Rotation-inspired circuit cut optimization

Gideon Uchehara, Tor M. Aamodt, Olivia Di Matteo
Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
{ gideon.uchehara, aamodt, olivia }@ece.ubc.ca

***Abstract*—Recent works have demonstrated that large quantum circuits can be cut and decomposed into smaller clusters of quantum circuits with fewer qubits that can be executed independently on a small quantum computer. Classical post-processing then combines the results from each cluster to reconstruct the output of the original quantum circuit. However, the runtime for such hybrid quantum-classical algorithms is exponential in the number of cuts on a circuit. We propose Rotation-Inspired Circuit Cut Optimization (RICCO), an alternative method which reduces the post-processing overhead of circuit cutting, at the cost of having to solve an optimization problem. RICCO introduces unitary rotations at cut locations to rotate the quantum state such that expectation values with respect to one set of observables are maximized and others are set to zero. We demonstrate practical application of RICCO to VQE by classically simulating a small instance of VQE and comparing it to one of the existing circuit-cutting methods.**

***Index Terms*—Quantum, circuit, cutting, rotation, optimization**

I. Introduction

Simulating quantum circuits with a large number of qubits is intractable on a classical computer. Also, it is difficult to program these circuits on actual quantum hardware due to size constraints (insufficient qubits) and because they are also error-prone. To increase the use of current noisy small-scale quantum devices, methods have been developed to combine classical and quantum computers to simulate quantum circuits with a large number of qubits [1], [3]–[5].

For example, Peng et al. [1] demonstrated that a quantum circuit can be divided into subcircuits, each with fewer qubits, which can then be run separately on a quantum computer too small to run the original circuit. The quantum computer's results are sent to a classical computer, which combines them to replicate the expected output of the original quantum circuit. The number of quantum measurements and traditional post-processing involved in replicating the original quantum circuit increases with the number of cuts. Specifically, the cost of executing the original quantum circuit is exponential in the number of cuts. Tang et al. [4] proposed CutQC, a framework that employs heuristics to select a cut reducing classical post-processing overhead. To reduce classical post-processing resources necessary to characterize circuit cutting, Perlin et al. [3] proposed maximum-likelihood fragment tomography (MLFT). More recently, Lowe et al. [5] employed a randomized measurements method to speedup quantum circuit cutting. We refer to these procedures generically as *quantum circuit cutting*.

In this paper we propose Rotation-Inspired Circuit Cut Optimization (RICCO), a method to reduce the cost of simulating a large quantum circuit on a small quantum computer. RICCO introduces a parameterized unitary operator and its adjoint at each cut location. After optimization, the unitary operator rotates a given quantum state such that it maximizes the expectation value of select Pauli observables. This is similar to rotating a quantum state vector such that it aligns with one of the Bloch sphere axes for a 1-qubit system. By this method, we effectively set the expectation value of the select Pauli observable(s) to maximum while the others are set to zero. This is relevant because we have eliminated the need to measure in multiple bases, as is the case in the original method, resulting in fewer circuit executions. We demonstrate through simulation that when compared to Peng et al. [1] RICCO reduces the total number of quantum circuit measurements required to reconstruct a cut circuit, at the expense of the optimization.

