions using a qubit routing sequence that can be used to efficiently execute the Quantum Approximate Optimization Algorithm (QAOA). Device performance is given by the Hellinger fidelity between the experimental output distribution and the true distribution obtained via classical simulation.
- QAOA Vanilla Proxy: Implements MaxCut on a Sherrington-Kirkpatrick model as a proxy of a full QAOA. Device performance is given as in the QAOA ZZ-Swap Proxy.

Initial experiments are conducted by following the measurement error mitigation tests described in the M3 library documentation. M3 corrects a reduced subspace of input bit strings from an experiment measurement, resulting in a linear system of equations that is simple to solve. For a small number of input strings the system of equations is solved using LU decomposition (a form of Gaussian elimination). For a large number of input strings the problem is solved in a matrix-free generalized minimal residual or bi-conjugate gradient stabilized method. The process is described in detail in [18], the collection of original tests is available in [15]. Our modified experiments are available as a software artifact [1].

V. Experimental Results

Experiments first focus on analyzing contrast filters via simulation under realistic but controlled noise and second hardware results for a variety of benchmarks.

A. M3: Probability Mitigation

First, the experiment in [29] in simulation provides insight on the feasibility of QONTRAST. This experiment is designed to correct readout errors and transform the outcomes into a true probability distribution. The experiment creates a circuit with 5 entangled qubits (center of Figure 4), performs measurements on all of them, and runs on the 14-qubit noisy simulator called "FakeMelbourne" collecting probability distributions for the raw device and M3. We utilize image intensity mitigation at an input range of 3% (0.03, 0.97).

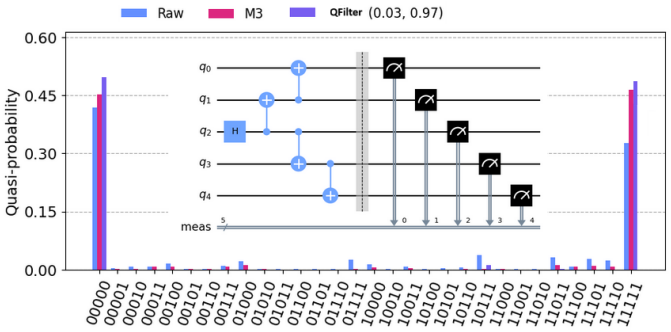

Fig. 4. Experimental results for readout error correction on "FakeMelbourne" to assess the feasibility of QONTRAST.

The expected outcome of the experiment are the all-zero and all-one strings (x-axis) at equal magnitude, which receive the highest probabilities (y-axis) in the bar graphs of Figure 4. Any other results (small bars) are due to quantum noise and should ideally be mitigated by noise reduction techniques. Results are depicted for raw measurements, M3 mitigated results and QONTRAST (ImageFilter). We observe that QONTRAST outperforms M3 by reducing the noisy outcomes as shown in the probability histogram. While the ideal outcome would be a 50% probability for 00000 and 11111, raw and M3 result in low probabilities of other strings as well, which lowers their probabilities on the correct values. QONTRAST does not results in low probabilities of incorrect results such that the two correct strings add up to 100%. Notice that the 00000 string still has higher probability than 11111, which is typical for noisy quantum devices as the ground state, zero, is stable whereas the excited state, one, decays. These results show the potential of QONTRAST. Let us assess the viability of QONTRAST on hardware with a collection of experiments from the M3 documentation next.

B. M3: Sampling with Bernstein-Vazirani (BV)

Our next experiments feature a noise mitigation test by sampling BV circuits of multiple qubit lengths as described in [16]. The test generates BV circuits for all-ones bit-strings of various lengths, and transpiles them against the 27 qubit noisy simulator "FakeKolkata". This is a stress test for mitigation as the noise accrues when the number of CNOT (Control-X) gates increases. We modified the test to run on hardware, only altering the code to insert the QONTRAST filter mitigation logic with intensity ranges of 3, 4, 5, and 10%. The success probability is displayed for the raw (unmitigated), M3 mitigated, and QONTRAST with two intensity levels in Figure 5. The experiment from the M3 documentation was conducted for 8192 shots transpiled at level 3.

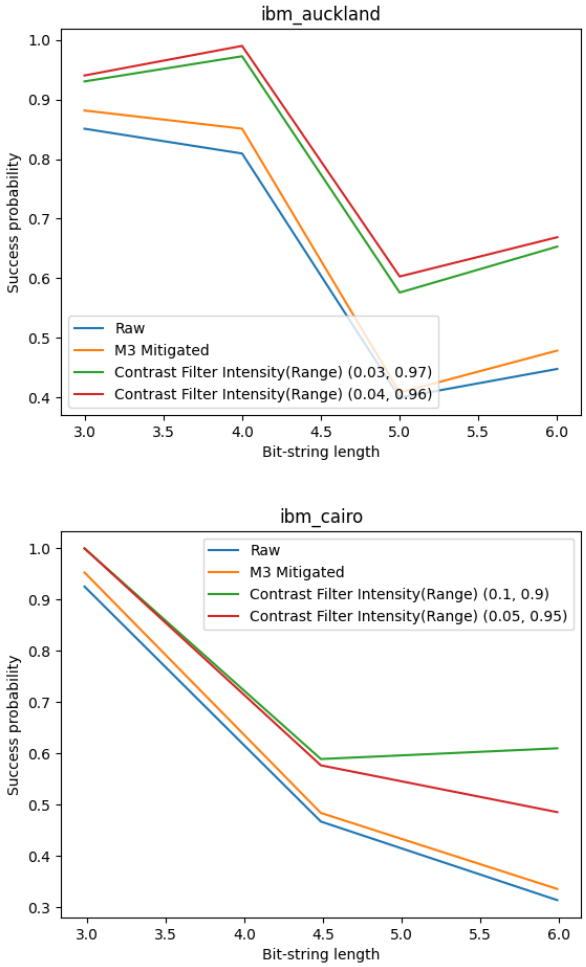

Fig. 5. Noise mitigation for BV of various lengths using M3 on 27-qubit devices (Auckland, Cairo): Success probability for all-ones (y-axis) over circuit qubit strength (x-axis).

The original experiment under noisy simulation correctly identifies strings with up to 12 qubits. Yet, on hardware an exception is raised by Qiskit runtime for qubit lengths above 6. This is due to the fact that noise accrues at such a high level for 7 or more qubits that the probability of the trivial (all-ones) outcome is overshadowed by an invalid outcome.

An explanation for this failure is also given in the M3 documentation describing the experiment [16]: In the setup stage, the code creates a (not-rigorous) function that finds the most likely bit string from a distribution (most_likely_bitstring). Next, the unmitigated (raw) data for the circuits is collected as a basis to perform mitigation using M3. However, the post-process logic validates that the returned bit string is what we expect before getting the success probability. This works as designed in noisy simulation, however in hardware, the noise accrues at such a high level that the most likely outcome does not match the expected outcome (i.e., the validation fails) throwing an exception and aborting the experiment. This can be seen in cells 7 and 12 from [16], where an exception is intentionally thrown if results are incorrect.

