Efficient Readout Error Mitigation Using Singular Value Decomposition

Bo Yang[1] and Rudy Raymond[2]
[1] Graduate School of Information Science and Technology, The University of Tokyo
[2] IBM Quantum, IBM Research Tokyo

Nowadays, the near-term quantum devices over 50 qubits are available and those of hundreds of qubits are expected to be realizable in several years. Mitigating their errors is one of the most important issues to exploit the better performance on such noisy devices. Readout error in the measurement process may have the biggest error rate among all errors, that can be mitigated by post-processing on classical computers. The standard readout error mitigation methods correct the measured probability distribution using the inverse of calibration stochastic matrices. However, such methods require computational resources exponential in the number of qubits, and are not applicable for devices with tens or more qubits. Here we investigate the polynomial time and memory classical post-processing based on the tensor product noise model. Our proposed algorithm requires $O(ns^2)$ time and $O(s)$ memory with n qubits and s shots. In order to meet the physical constraint for the probability distribution, our algorithm uses Lagrangian multiplier on the singular value decomposition of calibration matrices, which analytically finds the closest solution. This idea might help us theoretically upper bound the closeness of mitigated vector to the ideal one. The current version of our proposed algorithm is targeted to mitigate the output probability distribution with few expected labels, which should be further explored and improved for the wider applications. To study the performance of the proposed algorithm, we report the result of numerical simulation on the modified Grover-search algorithms which are important for near-term applications of quantum devices. Our proposed algorithm shows almost the same performance as the conventional standard algorithms on 10-qubits system and is also likely to well mitigats the results on 20-qubits system where the standard algorithms would take prohibitively long time.

I. INTRODUCTION

These years, many near-term quantum devices are open to public [1–5]. The number and quality of qubits has been continuously improved and the superconducting quantum machine over 50 qubits is available at IBM Quantum Experience now. In addition, the quantum devices with hundreds of qubits are expected to be realizable in several years. Since such near-term devices are still too noisy and small to incorporate quantum error correction methods for protecting quantum information [6–9], various error mitigation methods [10–14] have emerged for obtaining better results.

Among many types of errors on near-term devices, the error occurs in the final measurement process, which is called readout error, is one of the biggest factors. The measurement process of n-qubit is to obtain a probability vector labeled by n-bit string and the imperfect qubit measurements would flip some of bits in the output string. This measurement noise can be modeled as a $2^n \times 2^n$ stochastic matrix mapping a probability vector under ideal measurement process to one affected by measurement noise. Such stochastic matrix is made from the outcomes of calibration circuits and thus be also called as calibration matrix. Under the tensor product noise model assuming the measurement noise acts independently on each qubit, the calibration matrix of n-qubit outcome is described as a tensor product of calibration matrix of each qubit.

A simple way to mitigate this error is to apply the inverse of the calibration matrix to the probability vector at the post-processing on classical computers. Since the calibration matrix is sized $2^n \times 2^n$ to n-qubit measurement outcomes, this post-processing takes exponential time and memory, which makes this readout error mitigation lose scalability to the large qubit measurements. In addition, just applying the inverse matrix to probability vector would result in a vector which might not meet the physical requirements that the output vector should represent valid probability. Using optimization methods such as sequential least squares programming provided by the open-source software package Qiskit [5] to find the optimal physically proper probability vector is more time-consuming task.

Towards these problems, we propose a faster error mitigation method based on the tensor product noise model. Our algorithm analytically computes the optimal probability vector in two steps to satisfy the physical constraints that the sum of the vector must be 1 and the all elements of the vector must be non-negative. First, to meet the "sum to one" condition of probability vector, we correct the roughly mitigated vector using Lagrangian multiplier under the singular value decomposition (SVD) of the calibration matrices. Then, we use the technique by Smolin, Gambetta, and Smith [10] to make the corrected vector meet the "non-negativity" condition. Note that their algorithm also uses Lagrangian multiplier to analytically cancelling the negativity in the elements of the vector under "sum to one" condition.

