



UREQA: Leveraging Operation-Aware Error Rates for Effective Quantum Circuit Mapping on NISQ-Era Quantum Computers

Tirthak Patel, Baolin Li, Rohan Basu Roy, and Devesh Tiwari,
Northeastern University

<https://www.usenix.org/conference/atc20/presentation/patel>

This paper is included in the Proceedings of the
2020 USENIX Annual Technical Conference.

July 15–17, 2020

978-1-939133-14-4

Open access to the Proceedings of the
2020 USENIX Annual Technical Conference
is sponsored by USENIX.

UREQA: Leveraging Operation-Aware Error Rates for Effective Quantum Circuit Mapping on NISQ-Era Quantum Computers

Tirthak Patel, Baolin Li, Rohan Basu Roy, and Devesh Tiwari
Northeastern University

Abstract

Noisy Intermediate-Scale Quantum (NISQ) computers are enabling development and evaluation of real quantum algorithms, but due to their highly erroneous nature, careful selection of qubits to map the algorithm on to real hardware is required to minimize the error rate of the algorithm output. In this paper, we propose UREQA, a new approach that introduces quantum-operation-aware error rate prediction to minimize of output errors of quantum algorithms running on NISQ devices.

1 Introduction

Noisy Intermediate-Scale Quantum (NISQ) computers are enabling development and evaluation of quantum algorithms in various domains including chemistry and physics simulations, combinatorial and black-box optimization, and quantum cryptography [16, 20]. A major challenge toward increasing the practicality and wide-adoption of quantum computing is the high error rate observed on current NISQ devices, often orders of magnitude higher than the classical computing systems [16,20]. This challenge is likely to remain prevalent in the near-term, and requires innovative techniques to mitigate the side-effects of quantum errors [6, 12, 16, 20]. Next, we provide a brief background of quantum computers, errors, and algorithm execution.

Background and Problem Statement. In quantum computers, a *qubit* is the fundamental unit, analogous to a classical bit. A qubit state ($|\Psi\rangle$) can be expressed as a superposition of the two basis states: $|0\rangle$ and $|1\rangle$. More formally, $|\Psi\rangle = a|0\rangle + b|1\rangle$, where a and b are complex numbers such that $\|a\|^2 + \|b\|^2 = 1$. Due to the quantum physical behavior of qubits, upon measurement, this superposition collapses, and the qubit is found either in state $|0\rangle$ (with probability $\|a\|^2$) or in state $|1\rangle$ (with probability $\|b\|^2$). Multiple operations can be sequentially performed on a set of qubits on a computer to run a quantum algorithm. Upon the completion of the execution of a quantum algorithm, the qubit state is measured for all qubits and the output is analyzed.

The qubit operations, referred to as *quantum operations*, are of three types: 1-qubit gates, 2-qubit gates and readout. 1-qubit gates operates on a single qubit and change the superposition state of the qubit. 2-qubit gates

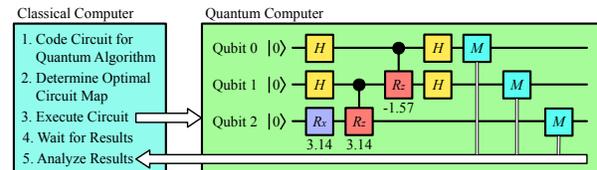


Figure 1: Execution flow of Quantum Phase Estimation (QPE) algorithm mapped to a quantum circuit. Each horizontal line represents a qubit, and each box represents a quantum operation. The time order of operations flows from left to right.

entangle two qubits and can change the state of the “target” qubit depending on the state of the “control” qubit (Sec. 2 provides more details on these operations). The readout operation simply refers to measuring a qubit’s state. The readout operation is applied only at the end of the execution as it destroys the qubit’s state. A *quantum algorithm* is expressed as a sequence of gate operations operating on multiple qubits. The mapping of a logical quantum algorithm on the physical qubits of a quantum computer is referred to as a *quantum circuit* — a set of quantum operations and the corresponding qubits on which the operation is being performed. An example of the execution of a quantum circuit is shown in Fig. 1. This example circuit has three qubits all initialized to the ground state $|0\rangle$. A few quantum gates are applied to them (details provided in Sec. 2) and their states are measured at the end and transferred over to a classical computer. It is important to note that a *single* quantum algorithm can map to *multiple* circuits in different ways on the same computer, much like how a classical algorithm can be executed on any permutation of transistors.