Observation 1: QONTRAST outperforms M3 in results by up to 20% for filter intensity ranges between 3-10%. Benefits of QONTRAST increase with the number of qubits.

This experiment used the calibration data of Table III.

TABLE III
HARDWARE SPECS FOR BV EXPERIMENT.

Processor	**Qubits**	**T1, T2 (us)**	**Readout, CNOT Error**	**Queue wait, Exec time**
ibm_auckland	27	128.2, 123.7	$8.9e^{-3}$, $4.8e^{-3}$	8h, 25.4s
ibm_cairo	27	105.1, 110.4	$1.25e^{-2}$, $9.3e^{-3}$	6h, 12.3s

C. M3: Basic Variational Quantum EigenSolver

This 2-qubit VQE test originates from [13]. It takes the sample Hamiltonian $H = 0.3979YZ - 0.3979ZI - 0.01128ZZ + 0.1809XX$ and calculates the expected energy value using the following VQE state preparation:

- Ansatz: Two-local from Qiskit library with full entanglement (each qubit is entangled with all others).
- Measurement circuits: Created by moving the observables in the Hamiltonian (YZ, ZI, ZZ, XX) into the computational basis by appending a Hadamard Gate (H) to each Pauli X gate, plus S-dagger and H gates for each Y gate in the observable while Z and I gates remaining unchanged.
- Initial parameter angles: [1.22253725, 0.39053752, 0.21462153, 5.48308027, 2.06984514, 3.65227416, 4.01911194, 0.35749589].

As the results in Table IV show, we again observe that QONTRAST at 1% input range (0.01, 0.99) yields a more accurate estimate of the final energy of the Hamiltonian than M3. Specifically, result shows an improvement of around 2%. The optimal solution is obtained from [13]. The final angles for QONTRAST and M3 where [1.25560585 0.38654154 0.20846952 5.48743517 2.03181833 3.68814594 3.91540956 0.36953303] and [1.23757107 0.4211179 0.08658293 5.49275181 2.05102127 3.67389758 4.03722484 0.3746737], respectively.

TABLE IV
BASIC VQE RESULTS FOR THE 27 QUBIT CAIRO PROCESSOR FOR T1=101.1μsec (RELAXATION) AND T2=98.9μsec (DEPHASING)

Method	Energy
M3	-0.4248273342124237
QONTRAST	-0.43324214733405414
optimal	-0.44841884382998787

D. Qiskit Runtime: Quantum Resilience on Energy Landscapes of Variational Circuits

This experiment showcases the error suppression and noise mitigation at the quantum level built into Qiskit Runtime [14]. These experiments feature expectation values as results so that we can now compare to a variety of other methods, namely:

- A two-qubit variational circuit from Mitiq [32]: An energy landscape using entangled rotations over the X-axis of the block sphere depending on a single parameter Theta: $< H > (\theta) =< Z \otimes Z > (\theta)$;
- IBM's noisy simulators in Qiskit Runtime [30] with variational phases over theta $[0, 2\pi]$ $f(x) = 2sin(\theta/2)^2 - 1$;
- IBM's error suppression using Trotterization circuits [14]. Trotterization is a technique to segment an exponentiated matrix into a series of smaller exponentiated matrices with minimal error. This method reduces the complexity by providing an estimate of the solution to a time-dependent Hamiltonian. For a detailed description of this technique see [9].

IBM's Qiskit runtime implements three levels of Quantum resilience, which were introduced in Section II:

1) Twirled readout error extinction (T-Rex) reduces noise by attaching extra measurement and calibration circuits for the estimation of error-mitigated averages;
2) Zero Noise Extrapolation (ZNE) generates as a set of circuits increasing amplified noise whose measurements are backwards projected onto the zero-noise plain;
3) Probabilistic error cancellation (PEC) samples a collection of circuits that mimic a noise inverting channel to cancel noise in the desired computation similar to the way noise-canceling headphones work.

We altered the experiment to include QONTRAST and obtained the results shown in Figure 6. The top row of the figure depicts the three circuits that generate the landscape. The bottom row shows graphs of the corresponding results for the Noisy (*unmitigated* raw device), Mitigated via ZNE and our QONTRAST filter at a 2.5% input range [0.025, 0.975] plotting expectation values (y-axis) over different phases (x-axis) for the first two circuits or number of trotterization steps (x-axis) for the third circuit. The experiment was run on devices Lagos, Auckland, and Hanoi for the respective plots. Runtime specs are shown in Table V. The plots indicate that QONTRAST outperforms T-Rex and ZNE, in particular where spikes occurs, by over 10% in the best case, while never performing worse than the other techniques.

We do not include results for PEC. Even though we experimented with PEC, its sampling overhead is exponential to the number of gates. All of our PEC experiments with complex circuits above 10 qubits crashed after multiple hours of run time, which was acknowledged by IBM as a bug.

Observation 2: Our results indicate that QONTRAST outperforms ZNE and has significantly lower complexity than any other scheme, particularly than PEC, which fails to provide results when scaling up the number of qubits.

E. Qiskit Runtime: State Preparation using VQE

For the next experiment, we chose the state preparation for the Kagome lattice using the VQE algorithm. This experiment was part of IBM's Open Science Prize 2022. The challenge amounts to preparing the frustrated ground state of a Heisenberg spin-1/2 model on a Kagome lattice using the VQE algorithm. A basic notebook was provided by IBM for participants to explore the circuit design space. We use this algorithm to assess the behavior of our method for more complex circuits, i.e., wider (more qubits) and deeper (more gates) ones. The state chosen for this experiment consists of:

TABLE V
HARDWARE SPECS AND EXECUTION METRICS FOR QUANTUM RESILIENCE ON ENERGY LANDSCAPES OF VARIATIONAL CIRCUITS.

Processor	**Qubits**	**T1,T2(us)**	**Readout, CNOT Error**	**Queue wait, Exec time**
ibm_lagos	7	97.9, 71.3	$1.4e^{-2}$, $2.9e^{-3}$	5m, 35.1s
ibm_auckland	27	132.0, 108.6	$8.6e^{-3}$, $1.9e^{-3}$	10m, 32.1s
ibm_hanoi	27	138.8, 107.9	$1.1e^{-2}$, $7.9e^{-3}$	1h, 5.1s

- Hamiltonian: The antiferromagnetic Heisenberg model arranged on a Kagome lattice $H = \sum_{i,j} X_iX_j + Y_iY_j + Z_iZ_j$.
- Ansatz: EfficientSU2, a circuit made of layers of single-qubit operations spanned by Unitary-Set-Dimension2 SU(2) and Controlled-X entanglements from the Qiskit library. This is a heuristic pattern that can be used to prepare trial wave functions for quantum algorithms or classification circuits for machine learning.
- Backend: The 16-qubit ibmq_guadalupe processor.
- Optimizer: The Nakanishi-Fujii-Todo algorithm [17] from the Qiskit library. This is a technique that divides the parameterized quantum circuits into sub-problems using only a subset of the parameters; this transforms the cost function into a sine curve with period 2π. By repeating this procedure, the parameterized quantum circuits are optimized and the cost function is minimized. Here, the maximum number of iterations to perform defaults to 100.
- Shots or number of measurements in experiment: 2048.
- The expected ground state: -18 calculated via a classical EigenSolver.