Since the n-qubit measurement outcomes are the n-bit strings, it can take 2^n possible labels which results in the exponential time and memory classical processing.

To counter this problem, we set an assumption that the labels appearing in the final probability vector always appear in the raw probability vector. Then our algorithm requires $O(ns^2)$ time and $O(s)$ memory with n qubits and s shots, wihle the assumption makes the accuracy analysis difficult and also restricts the target of our algorithm to the measurement result with only few labels in expected outcomes. These drawbacks should be improved in the further works.

We also tested our approach to the modified Grover algorithm [15] using noisy simulator provided by Qiskit [5]. The modified Grover algorithm is one of the variation of maximum likelihood amplitude estimation (MLAE) algorithm [16] achieving a higher tolerance to depolarization errors. In the modified Grover algorithm, only the probability of $|0\rangle$ state is focused and expected to have a higher probability, which is suitable for incorporating our algorithm to reduce readout error. We show our algorithm is likely to well mitigate the readout error in 20-qubit system to which the naive error mitigation methods in the current Qiskit libraries are not applicable.

The rest of the paper is organized as follows. Section II.A explains the noise model of readout error we employ, and section II.B explains our proposed error mitigation method. Section III shows the results of numerical simulation for modified Grover algorithm on Qiskit simulator. Section IV concludes with the discussion of the results and future works.

II. METHODS

A. Tensor Product Noise Model

The measurement process is modeled as a stochastic calibration matrix A, whose elements account to the probability describing how well the state is preserved through the measurement. The noises in calibration matrix A include local measurement error and cross-talk error among two or more qubits. Here we only consider the only consider the local measurement error by adopting tensor product noise model which assumes the calibration matrix A is approximated by tensors (Kronecker product) of matrices for each qubits: $A = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. As Mooney et al. shows [17], the cross-talk error is much smaller than the local measurement error on the current near-term IBM Quantum devices, this error model seems to be valid in the current and near future situations.

For the readout error mitigation protocol as a classical post-processing, we consider the following problem. First, we are given a probability vector $y \in \mathbb{R}^N$ as an n-qubit measurement outcome such that $y \geq 0$ and $1^T \cdot y = \sum_j y_j = 1$ where $N = 2^n$. Then we want to recover the *true* probability vector $\tilde{x} \in \mathbb{R}^N$.

B. Proposed Algorithms

Let $y \in \mathbb{R}^N$ be the measured probability vector and $x \in \mathbb{R}^N$ as $x := A^{-1}y$. Since the inverse of A may not be a stochastic matrix, x would include negative values. We call this x "roughly" mitigated vector. First, each element of the roughly mitigated vector x is computed by the following algorithm 1 in $O(ns)$ time with the number of shots s. In total, computing x rigorously requires $0(n2^n s)$ time for 2^n labels in x. The assumption explained at II B 3 will reduce this complexity to $O(ns^2)$. Here we want to find a valid probability vector $\tilde{x}$ which

Algorithm 1 Mitigate One State (mitigate_one_state)

Require: target state label t, probability vector y
Ensure: mitigated count c

```
c ← 0
for source state label s in y do
    product p ← 1
    for calibration matrix A^k in {A^0, ···, A^{n−1}} do
        i ← the value of k-th digit of t (0 or 1)
        j ← the value of k-th digit of s (0 or 1)
        p ← p · A^k_{i,j}
    end for
    c ← c + p · y[s]
end for
return c
```

is the closest one to x, satisfying the following condition.

$$1^T \cdot \tilde{x} = 1 \tag{1}$$

$$\tilde{x}_i \geq 0 \text{ for all } i \tag{2}$$

Our idea is to first use the singular value decomposition (SVD) to A, which is constructed by a tensor product of small matrices. The SVD of A can be efficiently performed because $(U_1 \Sigma_1 V_1^\dagger) \otimes (U_2 \Sigma_2 V_2^\dagger) = (U_1 \otimes U_2)(\Sigma_1 \otimes \Sigma_2)(V_1^\dagger \otimes V_2^\dagger)$ holds for the SVD of two matrices $A_1 = U_1 \Sigma_1 V_1^\dagger$ and $A_2 = U_2 \Sigma_2 V_2^\dagger$. Then, in order to use the algorithm by Smolin, Gambetta, and Smith [10] which finds the closest vector satisfying (2), we use Lagrange multiplier to find a correction vector added to x for the constraint (1), focusing on the singular values of A.