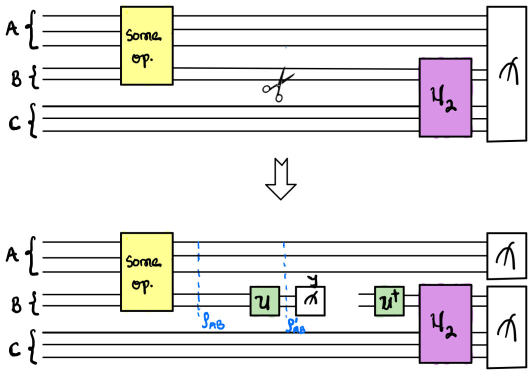
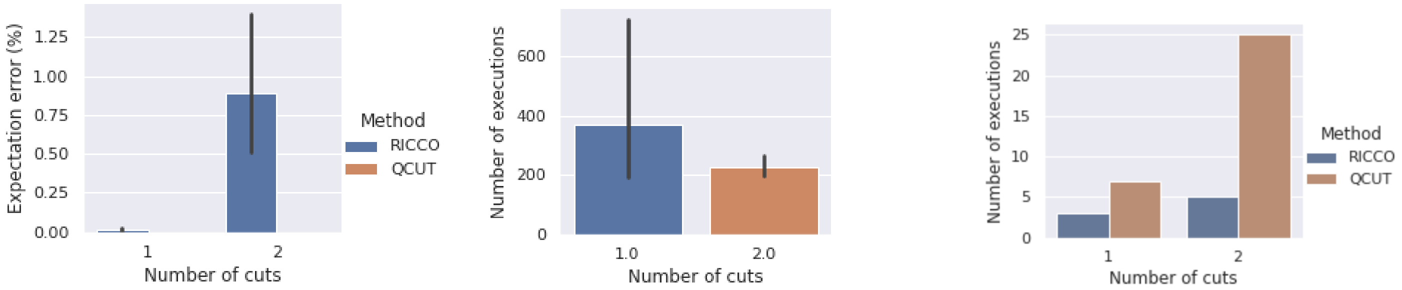


Fig. 1: Quantum circuit with unitary operator U applied at the cut location to rotate the state ρ_{AB} to ρ'_{AB}

II. Theory

Consider the quantum circuit in Fig. 1. A cut is applied on the qubits labeled B resulting in two separate subcircuits: upstream and downstream subcircuits. The ultimate goal is to combine the two subcircuits to reconstruct the output of the original uncut circuit. Our focus at the moment will be on the upstream subcircuit with the qubit register labeled A and the parts of qubit register B just before the cut. The downstream



(a) Expectation value error of the observable $Z^{\otimes n} = Z^{\otimes(n_A+n_B+n_C)}$ for random circuits. The error bar indicates a deviation of less than 0.3% for two qubit cuts

(b) Circuit executions during optimization. The error bar is high because of the variance in the class of quantum circuits optimized using RICCO

(c) The number of executions of quantum circuits for RICCO is significantly less than that for QCUT after optimization.

Fig. 2: Empirical results showing the performance of RICCO vs. QCUT for quantum circuits formed from random unitaries as the circuit in Fig. 4.

subcircuit consists of the part of qubit register B just after the cut and beyond and the qubit register labeled C.

In prior circuit cutting methods, measurements are made in multiple bases at the cut point, and specific eigenstates prepared to subsequently run the downstream circuits. This results in an exponentially large number of measurements and circuits. The goal of our proposed method is to find a way to reduce the number of measurements.

Without loss of generality, we are interested in computing the expectation value of some observable, say, $Z^{\otimes n} = Z^{\otimes(n_A+n_B+n_C)}$ (this also applies to observables with some Pauli X and Y, in which case we just apply the corresponding local basis rotations) for the circuit in Fig. 1, where n_A, n_B and n_C are the number of qubits for register labels A, B and C respectively. To do this, we seek a unitary operator U that rotates the state ρ_{AB} to the state ρ'_{AB} such that ρ'_{AB} aligns with one of the computational basis vectors of the observables in $\{\{Z^{\otimes n_A}\}\otimes\{I, Z\}^{n_B}\}$. As a result, we only have to measure in the computational basis. To find U, we decompose U into parameterized rotations and optimize for its parameters in the quantum circuit.