The results are shown in Figure 7 with the runtime metrics depicted in Table VI. The metric of performance is given by the relative difference between the expected ground state (-18.0) and the VQE result. The top plot depicts the unmitigated ground state with a relative error of 37.6%. The bottom plot shows QONTRAST mitigated results at a filter range of 0.2% (0.002-0.998) showing an improvement with a relative error of 15.2%, i.e., a noise reduction of 22 percentage points.

TABLE VI
CALIBRATION AND EXECUTION METRICS FOR THE GROUND STATE OF THE KAGOME LATTICE EXPERIMENT.

Processor	**Qubits**	**T1, T2 (us)**	**Readout, CNOT Error**	**Queue wait, Exec time**
ibm_guadalupe	16	95.1, 110.3	$1.7e^{-2}$, $8.7e^{-3}$	4d, 3.5h

The image stretching effect of QONTRAST assumes that the

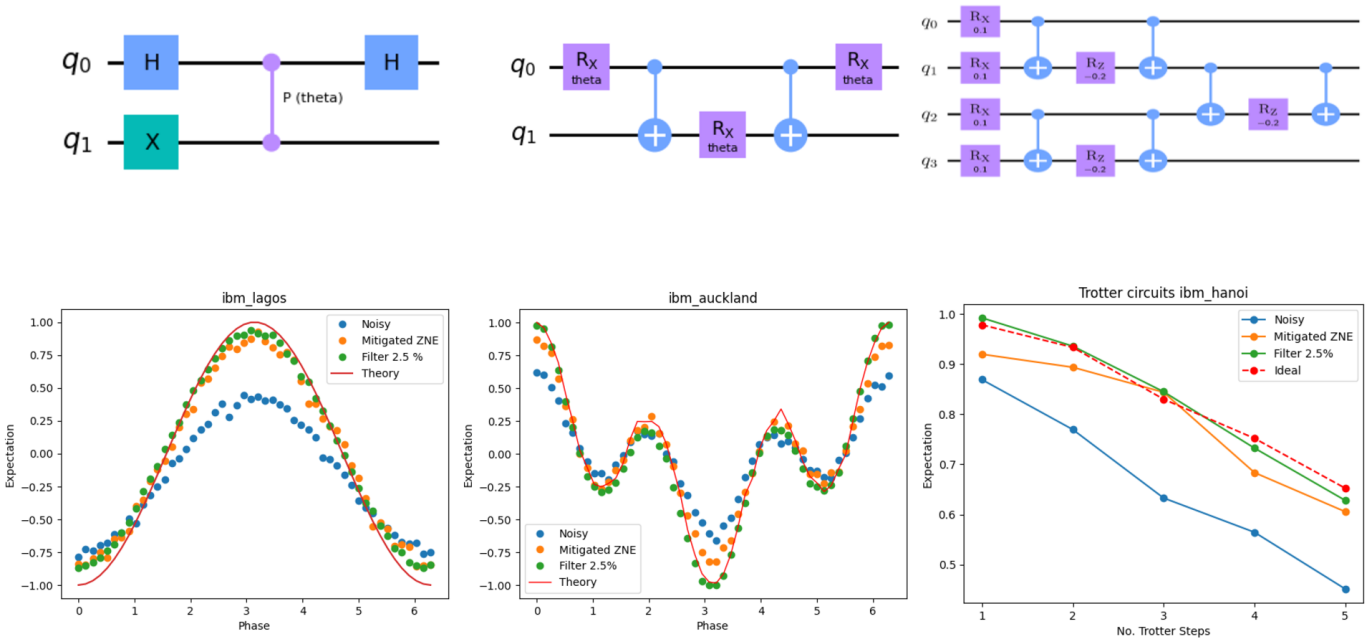

Fig. 6. Quantum resilience on 3 energy landscapes in Qiskit Runtime. Top row: circuits per landscape. Bottom row: Corresponding results for Noisy, ZNE Mitigated, QONTRAST filter and theory.

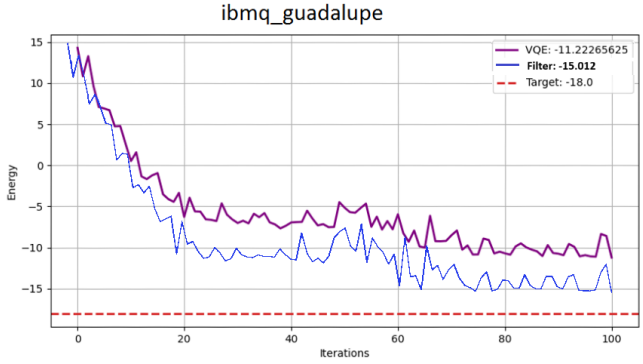

Fig. 7. Experimental results for the ground state preparation of the Kagome lattice.

low probability outcomes are made purely of noise, which is the case for non-variational circuits (Bell States, GHZ, BV, etc). However, this may not be the case for variational circuits (VQE, QAOA, etc), where low probability outcomes contain both information and noise. This is reflected in our results here, where average input ranges for VQE are very low (0.2%) compared to non-variational ranges (3-10)%.

Observation 3: Relatively high levels of contrast (above 2%) result in loss of information for variational circuits when employing QONTRAST, yet other techniques perform no better (and generally worse).

F. SupermarQ Benchmarks

We complement our results with SupermarQ, a scalable, hardware-agnostic quantum benchmark suite using application-level metrics to measure performance [34]. SupermarQ systematically applies techniques from classical benchmarking methodology to the quantum domain. We compare QONTRAST against M3 for the complete set of SupermarQ benchmarks.

The full set of benchmarks is shown in Figure 8. All measurement values are normalized based on the ideal/noiseless y-axis values.

For GHZ (top left two graphs), we observe that QONTRAST with a 5% filter range outperforms M3 in terms of Hellinger fidelity (y-axis), with a trending increase in benefit as the number of qubits is increased (x-axis). Differences between quantum devices indicate better results for Hanoi than Cairo irrespective of the error mitigation scheme, with a drop in Hellinger fidelity at 9 qubits.

Mermin-Bell (top right two graphs) show clear increases in the Mermin operator score (y-axis) for QONTRAST at 5% over M3, which again strengthens with qubit count and superior fidelity of Hanoi over Cairo.

Bit-Code and Phase-Code equally benefit from higher Hellinger fidelity for QONTRAST at 5% over M3 across qubit rounds (x-axis). Here, Hanoi provides inferior results to Cairo for M3. In contrast, QONTRAST provides consistently stronger and more stable results.

VQE shows small improvements in approximating ground energy (y-axis) of QONTRAST at 0.2% over M3, where Cairo outperforms Hanoi for larger qubit layers (x-axis). One exception is the 7-1 qubit layer on Cairo, where M3 outperforms QONTRAST, yet QONTRAST still outperforms M3 on Hanoi for the same 7-1 layer. As variational algorithms tolerate noise in quantum devices, results are close between QONTRAST, M3 and even noisy (unmitigated) experiments.