Unfortunately, qubits and operations on NISQ machines are highly error-prone and hence, quantum circuits have erroneous output. State-of-the-art approaches carefully choose the qubits and operations with the lowest overall error rate for determining the best circuit map [3, 15, 18, 25–28, 31, 33]. Error rate of each operation is estimated based on the historical information (e.g., IBM calibrates and publishes error rate for all qubits of a machine on a twice-a-day basis). Based on these error rates, a “good” circuit map is chosen to obtain an outcome that has a high probability of being close to the correct output. Estimating error rate correctly is the key to choosing the optimal circuit map for a given

quantum algorithm. For example, if the estimated error rate of each qubit is significantly different than the actual error rate when the circuit map is executed, then these differences add up over the circuit map execution and result in a significantly inaccurate outcome. *Therefore, previous works have focused on estimating the error rates accurately and using that to find the optimal circuit map [3, 15, 18, 25, 26, 28].*

What is Missing from Existing Solutions? Current approaches use a single number to characterize the error rate of a given qubit irrespective of the different quantum operations being performed on the qubit [25, 26, 28, 31]. For the first time, we show that error rate is not only qubit-specific, but also operation-specific (as explained in Sec. 2, a 1-qubit gate can perform different types of quantum operations on a single qubit). We show that quantum error rates can vary significantly depending on the specific quantum operation that is being performed, even if other conditions are kept constant (i.e., the physical qubit and the machine). Some qubits with low aggregate average error rate might experience high error rate for specific quantum operations. Hence, these qubits should be avoided for a circuit-map selection if a particular circuit consists of many such specific quantum operations. The reason for this phenomena is the unstable nature of current NISQ technology where qubits are erroneous and do not have consistent properties as different qubits interact differently with external control and the environmental features. *This is the first work to discover and leverage the above insight to choose better circuit maps that lower the impact of quantum errors, and push the state-of-the-art in improving the efficiency of quantum algorithm execution on NISQ computers.*

UREQA Solution. UREQA¹ builds a data driven model for correctly estimating the error rates of operations on different qubits of a quantum computer, and then, leverages this information to find the most optimized circuit for a quantum algorithm. UREQA builds its error rate prediction model by performing a large number of experiments on real IBM NISQ computers. Our evaluation shows that these error rate prediction methods are more accurate than current state-of-the-art approaches of simply using a general error rate number periodically published by the quantum computing platform provider.

To demonstrate UREQA’s effectiveness, we evaluate UREQA for a diverse set of quantum benchmarks, conducting experiments over more than 50 days on four different quantum computers in the IBM QX cloud. Our results show that our operation-aware solution achieves a small median prediction error rate of 1%. Using these operation error-rate prediction models, UREQA’s opti-

¹UREQA (Eureka) stands for utilizing operation-aware error rate predictions (for better circuit mapping) on quantum computers.

Table 1: IBM QX quantum computers.

Online Date	Computers (Num. Qubits)
Nov 06, 2018	Melbourne (14), Yorktown (5)
Jul 03, 2019	Ourense (5), Vigo (5)

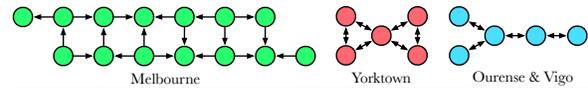


Figure 2: Layout of IBM quantum computers. The circles represent qubits. The arrows show possible 2-qubit gates: the direction points from control to target qubit.

mized circuit map selection achieves up to 15% reduction in error rate for a quantum algorithm, compared to the current approaches which rely on a single aggregated number for error rate estimation based on historical data.

UREQA’s quantum error prediction model and circuit mapping framework is open-sourced at <https://github.com/GoodwillComputingLab/UREQA>.

2 UREQA: The Solution

Background. This study is performed on the IBM Quantum Experience (QX) - a public cloud service. We use the IBM QX machines listed in Table 1. They cover a diverse range of quantum architectures in terms of error-rates, topology, and time of introduction (Fig. 2).