1. *Step1: Sum-to-one Constraint*

Here we will first find a vector $\hat{x}$ which only satisfy the condition (1). In order to correct the roughly mitigated vector x, we consider adding a correction vector Δ to x. Let $\hat{x}$ be $\hat{x} = x + \Delta$. What we want to minimize is the difference of $A\hat{x}$ and $y (= Ax)$:

$$\|A\hat{x} - y\|^2 = \|A(x + \Delta) - y\|^2 = \|A\Delta\|^2 \tag{3}$$

Suppose that A is decomposed into $A = U\Sigma V^T$ by singular value decomposition (SVD) as below.

$$A = U\Sigma V^T = \sum_{i=0}^{N-1} \sigma_i u_i v_i^T \tag{4}$$

Using the columns v_i of the left singular matrix V as a basis to represent Δ, $\Delta = \sum_{i=0}^{N-1} \Delta_i v_i$. If we just choose k vector components for Δ, that is $\Delta = \sum_{i=j_0}^{j_{k-1}} \Delta_i v_i$, the target value to be minimized is equivalent to

$$\|A\hat{x} - y\|^2 = \|A\Delta\|^2 = \sum_{i=j_0}^{j_{k-1}} \sigma_i^2 \Delta_i^2 \tag{5}$$

where the indices are re-aligned. As a result, the constrained optimization we are focusing on is:

$$\begin{aligned} &\min_{\Delta\in\mathbb{R}^N} \sum_{i=j_0}^{j_{k-1}} \sigma_i^2 \Delta_i^2 \\ &\text{s.t.} \sum_{i=j_0}^{j_{k-1}} \left(1^T \cdot v_i\right) \Delta_i = 1 - 1^T \cdot x \end{aligned} \tag{6}$$

This constrained optimization problem can be rigorously solved by Lagrange multiplier.

$$L := \sum_{i=j_0}^{j_{k-1}} \sigma_i^2 \Delta_i^2 - \lambda \left(\sum_{i=j_0}^{j_{k-1}} \left(1^T \cdot v_i\right) \Delta_i - \left(1 - 1^T \cdot x\right) \right) \tag{7}$$

Differentiating L with respect to Δ_i, we obtain

$$\frac{\partial L}{\partial \Delta_i} = 2\sigma_i^2 \Delta_i - \lambda \left(1^T \cdot v_i\right). \tag{8}$$

This implies the values of Δ_i's at optimality

$$\Delta_i = \lambda \frac{\left(1^T \cdot v_i\right)}{2\sigma_i^2}. \tag{9}$$

Then the value of λ can be computed from the condition imposed by Eq. (6).

$$\lambda = \frac{2\left(1 - 1^T \cdot x\right)}{\sum_{i=j_0}^{j_{k-1}} \frac{\left(1^T \cdot v_i\right)^2}{\sigma_i^2}}. \tag{10}$$

Summarizing, we finally obtain the elements of correction vector $\Delta = \Delta_{j_0} v_{j_0} + \Delta_{j_1} v_{j_1} + \cdots \Delta_{j_{k-1}} v_{j_{k-1}}$ as

$$\Delta_i = \frac{1 - 1^T \cdot x}{\sum_{l=j_0}^{j_{k-1}} \frac{\left(1^T \cdot v_l\right)^2}{\sigma_l^2}} \cdot \frac{1^T \cdot v_i}{\sigma_i^2}. \tag{11}$$

Using these values, we finally get the corrected vector $\hat{x} = x + \Delta$. This step naively requires $O(n2^n)$ to the number of qubits n. We avoid this cost by introducing the assumption explained at II B 3.