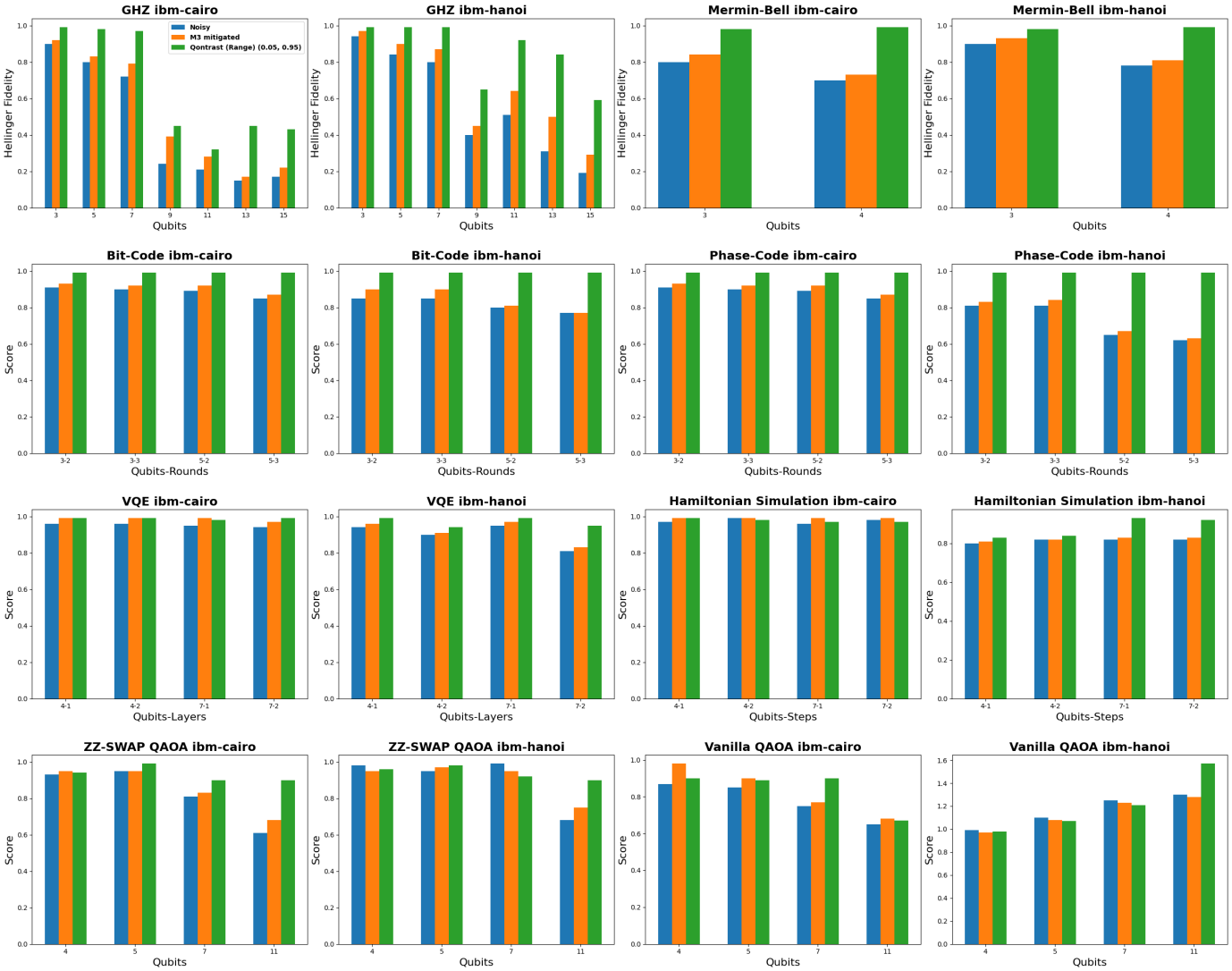

Fig. 8. SupermarQ benchmarks on 27-qubit Cairo+Hanoi for raw device (Noisy), M3 mitigated and QONTRAST's filter over benchmark setting (x-axis) and respective metrics (y-axis).

Hamiltonian Simulation for TFIM magnetization (y-axis) resembles VQE closely with gains for QONTRAST at 0.2% over M3 except for Cairo with qubit steps (x-axis) 4-2 and 7-1 due to noise tolerance. Benefits for QONTRAST on Hanoi are more pronounced for 7-1 and 7-2.

QAOA with ZZ-Swap shows QONTRAST benefits at 0.2% over M3 in Hellinger fidelity (y-axis), which is more pronounced for larger number of qubits (x-axis). Exceptions are 4 and 7 qubits on Hanoi, where noisy (unmitigated) results are best, due to occasional algorithmic noise toleration in variational algorithms. Vanilla QAOA has even more mixed results, where all three schemes perform about the same, except that M3 is best for 4-qubit Cairo and QONTRAST at 0.2% is best at 7-bit Cairo and 11-qubit Hanoi.

Observation 4: For SupermarQ, QONTRAST is consistently outperforming M3 noise mitigation over all non-variational benchmarks and tested quantum devices at a 5% filter range. QONTRAST often performs at par or outperforms other techniques for variational algorithms as well at a 0.2% filter range.

Discussion: Having demonstrated the benefits of QONTRAST for BV, VQE and a variety of SupermarQ benchmarks, our method has shown to be quite robust and superior to M3 in performance and at least as good (often even better) in fidelity. Even for more complex distributions with multiple results and in the presence of various noise sources on actual devices (e.g., crosstalk and other correlated noise), we empirically observe that QONTRAST performs better than prior techniques in practice.

VI. SENSITIVITY STUDIES AND DISCUSSION

Table VII shows that QONTRAST provides 1-18% higher fidelity results than M3 across all benchmarks for the best filter range when averaged overall all results (any number of qubits). To gain a deeper understanding of QONTRAST, we

TABLE VII
AVG. PERFORMANCE FOR QONTRAST VS. M3 ON CAIRO (TYPE, QUBIT RANGE, M3 FIDELITY OR SCORE, FILTER FIDELITY OR SCORE (WITH % CHANGE), M3 CALIBRATION TIME, M3 CORRECTION TIME, AND FILTER CORRECTION TIME WITH % CHANGE OVER M3, TIMES IN MSECS).