Quantum operations on these computers include both the gate and readout operations. Primary 1-qubit gates include the Hadamard (H) gate which puts the two basis states into equal superposition and the x -, y -, and z -rotation gates (R_x , R_y , and R_z , respectively) which rotate the qubit about the x -axis, y -axis and z -axis on the Bloch Sphere, respectively. The Bloch Sphere is a unit sphere with the $|0\rangle$ state represented as a vector pointing toward the positive z -axis and the $|1\rangle$ state is represented on the negative z -axis. The other two axis represent the qubit phase. The qubit state vector can point anywhere on the Bloch Sphere, but upon readout, it collapses to the positive ($|0\rangle$) or negative ($|1\rangle$) z -axis. As an example, Fig. 3 uses the Bloch Sphere to show the state changes after applying a H gate followed by a R_z gate with π rotation to a single qubit. When the H gate is applied, the qubit state vector points toward the positive x -axis and the qubit is equally probable to be measured as $|0\rangle$ or $|1\rangle$. This probability of measurement remains the same even after a R_z rotation is applied, except the qubit has a negative phase.

All 1-qubit gates have 2-qubit variants (CH , CR_x , CR_y and CR_z) where one qubit is the control and the other is the target. The respective 1-qubit gate is applied to the target qubit depending on the superposition of the control qubit. In Fig. 1, the connection between qubit 0 and R_z gate of qubit 1 means it is a CR_z gate with qubit 0 as control and qubit 1 as target.

These qubit operations can be erroneous. IBM’s qubits are fixed-frequency superconducting Transmon qubits

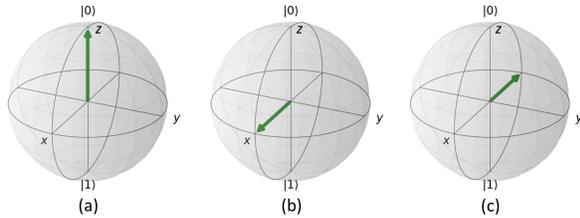


Figure 3: A qubit (green arrow tip) on a Bloch sphere. The qubit in (a) first gets manipulated by an H gate to state in (b), then by a R_z gate to state in (c).

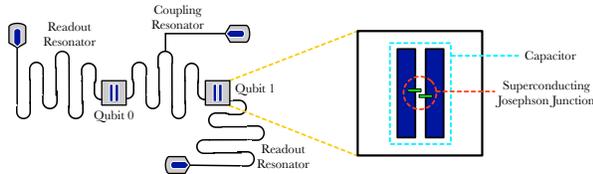


Figure 4: Design of IBM's superconducting qubits technology.

based on Josephson Junctions, and the Transmon frequency is referred to as the **qubit frequency**. On IBM's quantum computers, the qubits are implemented using Josephson Junctions created by separated superconducting electrodes and capacitors as shown in Fig. 4. 1-qubit gates are performed by applying external controls in the form of microwave pulses. Errors in applying these pulses cause **1-qubit gate errors**. Entanglement between two qubits is performed using coupling resonators. These coupling resonators can be highly erroneous causing **2-qubit gate errors**. Lastly, readout operation (or qubit state measurement) is performed using readout resonators as shown in Fig. 4. The readout resonators are also highly error-prone and cause **readout errors** when qubit states are measured. In fact, other factors can also affect error rates. Once initialized, a qubit can only retain its state for a limited time (coherence time). There are two types of coherence times: (1) The **T1 coherence time** is associated with amplitude damping due to the qubit's natural energy decay to the ground state. (2) The **T2 coherence time** is associated with phase damping due to environmental factors.

IBM's computers are calibrated twice a day, and the qubit coherence times change after each calibration. We note that the error rates are determined when calibration tasks are performed for all the operations of a quantum computer. Calibration is the task of determining qubit frequency and accordingly, setting the properties of the microwave tone which changes the state of a qubit. During calibration, operation characteristics such as the frequency of a qubit and the optimal microwave tone amplitude are determined based on new properties of the qubit. These characteristics are then used to perform all the operations. These new characteristics determine the error rate of the operation. The effect of environmental factors (such as the electromagnetic interference, fluctu-

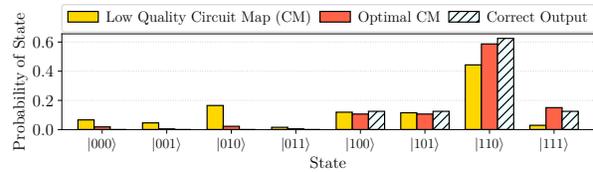


Figure 5: Choice of circuit map can greatly impact overall output error: different circuit maps for the QPE algorithm.

ating temperature, or mechanical vibrations) is already captured in the operation error rates. Coherence times are also measured immediately after calibration is performed. Note that regular circuits (jobs) cannot run on machines when calibration is being performed. Thus, it is impractical to constantly keep calibrating the machines, and hence, this practical constraint forces the calibration to be performed typically twice daily.