To understand the intuition behind our method, consider that the density matrix ρ of an arbitrary 1-qubit quantum state can be expressed in terms of Pauli matrices,

$$\rho = \frac{1}{2}\sum_i Tr(P_i\rho)P_i. \tag{1}$$

Here $P_i \in \{I, X, Y, Z\}$, and $\mathrm{Tr}(P_i\rho)$ is the expectation value, $\langle P_i\rangle$ of the Pauli P_i. By applying the unitary rotation U to ρ we transform ρ to the state ρ' as follows:

$$\rho \mapsto \rho' = U\rho U^\dagger. \tag{2}$$

Since ρ' is also a density matrix, we can express it as

$$\rho' = U\rho U^\dagger = \frac{1}{2}\sum_i Tr(P_iU\rho U^\dagger)P_i. \tag{3}$$

The first intuition behind our method is the Bloch sphere. The expectation values $\langle X\rangle$, $\langle Y\rangle$, and $\langle Z\rangle$ of the three Pauli matrices, allow one to identify the three coordinates with respect to X, Y, and Z axes. If we can find a unitary operator that rotates the Bloch vector of the density matrix to align with a particular *state* (e.g., the state with positive eigenvalue) of the Z-axis, we are by implication zeroing out the expectation values of Pauli X and Y observables. To avoid restricting ourselves to one eigenstate of Pauli Z, we can find a unitary, U such that when applied to ρ, its final state ρ' ends up along either the $|0\rangle$ axis ($\langle Z\rangle = +1$) or $|1\rangle$ axis ($\langle Z\rangle = -1$).

The second intuition behind our method stems from the theory of optimal measurements and mutually unbiased bases (MUBs). MUBs are defined such that for any two vectors $|\mu\rangle$ and $|\nu\rangle$ from bases M and N,

$$|\langle\nu|\mu\rangle|^2 = \frac{1}{d}, \tag{4}$$

where d is the dimension of the system. MUBs for an n-qubit system have a standard construction based on the partitioning of the Pauli group into 2^n+1 disjoint sets of 2^n-1 commuting operators whose mutual eigenvectors form the measurement bases [6]. If we align our state with an eigenvector of one such basis, it will have equal overlap with all vectors in the other sets. This has further implications for the expectation values of Paulis: the state will have a non-zero expectation value for all Paulis in that set we measure with respect to, while for Paulis from other sets, they will be 0. This enables us to perform only a single measurement in the Z basis.

A. Cost Function to Determine the Unitary Operator

Let's consider the case where ρ is a single-qubit state. We want to rotate the state such that we point along either the $|0\rangle$ or $|1\rangle$ axis of the Bloch sphere; these are eigenstates of only Pauli Z, so we can ignore any observable, $P_i \notin \{I, Z\}$. The identity operator is kept because $|0\rangle$ and $|1\rangle$ are also the eigenstates of I.

To design the right cost function, that optimizes the parameters of a unitary operator U such that it rotates ρ to a new state ρ', we can use a measure of distance such as fidelity [11]:

$$F\left(|\psi\rangle, \rho'\right) = \langle\psi| \rho' |\psi\rangle, \tag{5}$$

to determine how close ρ' is to our desired state $|\psi\rangle$. We decided to use fidelity in this case because it has a straightforward definition for the case of pure versus mixed state. We want to see how close our transformed mixed state ρ' from the cut is to the desired pure state (a computational basis state). The best possible outcome is $F = 1$, when ρ' is the same state as the desired state $|\psi\rangle\langle\psi|$. The worst is $F = 0$ when the desired state is perfectly orthogonal to ρ'.

Another nice property of fidelity is that it can be expressed in terms of expectation values as [10], [12]

$$\begin{aligned} F\left(|\psi\rangle, \rho'\right) &= \langle\psi| \left(\frac{1}{2}\sum_i Tr(P_i U\rho U^\dagger)P_i\right)|\psi\rangle, \\ &= \frac{1}{2}\sum_i Tr(P_i\rho')\langle\psi| P_i |\psi\rangle, \\ &= \frac{1}{2}\sum_i \langle P_i\rangle_{\rho'}\langle P_i\rangle_{|\psi\rangle}. \end{aligned} \tag{6}$$

These expectation values are what we measure in RICCO (see Algorithm 1). To determine U, we have to build a cost function that maximizes the fidelity with respect to either the $|0\rangle$ or $|1\rangle$ state. We propose to use the cost function

$$C\left(\rho'\right) = \left[1 - F\left(|0\rangle, \rho'\right) F\left(|1\rangle, \rho'\right)\right]^2 \tag{7}$$

This cost function is simply the squared difference between 1 and the product of the fidelities of $|0\rangle$ and $|1\rangle$ with ρ'. We squared the cost function to ensure that it is a convex function for optimization purposes.