Experiment	**Qubit range**	**M3 Fidelity**	**Filter Fidelity**	**M3 Calibration time**	**M3 Correction time**	**Filter Correction time**
M3 - Bernstein-Vazirani	5-10	0.62	0.78 (16% ↑)	450	9	0.19 (97% ↓)
Runtime -Trotterization	4-6	0.80	0.84 (4% ↑)	102	64	0.9 (98% ↓)
Runtime - Ising Models	12	0.63	0.75 (12% ↑)	1009	123	5 (95% ↓)
SupermarQ GHZ	3-15	0.54	0.69 (15% ↑)	55	10	0.08 (99% ↓)
SupermarQ Mermim-Bell	3,4	0.77	0.95 (18% ↑)	42	5	0.09 (98% ↓)
SupermarQ Bit-Code	3,5	0.86	0.97 (16% ↑)	29	5	0.98 (80% ↓)
SupermarQ Phase-Code	3,5	0.90	0.98 (8% ↑)	45	5	0.12 (97% ↓)
SupermarQ Hamiltonian Simulation	4,7	0.98	0.98 (0%)	150	25	1.05 (96% ↓)
SupermarQ VQE	4,7	0.985	0.985 (0%)	140	26	2.1 (92% ↓)
SupermarQ QAOA Fermionic Swap	4-11	0.87	0.92 (5% ↑)	500	96	0.5 (99% ↓)
SupermarQ QAOA Vanilla	4-11	0.85	0.86 (1% ↑)	450	45	0.71 (98% ↓)

perform additional sensitivity experiments in three parts: The optimal input range values control the power of QONTRAST and are critical to the performance of the filter. Execution metrics compare the run times of both methods, and platform independence gauges how flexible this method can be.

A. Automatic Input Range Selection and Artificial Bias in VQA Training

The stretching effect of QONTRAST penalizes output strings with low probability and amplifies output strings with high probability. This works well for circuit types like GHZ, Bernstein-Vazirani. However, for VQA this artificial bias can hurt training and cause convergence problems. This is because in VQE and QAOA, the output distributions are spread out and have higher entropy (variance). Nevertheless, we counter this bias by adjusting the input range values of the filter to control the loss of information. In fact, this input range controls the amount of contrast (or bias) that is applied. The result can be seen in Figure 7, the state preparation of the Kagome lattice using VQE (Sect. V-E), where a low input range (contrast level) of about (0.002-0.998) or 0.2% yields a 22 percentage point reduction on error levels. For GHZ, a non-VQA algorithm, a larger range of (0.05-0.95) or 5% is more beneficial resulting in increase in fidelity by about 15% (see Table VII).

We compiled information about estimated optimal values for the input ranges of the QONTRAST filter for multiple circuit types, qubit strengths and circuit depths in experiments presented so far. The results show that, as with any conventional contrast filter, too little contrast will have no effect on the final mitigated result. On the other hand, too much contrast may cause loss of information. Hence, this range must be carefully calibrated depending on the circuit type. Furthermore, we tested adaptive techniques for auto-input range selection in a variant called QONTRAST-Adaptive. This variant automatically adjusts the output range based on previous outcomes for an arbitrary number of cycles. Our tests show a fidelity boost up to 6% for most experiments with a small performance penalty. The results are presented in Table VIII detail contrast ranges that we tried for various benchmarks. They reinforce our claim that 5% is a good choice for noise-intolerant algorithms while 0.2% works best for variational ones. For the former, 5% is a good starting point to experiment with, which can be improved on if expected results are know. In the absence of known expectations, 5% might be the best choice, or one could experiment within the 1%-10% range if results can be classically checked at low cost.

TABLE VIII
ESTIMATED FIDELITY FOR AUTO-RANGE SELECTION IN QONTRAST-ADAPTIVE FOR 5 CYCLES.

Experiment	**Qubits (Depth)**	**Optimal input ranges (%)**	**Fidelity QONTRAST-Adaptive**
M3 - BV	5-10 (8-13)	5-10	0.81 (5% ↑)
Runtime - Trotter	4-6 (7-42)	2.5	0.87 (4% ↑)
Runtime - Ising	12 (6-24)	1-3	0.77 (3% ↑)
GHZ	3-15 (4-16)	5-10	0.73 (6% ↑)
Mermim-Bell	3-4 (5-6)	5-10	0.81 (5% ↑)
Bit-Code	3,5 (5,7)	5-10	0.88 (2% ↑)
Phase-Code	3,5 (5,7)	5-10	0.92 (3% ↑)
VQE	4,7 (6,7)	0.2	0.98 0%
Hamiltonian	4,7 (7-49)	0.2	0.99 (1% ↑)
QAOA ZZ-Swap	4-11 (7-30)	0.2	0.97 0%
QAOA Vanilla	4-11 (7-30)	0.2	0.87 (2% ↑)

B. Classical Analysis Cost

We gathered the execution times of both the QONTRAST and M3 methods to gain an insight into their runtime cost. The

tests were run on an Intel Core i5 CPU @ 2.60GHz with 8GB of memory and a 64-bit Windows OS. The results are shown in Table IX[1] complemented by Table VII across all benchmarks for the best results. For the latter table, notice that M3 incurs calibration overhead before any circuit is run *and* correction overhead afterwards, whereas QONTRAST only incurs correction cost, yet the % savings of QONTRAST correction is only comparing to M3 correction (excluding calibration), i.e., total overhead savings are even more significant when considering calibration.

Observation 5: The classical analysis cost of QONTRAST is at least 1 and occasionally as much as 4 orders of magnitude smaller than that of M3.

TABLE IX
EXECUTION TIMES [MS] (% CHANGE) FOR M3 VS. QONTRAST, SELECTED EXPERIMENTS FROM SECTION V.

Experiment	M3 Calibration	M3 Correction	Filter Correction	Qubit range
GHZ	452	10	0.92 (99% ↓)	5
BV Sample	670	0.9	0.1 (99% ↓)	3-7
Dyn. BV	77174	94	10 (99% ↓)	2-15
Basic VQE	190	12572	12177 (3% ↓)	2

C. Platform Independence

Platform independence is an important aspect of any effective error mitigation system because it enhances portability and flexibility. To the best of our knowledge, the M3 mitigator implementation targets a specific IBM backend by first computing the calibration matrices for the given qubits and given number of shots. It then applies mitigation to a given dictionary of raw counts over the specified qubits. Our QONTRAST, on the other hand, works by treating the measured probabilities as "image pixels" and applying an intensity contrast filter to reduce the noise. Therefore, our system should work on any quantum platform. To illustrate this, we ran some of the experiments from Section V on the IonQ cloud simulator with the results shown in Figure 9.[2] The results in the top bar chart show that QONTRAST improves output fidelity for a 5-qubit entanglement (see Section V-A) over raw measurements for the two expected results to where they add up to 100% coverage, whereas the noisy method provides incorrect results at low rates. The line graph below for BV (see Section V-B) also illustrates the benefits of QONTRAST at 0.2% (top green line, perfect results) and 0.1% (middle orange line) over noisy (raw) results (bottom blue line).

Observation 6: QONTRAST generalizes in terms of providing noise mitigation agnostic of quantum device technology and generations, as shown by experiments over a variety of superconducting devices and ion trap technology.

[1]Note that experiments in Section V-C, V-D and V-E using Qiskit runtime require server-side changes and cannot be easily integrated.

[2]Notice that we also submitted jobs for IonQ on Amazon Braket, however due to unavailability of IonQ devices in practice over many weeks, these jobs were never executed.