2. *Step2: Negative Cancelling*

Next, we are going to delete the negative values in $\hat{x}$. In this process, we can use the algorithm by Smolin, Gambetta, and Smith [10] which assumes the input vector satisfy the condition (1) (hereinafter, this is called "SGS algorithm"). The procedure of SGS algorithm is described at the following pseudo code. Through SGS algorithm, the finally mitigated probability vector $\tilde{x}$ (= sgs_algorithm($\hat{x}$)) is computed in $O(N \log N)$ time where N is the number of elements in $\hat{x}$.

Algorithm 2 Negativity Cancellation by Smolin, Gambetta, and Smith [10] (sgs_algorithm)

```
Require: vector x̂ (satisfying 1^T x̂ = 1)
Ensure: mitigated probability vector x̃
  queue ← make a priority queue
  accumulator of positive values p ← 0
  for state label s in y do
      if x̂[s] > 0 then
          queue.push(x̂[s])
          p ← p + x̂[s]
      end if
  end for
  accumulator of negative values neg ← 1 − p
  while queue is not empty do
      if (queue.top() + neg)/queue.size() < 0 then
          neg ← queue.pop()
      else
          break
      end if
  end while
  mitigated counts x̃ ← empty dictionary
  division value of negative accumulator d ← queue.size()
  while queue is not empty do
      x̃[s] ← x̂[s] + neg/d
  end while
  return x̃
```

3. *Assumption*

When mitigating the probability vector of n-qubit result, storing all 2^n labels is both time and space consuming. In order to run the error mitigation faster and memory efficiently, we introduce an assumption that the labels appearing in the mitigated vector always appear in the measured probability vector. In other words, only the the labels appearing in the elements of y are required in the mitigation process.

Let S be the label set of measured probability vector y and let the vector with subscript $_S$ be the vector which only contain the elements labeled by the labels in S (e.g. x_S). Using these notations, the assumption introduced above means the finally mitigated probability vector is $\tilde{x}_S$. Note that the size of S meets $|S| < s$

where s is the number of shots. When we just choosing the largest factor in correction vector Δ, the proposed efficient mitigation process is described as follows: The $x_S = (A^{-1}y)_S$ can be computed through algorithm 1 in $O(ns^2)$ time, which is one of the most time consuming parts. Then $\Delta_S = (\max_{\Delta_i\|v_i\|} \Delta_i v_i)_S$ is also computed in $O(ns^2)$ time. After adding x_S and Δ_S in $O(s)$ time, resulting in $\hat{x}$, the finally mitigated vector $\tilde{x}$ is computed by SGS algorithm 2 [10] in $O(s\log s)$. Therefore, the total computational complexity of our proposed algorithm is $O(ns^2)$.

III. NUMERICAL EXPERIMENTS

Since bounding precision of our algorithm seems hard to be analyzed, we performed numerical simulation to study the performance. One of the application may be the modified Grover algorithm by Uno et al. [15].

1. *Modified Grover Algorithm*

The modified Grover algorithm [15] is a variation of the maximum likelihood amplitude estimation (MLAE) algorithm [16], which avoids the quantum Fourier transform used in conventional amplitude estimation methods. The modified Grover algorithm is more robust to the depolarization error than the original MLAE algorithm and only the outcome of $|0\rangle$ state is expected to have a higher probability. The modified amplitude amplification operator is represented as $Q = U_0 A^\dagger U_f A$, where U_0 and U_f are the reflection operators defined as

$$\begin{aligned} U_0 &= -\mathbf{I}_{n+1} + 2|0\rangle_{n+1}\left\langle 0\right|_{n+1}, \\ U_f &= -\mathbf{I}_{n+1} + 2\mathbf{I}_n \otimes |0\rangle\langle 0|. \end{aligned} \tag{12}$$

The initial state $|0\rangle_{n+1}$ after m iterations of operator Q becomes

$$\begin{aligned} \left|\psi_Q\left(\theta, m\right)\right\rangle &:= Q^m|0\rangle_{n+1} \\ &= \cos\left(2m\theta\right)|0\rangle_{n+1} + \sin\left(2m\theta\right)|\phi\rangle_{n+1} \end{aligned} \tag{13}$$

where $|\phi\rangle_{n+1} \neq |0\rangle_{n+1}$. In order to approximate the value of θ, only the probability of getting state $|0\rangle_{n+1}$ is required. According to [16], the estimation error would be lower bounded by the Fisher information, which follows the Heisenberg limit that the error decreases in the order of $O(1/m)$ for m rounds of Grover iterations. The advantage of [16] can be checked by seeing whether the estimation error follows the Heisenberg limit or not.