Current Efforts in Circuit Mapping. IBM posts a single error number for all 1-qubit and 2-qubit gates for each qubit twice a day. One solution to the aforementioned circuit mapping problem can be to map quantum operations on qubits which have the minimum operation error rates according to these posted numbers [10, 25, 26, 28]. The idea is to maximize the Estimated Success Probability (*ESP*) of a quantum circuit [25]. The ESP is calculated as $\prod_{i=1}^{N_{gates}} g_i * \prod_{j=1}^{N_{readout}} m_j$, where g is the success rate of gates and m is the success rate of readout (success rate = 1 - error rate). The circuit map with highest ESP is the optimal circuit map. Fig. 5 shows the impact of choosing a low quality circuit map vs. an optimal circuit map for executing the quantum phase estimation (QPE) algorithm. The correct output of QPE has states $|100\rangle$, $|101\rangle$, and $|111\rangle$ with probability 0.125, and state $|110\rangle$ with probability 0.625. On real-systems, executing a circuit map results in state probabilities that are different than the correct probabilities. Using the correct probabilities as reference, the optimal circuit map has an overall error of 6% (sum of errors of all states divided by 2), while the low quality circuit has an overall error of 28%. Thus, estimating the ESP of a circuit map accurately (and hence, in turn estimating the error rate of quantum operations) is critical for mitigating the side-effects of erroneous quantum operations. However, current approach of using the published numbers to estimate the error rates implicitly assumes that all 1-qubit operations have uniform errors and that all 2-qubit operations also have uniform errors. This is far from the actual behavior of errors as we show next.

Different quantum operations exhibit significant variation in observed error rates. Fig.6 shows that quantum errors are correlated with the specific type of operation being performed (on the same qubit; results are averaged over all available qubits and platforms for simplicity). For example, on IBM computers R_z is imple-

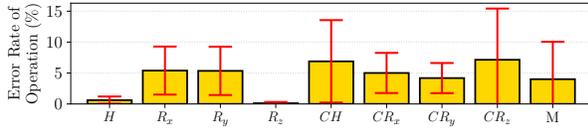


Figure 6: Different quantum operations can have significant different error rates with a high degree of variance.

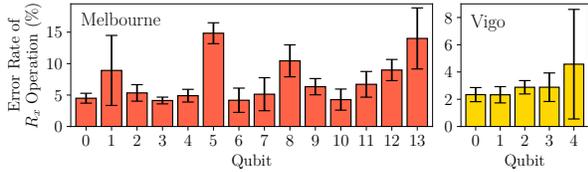


Figure 7: Error rate of a quantum operation varies across machines and among qubits within the same machine.

mented as a simple frame change with no physical computation; hence, it has close to 0 error rate. The H gate also has a low error rate; however, the error rates of the other two 1-qubit gates R_x and R_y are much higher. 2-qubit gates like CH and CR_z also have high error rates. It is a conventional belief that 2-qubit gates have a higher error rate than 1-qubit gates [10, 23, 27]. However, our analysis reveals that while on average 2-qubit gates have higher error rates than 1-qubit gates, certain types of 1-qubit gates such as R_x and R_y have error rates comparable to 2-qubit error rates and readout (M) error rate.

A potential reason for varying error rates among different quantum operations can be the difference in microwave tones that are applied to implement the operation on a gate. For example, an R_x gate with a π rotation (on a single qubit) is applied using a Gaussian microwave pulse of a certain calibrated amplitude A . On the other hand, the H gate (also, on a single qubit) is implemented using a Gaussian microwave pulse of half the amplitude ($A/2$) and pre- and post-pulse frame changes. It can lead to a lower error rate because its pulse has half the amplitude of the R_x gate and the frame changes have zero error. Overall, this finding of operation-specific error rates motivates the need for accounting for the operation type when estimating the error rates.

Errors rates also vary temporally and spatially across computers and qubits. In Fig. 6, the error bars show the high standard deviation (variation) of operation error rates across time. This situation is further exacerbated by the fact that error rates for the same operation also vary across different computers and qubits, as shown in Fig. 7 for the R_x gate. The variation in the error rates for R_x across different qubits is considerable and has not decreased even in the newest IBM quantum computer, Vigo. The error rates vary across qubits and across operations performed on the same qubit.