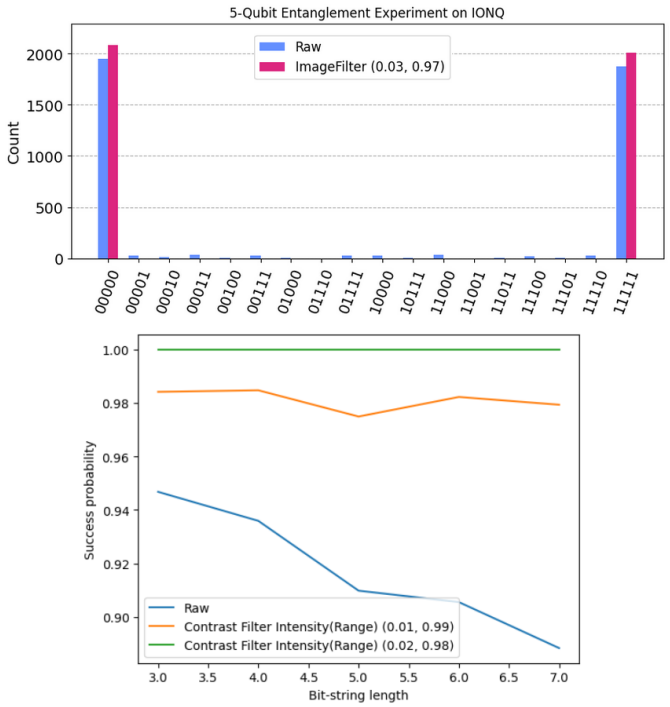

Fig. 9. IonQ cloud simulation under noise model IonQ Aria-1 for experiments from Sec. V-A (top) and V-B (bottom).

D. Performance on Large Number of Qubits

Tables X and XI show simulation results on large qubits consistent with our lower qubit benchmarks. The fidelity of our method outperforms M3 and it is competitive with ZNE with significant savings on execution time. Note that the statevector simulator has a hard limit of 30 qubits on our single CPU machine.

TABLE X
FIDELITY UP TO 30 QUBITS ON STATEVECTOR SIMULATOR WITH NOISE MODEL: 10% BIT-FLIP, 10% PHASE-FLIP AND 10% READOUT ERROR.

Experiment	M3	ZNE	QONTRAST
GHZ-20	0.75	0.81	0.82
GHZ-25	0.63	0.71	0.70
GHZ-30	0.51	0.51	0.54
BV-20	0.75	0.80	0.82
BV-25	0.60	0.62	0.64
BV-30	0.43	0.48	0.48
VQE-20	0.8	0.85	0.86
VQE-25	0.71	0.76	0.78
VQE-30	0.66	0.70	0.71

VII. OTHER RELATED WORK

JigSaw [4] is a method that mitigates the impact of measurement errors executing programs in two modes: (1) A global measurement of all qubits to generate a global Probability Mass Function (PMF) with low fidelity but high correlation; and (2) a subset-mode with Partial Measurements (CPM) to produce local-PMFs with higher fidelity but poor correlation. A Bayesian update is finally applied to the local-PMFs to

TABLE XI
EXECUTION TIMES (M=MINUTES, H=HOUR) FOR TABLE X. HARDWARE: LINUX X86/64 (4 VCPUS, 32 GB RAM) GPU:NVIDIA L4-24GB.

Experiment	M3	ZNE	QONTRAST
GHZ-20	9.6m	13.5m	8.2m
GHZ-25	17.3m	21.4m	12.4m
GHZ-30	26.3m	32.1m	19.1m
BV-20	8.1m	15.3m	6.1m
BV-25	13.3m	25.2m	9.3m
BV-30	22.0m	29.1m	18.0m
VQE-20	36.2m	39.3m	34.0m
VQE-25	62.4m	70.2m	58.1m
VQE-30	1.8h	2.1h	1.1h

enhance the global-PMF. Table XII shows averages of Probability of Successful Trial (PST) from the authors for the GHZ, Bernstein-Vazirani and QAOA benchmarks vs. QONTRAST.

TABLE XII
PST AVERAGES FOR JIGSAW, AND JIGSAW-M RUN IN IBMQ_TORONTO. QONTRAST RUN IN IBM_AUCKLAND, AND IBM_CAIRO. THE NUMBER OF QUBITS IS SHOWN IN PARENTHESIS.

Benchmark	Jigsaw	Jigsaw-M	QONTRAST
GHZ	2.0 (14)	2.2 (14)	2.1 (13)
BV	6.5 (6)	7.0 (6)	15 (6)
QAOA	2.3 (8)	2.4 (8)	2.2 (11)

Furthermore, the authors estimate JigSaw's time complexity scales linearly with the number of qubits and trials. Our method, does not require extra executions, which inflate runtimes in variational applications like VQE. For example, the state preparation of the Kagome lattice using VQE (Sect. V-E) requires 150 quantum-optimizer cycles imposing a considerable overhead on final results, even under linear scaling as the linear factor is high.

HAMMER [28] is a post processing technique to boost the fidelity by estimating the likelihood of each outcome based on a neighborhood of answers within a small Hamming distance. The authors claim that erroneous outcomes are not arbitrary but exhibit a well-defined structure when represented in the Hamming space. Their results show that HAMMER improves quality by 1.37x on average in 500 quantum circuits on IBM and Google datasets. The authors also claim it scales to thousands of qubits. Just like HAMMER, QONTRAST is a post processing method, yet we do not need to consider Hamming distances but treat outcome probabilities as gray scale pixels, followed by a filtering technique to stretch the good outcomes and shrink the bad ones improving overall fidelity. HAMMER uses a solution quality metric based on cost ratios, probability of successful trial, and magnitude of correct vs. incorrect answers. On the other hand, our QONTRAST focuses on standard fidelities, which makes a direct comparison difficult. All in all, both methods scale to large number of qubits and are platform independent.

Q-Beep [25] builds upon HAMMER's idea about error structure in the Hamming spectrum. While HAMMER assumes errors occur in local clusters, Q-Beep shows that they also occur non-local clusters characterized through a Poisson distribution. Q-Beep uses an iterative algorithm with a Bayesian network state-graph for error mitigation taking into account input circuit, calibration statistics, and qubit topology. Just like HAMMER and Q-Beep, QONTRAST is a light-weight post-processing technique with competitive fidelity improvements (up to 18% in SupermarQ for QONTRAST, and up to 17.8% in QASMBench for Q-Beep); Nevertheless, QONTRAST does not use an iterative algorithm, thus it is faster and consumes less resources. All in all, these three techniques are useful tools delivering state-of-the-art readout noise mitigation in different ways.

VIII. CONCLUSION

During the current NISQ stage, error mitigation has become a prime area of research. It is critical to develop an efficient and fast method to eliminate noise from measurement outcomes. At the quantum level, there are many resilience techniques, error correction codes, and other methods. However, these approaches require an excessive number of ancillary qubits and furthermore contribute to the overall noise themselves. We developed QONTRAST, a method to mitigate measurement errors in the distribution counts of a Quantum computer using image contrast filters at various input ranges. In comparison to IBM's M3 mitigation library, QONTRAST outperformed M3 across all benchmarks at most settings. We also compared to IBM's quantum resilience (T-Rex, ZNE, and PEC) with equally positive results for QONTRAST. We believe, at this stage, that noise can be managed at the classical level with our fast and efficient QONTRAST technique more adequately than any other mitigation technique, to the best of our knowledge. As such, QONTRAST may also lower the feasibility barrier to quantum error correction in the future. The source code is available for download [1].