2. *Monte Carlo Integration*

We applied this modified Grover algorithm to Monte Carlo integration, following the procedures in [16]. Using the expression in [16], the goal of the Monte Carlo integration is to compute the following value

$$\begin{aligned} I &= \frac{1}{b_{\max}}\int_0^{b_{\max}} \sin(x)^2 dx \\ &= \frac{1}{b_{max}}\left(\frac{b_{max}}{2} - \frac{1}{4}\sin(2b_{max})\right) \end{aligned} \tag{14}$$

where b_{max} is a constant that determines the upper limit of the integral. This value can be estimated via the modified Grover operations above and we numerically checked the gap of estimation error to the Heisenberg limit.

3. *Results*

The numerical simulation was performed with 10-qubit and 20-qubit search space respectively on the Qiskit simulator [5]. Here we run the modified Grover operators for $m = 0, 1, 2, \cdots, 99$ times with $b_{max} = \pi/100$. Since our algorithm scales quadratic to the number of shots, the number of shots is set to $N_{shot} = 100$. Besides, we tested different readout error rates $p(0 \to 1)$ and $p(1 \to 0)$ among $0.01, 0.03$, and 0.05 assuming $p(0 \to 1) = p(1 \to 0)$. The parameters used in the numerical simulation are listed in Table I.

TABLE I. List of parameters used in the numerical simulation.

number of qubits	n	$\{10, 20\}$
number of shots	N_{shot}	100
number of Grover iteration	m	$\{0, 1, 2, \cdots, 99\}$
target values	$I = \cos^2\theta$	$b_{max} = \pi/100$
readout noise	$p(0 \to 1)$ $p(1 \to 0)$	$\{0.01, 0.03, 0.05\}$

The results of the numerical simulation are shown in Fig. 1. For the 10-qubit system, we can see the estimation errors of mitigated plots are more closed to the Heisenberg limit than the plots containing readout error for all readout error rates. In the case of $p(0 \to 1) = p(1 \to 0) = 0.05$, the readout error might become so large that the plot with readout errors (the orange dots) is not likely to follow the Heisenberg limit.

The purple, green, and red plots are almost at the same position on each figure. Since the the purple plot means the standard method through the rigorous inverse calibration matrix, our algorithm can be seen as good approximation to such standard way.

For the 20-qubit system, we can also see the estimation errors of mitigated plots are well following the Heisenberg limit, while the noisy plots colored by orange are not following it at the readout error rate 0.03 and 0.05. This means one need readout error mitigation to obtain the advantageous results from modified Grover algorithm. In the size of 20-qubit, the conventional exponential time methods would not finish in practical time. On the other hand, our algorithm still outputs the plots which follows