UREQA Overview. While using published errors does offer simplicity, accurate estimation of ESP needs bet-

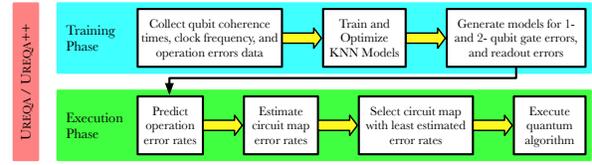


Figure 8: Key steps in UREQA workflow.

Table 2: Predictive features of different operations.

Operation	Predictive Features
1-Qubit Gate	Computer ID, Qubit T1 Coherence Time, Qubit T2 Coherence Time, Qubit Frequency, Gate Type (H , R_x , R_y , R_z)
2-Qubit Gate	Computer ID, Control Qubit T1 Coherence Time, Control Qubit T2 Coherence Time, Control Qubit Frequency, Target Qubit T1 Coherence Time, Target Qubit T2 Coherence Time, Target Qubit Frequency, Gate Type (CH , CR_x , CR_y , CR_z)
Measurement (Readout)	Computer ID, Qubit T1 Coherence Time, Qubit T2 Coherence Time, Qubit Frequency

ter prediction of operation error rates. The high degree of instability and uncertainty makes it difficult to model the behavior of these errors using analytical or rule-based models. Therefore, UREQA takes a data-driven machine-learning-based approach, as shown in Fig. 8. We develop a data-driven model which helps perform accurate predictions of error rates of individual operations. Then, when executing quantum algorithms, these pre-trained models can be used to estimate circuit map error rates and the best circuit map can thus be selected to execute the algorithm with minimal errors.

UREQA Model Development. To collect the error rate data for different types of operations, we developed micro benchmarks that perform a specific operation on every qubit of all available quantum computers. For example, to get the readout error, every qubit was measured in its initialization state of $|0\rangle$ without running any gate operation. To get the error of H , R_x , R_y , R_z , CH , CR_x , CR_y , and CR_z , the corresponding gate-operation was performed, and result was measured and compared against the ground truth. Our automated workflow collected over 20,000 samples. Each run consisted of 1024 trials — multiple trials need to be conducted because the output of a quantum circuit is probabilistic. Other data such as coherence times and frequencies of individual qubits were obtained from IBM’s daily calibration results.

The goal of model development is to predict the error rate of a given operation given a set of predictive features. A complete list of these features is provided in Table 2 for the three types of quantum operations. For example, given qubit 0 on Melbourne computer’s T1 coherence time, T2 coherence time, and frequency after today’s calibration, predict its readout error rate. The predictive features are chosen based on their availability on the IBM QX machines and based on their relevance to the qubit operation (e.g., T1 coherence time of qubit 1 is relevant to the 1-qubit error rate of the same qubit but not to other qubits on the machine). Principle Component Analysis (PCA) was performed to determine the features

Table 3: Optimal parameters tuned for KNN learners.

Operation	Number of Neighbors	Distance Metric	Distance Weight
1-Qubit Gate	13	Euclidean	Squared Inverse
2-Qubit Gate	31	Correlation	Inverse
Readout	68	Jaccard	Inverse

which contribute the most to the variance of the dataset and the features shown in Table 2 were found to account for more than 95% of all variance in the dataset.

After thorough experimentation and hyper-parameter tuning, we assessed that k Nearest Neighbors (k-NN) classification learner [1] was the best learning model for error rate prediction – it has the lowest mis-classification error. The output of a k-NN learner is a membership to a class (class here refers to a particular error rate). A sample is classified by a plurality vote of its neighbors given a set of predictors. The sample gets assigned the class most common among its k nearest neighbors.

The hyper-parameter optimization of the k-NN learners was performed using Bayesian Optimization [21] which builds a stochastic model of the parameter space by progressively sampling parameters which have the highest expected improvement based on the constructed model. The parameters optimized include number of nearest neighbors involved in the voting process, distance metric (e.g., Euclidean, Manhattan, etc.), and distance weight (e.g., Equal, Inverse, etc.). Table 3 shows the optimal parameters obtained after performing Bayesian-Optimization-based hyper-parameter tuning for the three error rate learners. 90% of the dataset was used for training and 10% was used for testing. Data was randomized so as to ensure that all quantum computers, qubits, and operation characteristics were included in the training dataset. The training was performed using 5-fold cross validation [32] to avoid machine-learning methodology pitfalls such as over-fitting the model.