Thus, for the one qubit case, our cost function is minimized when $\langle Z\rangle_{\rho'} = \pm 1$. It is maximized when $\langle Z\rangle_{\rho'} = 0$, which corresponds to states along the other two axes X and Y. Thus, by minimizing this cost function, we are effectively setting the expectation values of X and Y to zero.

For the general case, to zero out as many expectation values as possible, we need to rotate only the qubits being cut so that ρ' aligns with any of the computational basis vectors. Then we can measure the upper subcircuit in only the computational basis. It is important to note that as a result of optimization error, the other uncut qubits of ρ may not align perfectly with the computational basis vectors of their observables. Our results show that this error is negligible in most cases and does not constitute a major limitation in reconstructing the original expectation value. For n qubits, we can express the density matrix of an arbitrary state ρ' as in (8),

$$\rho' = \frac{1}{2^n}\sum_{P_i \in \{I,X,Y,Z\}^n} Tr(P_i\rho')P_i. \tag{8}$$

The formula for the fidelity is the same, except now we must consider all observables that consist only of Zs and Is for the cut qubit. The uncut qubits still retain their respective Pauli observables. For n_B cut qubits, the observables are in the set $\{I, Z\}^{n_B}$. The fidelity of the transformed state ρ' and one of the computational basis states $|\psi\rangle$ of the combined observables $P_i \in P$ (where $P = \{\{Z^{\otimes n_A}\} \otimes \{I, Z\}^{n_B}\}$) of the cut qubits, n_B and the uncut qubits, n_A is

$$F\left(|\psi\rangle, \rho'\right) = \frac{1}{2^n}\sum_{P_i \in P}\langle P_i\rangle_{\rho'}\langle P_i\rangle_{|\psi\rangle} \tag{9}$$

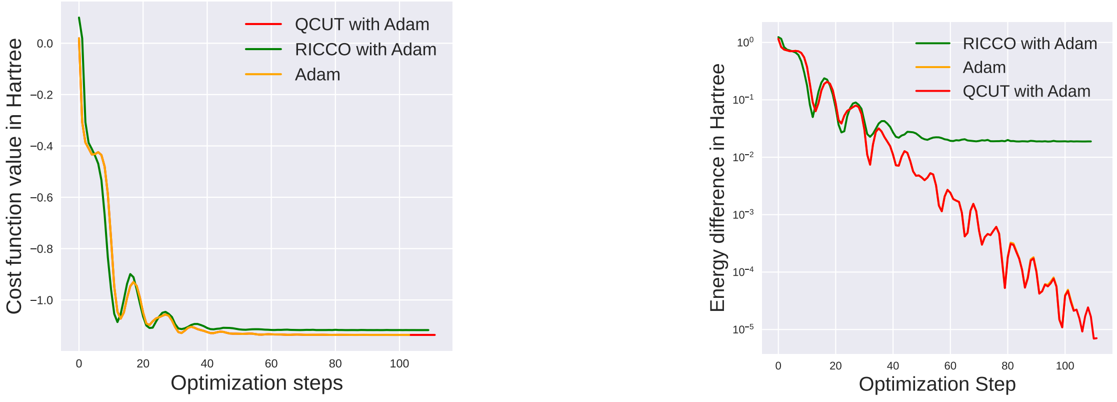
Algorithm 1: RICCO algorithm

```
Input: upstream and downstream subcircuit,
  observables in Hamiltonian and cost function
  equation 7
Output: $\langle Y\rangle$, Expectation value
$\epsilon \leftarrow 10^{-7}$ set the tolerance of RICCO;
$t \leftarrow 1$ set the initial tolerance;
Initialize $X$ (parameters of $U$);
Initialize $cost_{previous}$;
for observable in optimization observables do
    Initialize $params$;
    while $t > \epsilon$ do
        $params$, $cost_{new} \leftarrow$
          Optimize the cost function in equation 7;
        $t \leftarrow |cost_{previous} - cost_{new}|$ ;
    end
    $X \leftarrow params$;
end
$E \leftarrow$ measurement of expectation value of subcircuits
  with optimized $X$;
$\langle Y\rangle \leftarrow$ combining results of subcircuits' expectation
  values using tensor product of corresponding
  observables;
return $\langle Y\rangle$
```