ACKNOWLEDGMENT

This work was funded in part by NSF grants OSI-2410675, OSI-2531350, PHY-2325080, CISE-2316201, and OMA-2120757.

REFERENCES

[1] anonymous. Qontract software artifact, 2024. https://omitted...
[2] Piotr Czarnik, Andrew Arrasmith, Patrick J. Coles, and Lukasz Cincio. Error mitigation with Clifford quantum-circuit data. *Quantum*, 5:592, November 2021. https://doi.org/10.22331/q-2021-11-26-592.
[3] Aniket S. Dalvi, Filip Mazurek, Leon Riesebos, Jacob Whitlow, Swarnadeep Majumder, and Kenneth R. Brown. Modular architecture for classical simulation of quantum circuits. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 810–812, 2022.
[4] Poulami Das, Swamit Tannu, and Moinuddin Qureshi. Jigsaw: Boosting fidelity of nisq programs via measurement subsetting. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21. ACM, October 2021.
[5] Bo Fang, M. Yusuf Ozkaya, Ang Li, Umit V. Catalyurek, and Sriram Krishnamoorthy. Efficient Hierarchical State Vector Simulation of Quantum Circuits via Acyclic Graph Partitioning . In *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 289–300, Los Alamitos, CA, USA, September 2022. IEEE Computer Society.

[6] Tudor Giurgica-Tiron, 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)*. IEEE, oct 2020. https://doi.org/10.1109%2Fqce49297.2020.00045.

[7] Thomas Häner, Damian S. Steiger, Torsten Hoefler, and Matthias Troyer. Distributed quantum computing with qmpi. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA, 2021. Association for Computing Machinery.

[8] Tsung-Wei Huang. qTask: Task-parallel Quantum Circuit Simulation with Incrementality . In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 746–756, Los Alamitos, CA, USA, May 2023. IEEE Computer Society.

[9] Grant Kluber. Trotterization in quantum theory, 2023.

[10] Gushu Li, Yufei Ding, and Yuan Xie. Eliminating redundant computation in noisy quantum computing simulation. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2020.

[11] Ying Li and Simon C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2), jun 2017. https://doi.org/10.1103%2Fphysrevx.7.021050.

[12] Chao-Yang Lu, Daniel E. Browne, Tao Yang, and Jian-Wei Pan. Demonstration of a compiled version of shor's quantum factoring algorithm using photonic qubits. *Phys. Rev. Lett.*, 99:250504, Dec 2007. https://link.aps.org/doi/10.1103/PhysRevLett.99.250504.

[13] mthree (2.5.1) Team. Basic mitigated vqe, 2021. https://qiskit.org/ecosystem/mthree/tutorials/10_basic_vqe.html.

[14] mthree (2.5.1) Team. Error suppression and error mitigation with qiskit runtime, 2021. https://qiskit.org/ecosystem/ibm-runtime/tutorials/Error-Suppression-and-Error-Mitigation.html.

[15] mthree (2.5.1) Team. Scalable mitigation library for quantum computers, 2021. https://qiskit.org/ecosystem/mthree/index.html.

[16] mthree (2.5.1) Team. Scalable mitigation library for quantum computers, 2021. https://qiskit.org/ecosystem/mthree/sampling.html.

[17] Ken M. Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4), oct 2020. https://doi.org/10.1103%2Fphysrevresearch.2.043158.

[18] Paul D. Nation, Hwajung Kang, Neereja Sundaresan, and Jay M. Gambetta. Scalable mitigation of measurement errors on quantum computers. *PRX Quantum*, 2(4), nov 2021. https://doi.org/10.1103%2Fprxquantum.2.040326.

[19] Tirthak Patel, Daniel Silver, and Devesh Tiwari. Geyser: a compilation framework for quantum computing with neutral atoms. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, page 383–395, New York, NY, USA, 2022. Association for Computing Machinery.

[20] Tirthak Patel and Devesh Tiwari. Qraft: Reverse your quantum circuit and know the correct program output. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '21, page 443–455, New York, NY, USA, 2021. Association for Computing Machinery. https://doi.org/10.1145/3445814.3446743.

[21] Anurudh Peduri, Siddharth Bhat, and Tobias Grosser. Qssa: an ssa-based ir for quantum computing. In *Proceedings of the 31st ACM SIGPLAN International Conference on Compiler Construction*, CC 2022, page 2–14, New York, NY, USA, 2022. Association for Computing Machinery.

[22] XinYu Piao, JooYong Shim, Joongheon Kim, and Jong-Kook Kim. Aqua: Hardware-agnostic qubit allocation for quantum multi-programming. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 154–161, 2025.

[23] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. https://doi.org/10.1137/S0036144598347011.

[24] Kaitlin N. Smith, Michael A. Perlin, Pranav Gokhale, Paige Frederick, David Owusu-Antwi, Richard Rines, Victory Omole, and Frederic Chong. Clifford-based circuit cutting for quantum simulation. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA '23, New York, NY, USA, 2023. Association for Computing Machinery.

[25] Samuel Stein, Nathan Wiebe, Yufei Ding, James Ang, and Ang Li. Quantum bayesian error mitigation employing poisson modelling over the hamming spectrum for quantum error mitigation, 2023.

[26] Samuel Stein, Nathan Wiebe, Yufei Ding, Peng Bo, Karol Kowalski, Nathan Baker, James Ang, and Ang Li. Eqc: ensembled quantum computing for variational quantum algorithms. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, page 59–71, New York, NY, USA, 2022. Association for Computing Machinery.

[27] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. Cutqc: using small quantum computers for large quantum circuit evaluations. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '21, page 473–486, New York, NY, USA, 2021. Association for Computing Machinery.

[28] Swamit Tannu, Poulami Das, Ramin Ayanzadeh, and Moinuddin Qureshi. Hammer: boosting fidelity of noisy quantum circuits by exploiting hamming behavior of erroneous outcomes. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22. ACM, February 2022.

[29] IBM Team. Correcting probabilities, 2021. https://qiskit.org/ecosystem/mthree/sampling.html.

[30] IBM Team. Noisy simulators in qiskit runtime, 2021. https://qiskit.org/ecosystem/ibm-runtime/how_to/noisy_simulators.html.

[31] IBM Team. Qiskit runtime ibm client error mitigation., 2021. https://qiskit.org/ecosystem/ibm-runtime/how_to/error-mitigation.html.

[32] Mitiq Team. Mitigating the energy landscape of a variational circuit with mitiq., 2021. https://mitiq.readthedocs.io/en/stable/examples/simple-landscape-cirq.html.

[33] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18), nov 2017. https://doi.org/10.1103%2Fphysrevlett.119.180509.

[34] Teague Tomesh, Pranav Gokhale, Victory Omole, Gokul Subramanian Ravi, Kaitlin N. Smith, Joshua Viszlai, Xin-Chuan Wu, Nikos Hardavellas, Margaret R. Martonosi, and Frederic T. Chong. Supermarq: A scalable quantum benchmark suite, 2022.