Finally, we note that UREQA is practically feasible. The model building and training can be completed within a few days. This process needs to be invoked or refreshed when significant architectural/operational changes are introduced to a particular machine. UREQA’s result quality is not highly sensitive to the number of samples (e.g., an incremental improvement in result quality over 10,000 samples is limited). Note that the investment of one-time 20,000 sample based training can potentially be amortized over multiple months in choosing better circuit mappings which reduce the error rate.

3 UREQA: Evaluation and Analysis

Quantum Algorithms and Circuits. Real quantum benchmark algorithms were programmed using *Qiskit*, IBM’s python-based language for quantum circuits [2], and executed on different IBM computers (benchmarks are listed in Table 4). The results from these circuits are

Table 4: Quantum benchmarks used for evaluation.

Benchmark ID	Benchmark description
$BV2$	2-Qubit Bernstein-Vazirani [5]
$BV3$	3-Qubit Bernstein-Vazirani [5]
QPE	Quantum Phase Estimation [8]
SIA	Simon’s algorithm [13]
HR_x^8H	Circuit to stress X gate errors (expected output $ 0\rangle$)
$R_xHR_x^8H$	Circuit to stress X gate errors (expected output $ 1\rangle$)

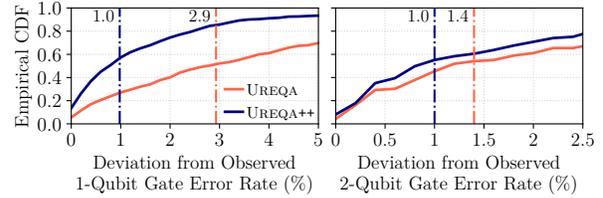


Figure 9: The prediction quality is much better when operation-aware predictor is used in UREQA++.

used to assess the improvement in circuit mapping due to improved prediction power. *Note that the training data of individual operation execution and the assessment data of execution of real quantum circuits were generated at disjoint sets of time periods to avoid biasing the results.*

The benchmarks are chosen to cover a diverse set of quantum algorithm characteristics. For example, the Bernstein-Vazirani (BV) benchmarks heavily use H and R_z gates and the QPE benchmark has high number of 2-qubit gates such as the CR_z gates. Finally, two home-grown benchmarks HR_x^8H (H gate followed by 8 R_x gates followed by H gate) and $R_xHR_x^8H$ are used. These benchmarks are designed to generate and test large 1-qubit gate errors with a long sequence of R_x gates.

Evaluation Metrics. The prediction quality of a model is assessed using *deviation of the predicted value from the observed value of the operation error rate*. The effectiveness of a method’s prediction on real quantum algorithms is assessed using the *overall output error rate when a circuit map is selected for an algorithm using the prediction provided by a model/method*.

Evaluated Techniques. (1) The base method: Best circuit map is selected by maximizing the ESP using the operation errors posted by IBM as used in current approaches [10, 25, 26, 28]. (2) UREQA: Best circuit map is selected by predicting operation errors using k-NN models trained without using operation-specific information. (3) UREQA++: Best circuit map is selected by predicting operation errors using k-NN models trained with all features including operation-specific information.

Including operation-specific information in the prediction improves the prediction quality significantly. Fig. 9 shows that when operation-specific information (type of 1-qubit or 2-qubit gate) is included as the predictor, the prediction error is much lower on average. For 1-qubit gates, the median deviation from observed error rate is only 1% with UREQA++ compared to 2.9% with

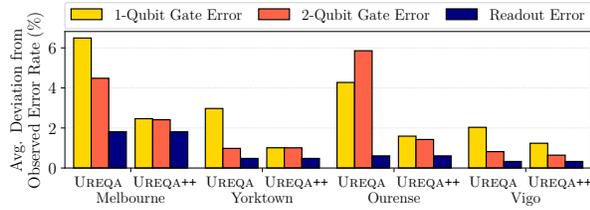


Figure 10: The prediction quality is good across different computers with UREQA++.

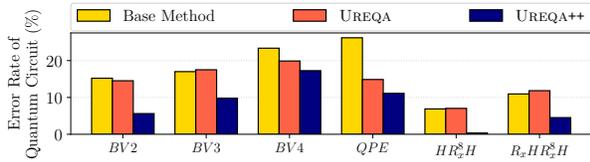


Figure 11: Circuit maps selected using UREQA++ perform much better as compared to other methods.