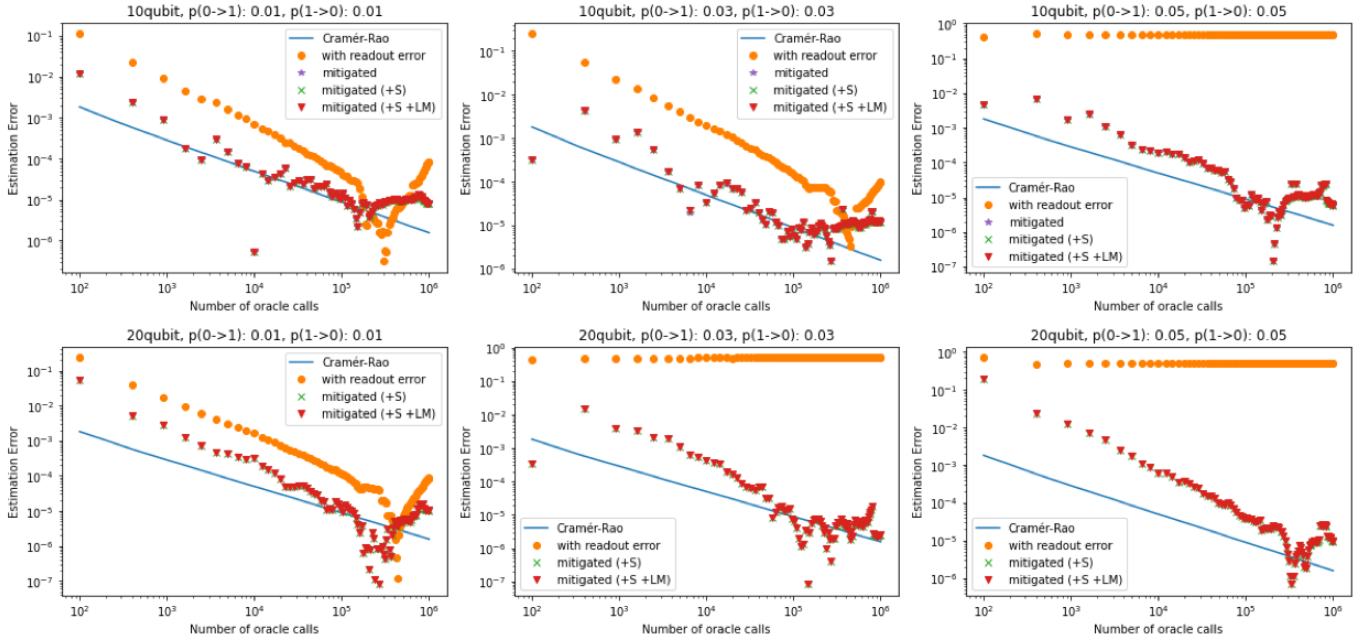

FIG. 1. The estimation error of Monte Carlo integration on 10-qubit and 20-qubit system. The blue curves are the lower bound of estimation error in terms of the Fisher information. The orange plots are the results of noisy circuit simulation with readout error. The scale of readout errors was changed among $0.01, 0.03, 0.05$. The label "mitigated" means the mitigated results applying rigorous inverse of calibration matrix and SGS algorithm. The label "mitigated (+S)" added assumption explained at II B 3 to the method of "mitigated". The label "mitigated (+S +LM)" added the Lgarangian multiplier based on the SVD of calibration matrix to the method of "mitigated (+S)".

the Heisenberg limits. Since the results by conventional method and the results of our methods are very closed for 10-qubit system, we guess the output by conventional methods for 20-qubit system would also closed to the results by our algorithms, meaning the potential advantage of our algorithm.

IV. CONCLUSION

Our algorithm mitigates the readout error with $O(ns^2)$ time and $O(s)$ memory with n qubits and s shots through the post-processing on classical computers. Fixing the number of shots, our algorithm scales linearly to the number of measured qubits, which provides a scalable readout error mitigation tool for the current and near future quantum devices with larger qubits. Besides, every step in our algorithm can be implemented in parallel to the vector elements. Therefore, our algorithm has the potential to get more accelerated with the help of GPU.

The numerical simulation on the modified Grover algorithm shows our algorithm has good performance even on the system with larger qubits towards the mitigation of modified Grover algorithm with readout errors. The the estimation error of mitigated results on 20-quibt system still follows the Heisenberg limit, which might support the advantage of our algorithm when the size of qubit becomes large.

Some points in our current algorithm are to be further improved and those are still work in progress. First, the precision of our current algorithm is not theoretically bounded due to the assumption we set in our algorithm that the labels appearing in the mitigated vector always appear in the measured probability vector. The remaining task includes the analysis of the closeness between the mitigated probability distribution and the ideal probability distribution. In addition, this assumption also restricts the application range of our algorithm to the measurement result with only few labels in expected outcomes because there might be much more source labels than those are measured if the number of shots are much smaller than the number of possible state labels. To better utilize our proposed algorithm, one can find more practical cases where this assumption is valid to be applied. Furthermore, one could also consider other assumptions such as the situation where the measured state string contains at most one or two errors.