Finally, we use these to construct our cost function. Let $|\bar{x}\rangle$, for $\bar{x} \in \{0, 1\}^n$ denote our n-qubit computational basis state. The function to minimize is

$$\begin{aligned} C_n(\rho') &= \left(1 - \prod_{\bar{x}\in\{0,1\}^n} F\left(|\bar{x}\rangle, \rho'\right)\right)^2 \\ &= \left(1 - \left(\frac{1}{2^n}\right)^{2^n}\prod_{\bar{x}\in\{0,1\}^n}\sum_{P_i\in P}\langle P_i\rangle_{\rho'}\langle P_i\rangle_{|\bar{x}\rangle}\right)^2 \end{aligned} \tag{10}$$

Whenever we rotate the state ρ and align it with one of the computational basis states, the fidelities with respect to all others will be minimized. Furthermore, based on our choice of basis state, we can reconstruct the original expectation value for the observable of interest with only one measurement setting in the computational basis.



(a) VQE cost function

(b) Energy difference curve

Fig. 3: Training curve for VQE with RICCO, QCUT and conventional VQE methods. 3a shows that RICCO's performance was similar to other methods. 3b shows the deviation of the ground state energy for the different methods compared with the actual ground state energy. RICCO converges to near the true energy, but with a larger energy difference with the true value than other methods.

III. Experimental Results

We evaluated RICCO by comparing the total number of quantum circuit executions after optimization against the method by Peng et al. [1], which we denote as QCUT. Because of the available computing resources, we simulated a total of 220 quantum circuits. The circuits were composed of two randomly-selected operations from unitary groups $U(16)$ and $U(32)$ for 1-qubit cut and 2-qubit cuts respectively. This is to ensure that we tested our method on the worst case possible. The total number of qubits for each circuit was 7 and 8 for 1-qubit cut and 2-qubit cuts respectively. Simulations and optimization (using Adam [7]) were performed using PennyLane [2], an open-source Python software framework for quantum differentiable programming.

Fig. 2a plots the percentage error of the expectation values $Z^{\otimes n} = Z^{\otimes(n_A+n_B+n_C)}$ of RICCO and QCUT circuit cutting methods compared with the ideal classical simulation of the uncut circuit for different number of cut qubits. This plot gives an insight into the accuracy of our method. We observed a reasonably high accuracy in RICCO with the worst case having less than 1.5% error. Fig. 2b plots the number of quantum circuit executions during optimization for the parameters of U in RICCO for 1-qubit and 2-qubits cut. On average, we require about 400 quantum circuit executions during optimization for the 15 parameters of U in the case of a 2-qubit cut. This number is very high partly because of the type of optimizer used. A better optimizer could reduce the number of executions required. We believe that more work needs to be done to explore how the optimization can be improved.

Fig. 2c plots the number of executions required to reconstruct the output of the uncut quantum circuit for RICCO and QCUT versus the number of cuts after optimization of the parameters of U. For circuits with 1 cut, the total number of quantum circuit executions for RICCO was 3 while QCUT was 7; and for 2 cuts, the total number of quantum circuit executions for RICCO was 5 while QCUT was 25. This result shows that, after optimization, RICCO improves runtime compared to QCUT.