UREQA. We note that the CDF of UREQA++ has a much steeper rise, indicating that the prediction error is smaller for a large majority of test samples. Similarly, for 2-qubit gates, the median deviation from observed error rate is only 1% with UREQA++ and it is 1.4% with UREQA. As our results show later, even this seemingly small difference has a compounded impact when these predictions are used to estimate the error rate of an entire circuit because multiple operations are used in real quantum algorithms. Note that both UREQA and UREQA++ are hyperparameter optimized using the Bayesian-Optimization-based procedure described earlier. Thus, both models are optimal for their given set of predictor features. Yet, UREQA++ performs better as it is trained in an operation-aware manner.

The deviation from observed value is small across different quantum computers when using UREQA and UREQA++. Fig. 10 shows that UREQA and UREQA++ have another desirable result: the deviation from the observed value is low across the four computers for all the three types of operation errors. Melbourne has slightly higher deviation from the observed error rates than other computers because it has older and more unstable technology which makes error rates vary considerably. This makes prediction difficult. For the other three computers, the average deviation is less than 2% with UREQA++.

Quantum circuit error rate drops significantly when the circuit map is chosen using predictions provided by UREQA++. Fig. 11 shows error rates when the best circuit maps to execute a quantum algorithm are selected using different methods. UREQA achieves similar or significantly better results in some cases (e.g., more than 10% in QPE) compared to the base method. The similar results are mostly due to the fact that both methods do not consider operation-specific information. On the other hand, UREQA++ performs much better across dif-

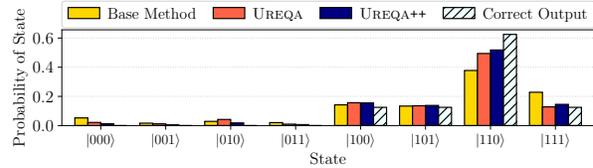


Figure 12: Overall output is much less erroneous with UREQA++ and UREQA than with the base method.

ferent quantum algorithms, generally resulting in over 5-15% improvement in the error rate compared to the base method (e.g., *BV2*, *QPE*, *HR_x⁸H*). UREQA++ is able to achieve this low circuit error by producing output state probabilities close to the correct output. Fig. 12 provides an example of this for QPE, where evidently, the circuits produced by the base method and UREQA have more error from the correct output (Table 5) than UREQA++. This demonstrates that the improved prediction quality with UREQA++ when using operation-specific information results in the selection of better circuit maps, which ultimately reduces the side-effect of erroneous quantum operations on current real quantum hardware.

Table 5: State error results for the QPE output in Fig. 12.

Method	Error of Each State from the Correct Output (%)								Overall Error (%)
	000>	001>	010>	011>	100>	101>	110>	111>	
Base	5.3	1.6	2.9	2.0	1.7	0.9	24.7	10.3	$\Sigma/2=24.7$
UREQA	2.1	1.2	4.2	1.0	3.2	1.1	13.1	0.3	$\Sigma/2=13.1$
UREQA++	1.3	0.5	1.8	0.7	3.0	1.3	10.7	2.0	$\Sigma/2=10.7$

4 Related Work and Conclusion

Quantum Error Correction. Previous works have proposed algorithms for error correction in qubits which rely on heavy computational requirements which are unavailable in the current NISQ computers [4, 7, 11, 14, 24, 29, 30]. Thus, these methods are not applicable to NISQ technology, while UREQA++ helps reduce the error rate of quantum algorithms executed on NISQ hardware.

Minimization of Error Rates. Many recent works have employed approaches to mitigate the effects of error rates in quantum circuits via both online and offline methods [3, 9, 10, 12, 15, 17–19, 22, 23, 23, 26, 27, 31, 33, 34]. These include optimizing circuit maps to minimize error rates [23], migrating circuits to less-erroneous qubits [34], and minimizing error-prone operations [31]. UREQA’s operation-aware circuit mapping technique achieves up to 15% reduction in error rate for quantum programs, compared to the current approaches. UREQA’s open-source contribution pushes the state-of-the-art in quantum error rate prediction to minimize erroneous output, which can be leveraged by the quantum computing systems community.

Acknowledgment We are thankful to anonymous reviewers and our shepherd, Srinivasan Narayanamurthy, for improving the quality of this paper.