[35] Ewout van den Berg, Zlatko K. Minev, and Kristan Temme. Model-free readout-error mitigation for quantum expectation values. *Physical Review A*, 105(3), mar 2022. https://doi.org/10.1103%2Fphysreva.105.032620.

[36] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883 EP –, Dec 2001. http://dx.doi.org/10.1038/414883a.

[37] Zhiqian Xu, Honghui Shang, Yi Fan, Xiongzhi Zeng, Yunquan Zhang, and Chu Guo. Scalable and differentiable simulator for quantum computational chemistry. In *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 230–240, 2024.

Appendix: Artifact Description/Artifact Evaluation

Artifact Description (AD)

IX. Overview of Contributions and Artifacts

A. Paper's Main Contributions

This work introduces contrast filters as another method to reduce noise in quantum computing. An implementation of the technique, QONTRAST, is compared to the state-of-the techniques in both fidelity improvements and overhead for a variety of benchmarks.

- C_1 Experiment 1: Probability Mitigation.
- C_2 Experiment 2: Sampling with Bernstein-Vazirani (BV).
- C_3 Experiment 3: Basic Variational Quantum Eigen-Solver.
- C_4 Experiment 4: Quantum Resilience on Energy Landscapes of Variational Circuits.
- C_5 Experiment 5: State Preparation using VQE.
- C_6 Experiment 6: SupermarQ Benchmarks.

B. Computational Artifacts

- A_1 https://github.com/Shark-y/quantum_mitigation/blob/main/00_m3_ghz.ipynb
- A_2 https://github.com/Shark-y/quantum_mitigation/blob/main/02_m3_bv.ipynb
- A_3 https://github.com/Shark-y/quantum_mitigation/blob/main/05_m3_vqe.ipynb
- A_4 https://github.com/Shark-y/quantum_mitigation/blob/main/04_m3_trotter.ipynb

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_1	Figure 2,3,4
A_2	C_2	Tables 3 Figures 5
A_3	C_3	Tables 4 Figures 7
A_4	C_4	Tables 5 Figures 6

X. Artifact Identification

A. Computational Artifact A_1

A method to mitigate noise in quantum computing with low classical overhead compared to other QNM methods.

Relation To Contributions

A generalized description of C_1.

Expected Results

A reduction in the noise distribution of the probability outcomes.

Expected Reproduction Time (in Minutes)

See Table III.

Artifact Setup (incl. Inputs)

Hardware: Access to IBM's quantum computers via the cloud-based ecosystem.

Software: Qiskit open-source Software Development Kit (SDK) for building, running, and optimizing quantum programs.

Datasets / Inputs: Quantum Circuit for a Bell pair.

Installation and Deployment:

a) Environment Setup:

1) Use Miniconda or similar to create isolated Python environments to avoid conflicts.
2) Activate Environment: Activate your new environment
3) Install Qiskit: Use pip install qiskit.
4) Add Extras: Install visualization (qiskit-visualization) or other specialized libraries (qiskit-algorithms, qiskit-machine-learning) with pip as needed.
5) Verification: Run import qiskit in a Jupyter notebook or Python script; no errors means success.

b) Deployment (Running and Simulation):

1) Authentication: Get an API key from the IBM Quantum Platform and load it into your environment (via file or environment variable).
2) Execution:
   - Local: Use Qiskit Aer simulators for quick testing.
   - Cloud: Submit circuits to actual quantum hardware or advanced simulators via Qiskit Runtime.

Artifact Execution

Follow the environment Setup and Deployment instructions, run artifact A_1 lab.

Artifact Analysis (incl. Outputs)

Insights and Analysis are described in section III.

B. Computational Artifact A_2

A noise mitigation experiment by sampling Bernstein-Vazirani circuits of multiple qubit lengths.

Relation To Contributions

An experiment for contribution C_2.

Expected Results

QONTRAST outperforms M3 in results by up to 20% for filter intensity ranges between 3-10%. Benefits of QONTRAST increase with the number of qubits.

Expected Reproduction Time (in Minutes)

See Table III.

Artifact Setup (incl. Inputs)

Hardware: Same as C_1.

Software: Same as C_1.

Datasets / Inputs: Bernstein-Vazirani circuits for all ones bitstrings of various lengths.

Installation and Deployment: Same as C_1.

Artifact Execution

Run the experiment lab for A_2, C_2.

Artifact Analysis (incl. Outputs)

Insights and Analysis are described in section V subsection B.

C. Computational Artifact A_3

A noistate preparation for the Kagome lattice using the VQE algorithm

Relation To Contributions

An experiment for contribution C_3.

Expected Results

QONTRAST shows an improvement with a relative error of 15.2%, i.e., a noise reduction of 22 percentage points.

Expected Reproduction Time (in Minutes)

See Table VI, Figure 7.

Artifact Setup (incl. Inputs)

Hardware: Same as C_1.

Software: Same as C_1.

Datasets / Inputs: See section V.E also Table VI, Figure 7.

Installation and Deployment: Same as C_1.

Artifact Execution

Run the experiment lab for A_3, C_3.

Artifact Analysis (incl. Outputs)

Insights and Analysis are described in section V.E.

D. Computational Artifact A_4

Quantum Resilience experiment on Energy Landscapes of Variational Circuits.

Relation To Contributions

An experiment for contribution C_4.

Expected Results

QONTRAST outperforms T-Rex and ZNE, in particular where spikes occurs, by over 10performing worse than the other techniques.

Expected Reproduction Time (in Minutes)

See Table V.

Artifact Setup (incl. Inputs)

Hardware: Same as C_1.

Software: Same as C_1.

Datasets / Inputs: See section V.D also Table V, VII, Figure 6.

Installation and Deployment: Same as C_1.

Artifact Execution

Run the experiment lab for A_4, C_4.

Artifact Analysis (incl. Outputs)

Insights and Analysis are described in section V.D, Table VII.

Artifact Evaluation (AE)

a) Environment Setup:

1) Use Miniconda or similar to create isolated Python environments to avoid conflicts.
2) Activate Environment: Activate your new environment
3) Install Qiskit: Use pip install qiskit.
4) Add Extras: Install visualization (qiskit-visualization) or other specialized libraries (qiskit-algorithms, qiskit-machine-learning) with pip as needed.
5) Verification: Run import qiskit in a Jupyter notebook or Python script; no errors means success.

b) Deployment:

1) Authentication: Get an API key from the IBM Quantum Platform and load it into your environment (via file or environment variable).
2) Execution:
   - Local: Use Qiskit Aer simulators for quick testing.
   - Cloud: Submit circuits to actual quantum hardware or advanced simulators via Qiskit Runtime.

A. Computational Artifact A_1

This experiment is designed to correct readout errors and transform the outcomes into a true probability distribution. The experiment creates a circuit with 5 entangled qubits (Figure 4), performs measurements on all of them, and runs on the 14-qubit noisy simulator.

Artifact Setup (incl. Inputs)

Follow environment/deployment instructions.

Artifact Execution

Execute the experiment for A_1, C_1.

Artifact Analysis (incl. Outputs)

QONTRAST outperforms M3 by reducing the noisy outcomes as shown in the probability histogram.