Applying our algorithm to a different noise model is also a possible future direction. The tensor product noise model can only capture the local measurement error although local error is dominant in the current near-term quantum devices [17]. Instead, the correlated Markovian noise model may include cross-talk errors. Using the variation of this noise model called continuous time Markov processes (CTMP), Bravyi et al. [12] have recently developed more efficient readout error mitigation protocol to directly compute expectation values form measured probability vectors. Since our algorithm aims to correct the probability vector itself, one might incorporate the method in [12] to develop our algorithm to the CTMP noise model.

[1] P. Jurcevic, A. Javadi-Abhari, L. S. Bishop, I. Lauer, D. F. Bogorin, M. Brink, L. Capelluto, O. Günlük, T. Itoko, N. Kanazawa, A. Kandala, G. A. Keefe, K. Krsulich, W. Landers, E. P. Lewandowski, D. T. McClure, G. Nannicini, A. Narasgond, H. M. Nayfeh, E. Pritchett, M. B. Rothwell, S. Srinivasan, N. Sundaresan, C. Wang, K. X. Wei, C. J. Wood, J.-B. Yau, E. J. Zhang, O. E. Dial, J. M. Chow, and J. M. Gambetta, Demonstration of quantum volume 64 on a superconducting quantum computing system (2020), arXiv:2008.08571 [quant-ph].
[2] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, Demonstration of the qccd trapped-ion quantum computer architecture (2020), arXiv:2003.01293 [quant-ph].
[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. Dunsworth, D. Eppens, E. Farhi, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, S. Habegger, M. P. Harrigan, A. Ho, S. Hong, T. Huang, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, C. Jones, D. Kafri, K. Kechedzhi, J. Kelly, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, M. Lindmark, M. Leib, E. Lucero, O. Martin, J. M. Martinis, J. R. McClean, M. McEwen, A. Megrant, X. Mi, M. Mohseni, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, F. Neukart, H. Neven, M. Y. Niu, T. E. O'Brien, B. O'Gorman, E. Ostby, A. Petukhov, H. Putterman, C. Quintana, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, A. Skolik, V. Smelyanskiy, D. Strain, M. Streif, K. J. Sung, M. Szalay, A. Vainsencher, T. White, Z. J. Yao, P. Yeh, A. Zalcman, and L. Zhou, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor (2020), arXiv:2004.04197 [quant-ph].
[4] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, and et al., Science **369**, 1084–1089 (2020).
[5] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Łukasz Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya, and C. Zoufal, Qiskit: An Open-source Framework for Quantum Computing (2019).
[6] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).
[7] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[8] A. Kitaev, Annals of Physics **303**, 2–30 (2003).
[9] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Physical Review X **10**, 10.1103/physrevx.10.011022 (2020).
[10] J. A. Smolin, J. M. Gambetta, and G. Smith, Physical Review Letters **108**, 10.1103/physrevlett.108.070502 (2012).
[11] K. Temme, S. Bravyi, and J. M. Gambetta, Physical Review Letters **119**, 10.1103/physrevlett.119.180509 (2017).
[12] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, Phys. Rev. A **103**, 042605 (2021).
[13] G. S. Barron and C. J. Wood, Measurement error mitigation for variational quantum algorithms (2020), arXiv:2010.08520 [quant-ph].
[14] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, Physical Review Letters **121**, 10.1103/physrevlett.121.220502 (2018).
[15] S. Uno, Y. Suzuki, K. Hisanaga, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Modified grover operator for amplitude estimation (2020), arXiv:2010.11656 [quant-ph].
[16] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Quantum Information Processing **19**, 10.1007/s11128-019-2565-2 (2020).
[17] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, Generation and verification of 27-qubit Greenberger-Horne-Zeilinger states in a superconducting quantum computer (2021), arXiv:2101.08946 [quant-ph].