References

- [1] AHA, D. W., KIBLER, D., AND ALBERT, M. K. Instance-based Learning Algorithms. *Machine learning* 6, 1 (1991), 37–66.
- [2] ALEKSANDROWICZ, ET AL. Qiskit: An Open-source Framework for Quantum Computing.(2019).
- [3] ASH-SAKI, ET AL. QURE: Qubit Re-allocation in Noisy Intermediate-Scale Quantum Computers. DAC.
- [4] BENNETT, ET AL. Mixed-State Entanglement and Quantum Error Correction. *Physical Review A* 54, 5 (1996).
- [5] BERNSTEIN, E., AND VAZIRANI, U. Quantum Complexity Theory. *SIAM Journal on computing* 26, 5 (1997).
- [6] BRAVYI, ET AL. Trading Classical and Quantum Computational Resources. *Physical Review X* 6, 2 (2016).
- [7] BURNETT, ET AL. Decoherence Benchmarking of Superconducting Qubits. *npj Quantum Information* 5, 1 (2019).
- [8] CLEVE, R., EKERT, A., MACCHIAVELLO, C., AND MOSCA, M. Quantum Algorithms Revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454, 1969 (Jan 1998).
- [9] DAS, ET AL. A Case for Multi-Prog. Quantum Comps. MICRO.
- [10] GOKHALE, ET AL. Partial Compilation of Variational Algorithms for Noisy Intermediate-Scale Quantum Machines. MICRO.
- [11] HUANG, ET AL. Performance of Quantum Error Correction with Coherent Errors. *Physical Review A* 99, 2 (2019).
- [12] HUANG, ET AL. Statistical Assertions for Validating Patterns and Finding Bugs in Quantum Programs. ISCA.
- [13] KOIRAN, P., NESME, V., AND PORTIER, N. A Quantum Lower Bound for the Query Complexity of Simon’s Problem. In *International Colloquium on Automata, Languages, and Programming* (2005), Springer.
- [14] LAYDEN, ET AL. Ancilla-Free Quantum Error Correction Codes for Quantum Metrology. *Physical review letters* 122, 4 (2019).
- [15] LI, G., DING, Y., AND XIE, Y. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. ASPLOS.
- [16] MARTONOSI, ET AL. Next Steps in Quantum Computing: Computer Science’s Role. *arXiv preprint arXiv:1903.10541* (2019).
- [17] MAVADIA, ET AL. Prediction and Real-Time Compensation of Qubit Decoherence via Machine Learning. *Nature communications* 8 (2017).
- [18] MURALI, ET AL. Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers. ASPLOS.
- [19] MURPHY, ET AL. Controlling Error Orientation to Improve Quantum Algorithm Success Rates. *Physical Review A* 99, 3 (2019), 032318.
- [20] PRESKILL, J. Quantum Computing in the NISQ Era and Beyond. *Quantum* 2 (2018).
- [21] SHAHRIARI, B., SWERSKY, K., WANG, Z., ADAMS, R. P., AND DE FREITAS, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE* 104, 1 (2015), 148–175.
- [22] SHI, ET AL. Optimized Compilation of Aggregated Instructions for Realistic Quantum Computers. ASPLOS.
- [23] SMITH, K. N., AND THORNTON, M. A. A Quantum Computational Compiler and Design Tool for Technology-Specific Targets. ISCA.
- [24] SUN, ET AL. Experimental Quantum Error Correction with Binomial Bosonic Codes. In *APS Meeting Abstracts* (2019).
- [25] TANNU, ET AL. Ensemble of Diverse Mappings: Improving Reliability of Quantum Comps. by Orchestrating Dissimilar Mistakes. MICRO.
- [26] TANNU, ET AL. Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias. MICRO.
- [27] TANNU, ET AL. Not All Qubits are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers. ASPLOS.
- [28] TANNU, S. S., MYERS, Z. A., NAIR, P. J., CARMEAN, D. M., AND QURESHI, M. K. Taming the instruction bandwidth of quantum computers via hardware-managed error correction. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2017), IEEE, pp. 679–691.
- [29] TERHAL, ET AL. Scalable Quantum Error Correction with the Bosonic GKP Code. In *APS Meeting Abstracts* (2019).
- [30] VUILLOT, ET AL. Quantum Error Correction with the Toric Gottesman-Kitaev-Preskill Code. *Physical Review A* 99, 3 (2019).
- [31] WILLE, ET AL. Mapping Quantum Circuits to IBM QX Architectures Using the Minimal Number of SWAP and H Operations. DAC.
- [32] ZHANG, P. Model Selection via Multifold Cross Validation. *The Annals of Statistics* (1993), 299–313.
- [33] ZULEHNER, ET AL. An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. *IEEE TCAD* (2018).
- [34] ZULEHNER, A., AND WILLE, R. Compiling SU (4) Quantum Circuits to IBM QX Architectures. ASPDAC.