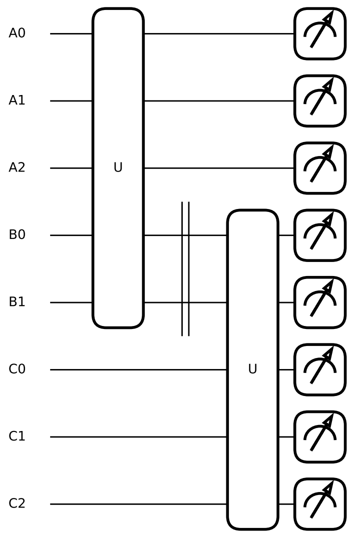


Fig. 4: Example of one of the quantum circuits formed by the combination of two unitaries from the $U(32)$ unitary group. The double line indicates the cut location on two qubits

IV. Circuit Cutting with VQE

After verifying correctness of the procedure and its behaviour in the worst case, it is of interest to evaluate its poten-

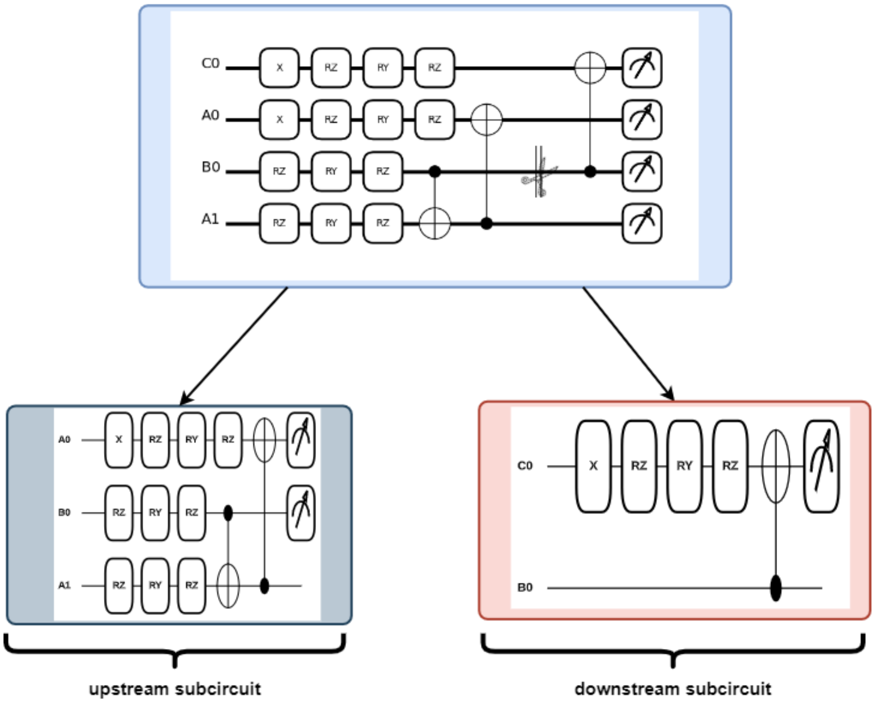


Fig. 5: VQE circuit ansatz for the simulation of hydrogen molecule. The double line with scissors on register $B0$ indicates the cut location. Two subcircuits emerge after the cut: the upstream and downstream subcircuits. The upstream subcircuit consists of qubit $A0$, the part of qubit $B0$ just before the cut and $A1$. The downstream subcircuit consists of qubit $C0$ and the part of qubit $B0$ just after the cut location

tial in practical applications. Variational quantum eigensolver (VQE) is one of the quantum algorithms in computational quantum chemistry used to predict the electronic structure and properties of molecules. In this section, we discuss the results of applying RICCO to perform numerical simulations of VQE to estimate the ground state energy of the hydrogen molecule.

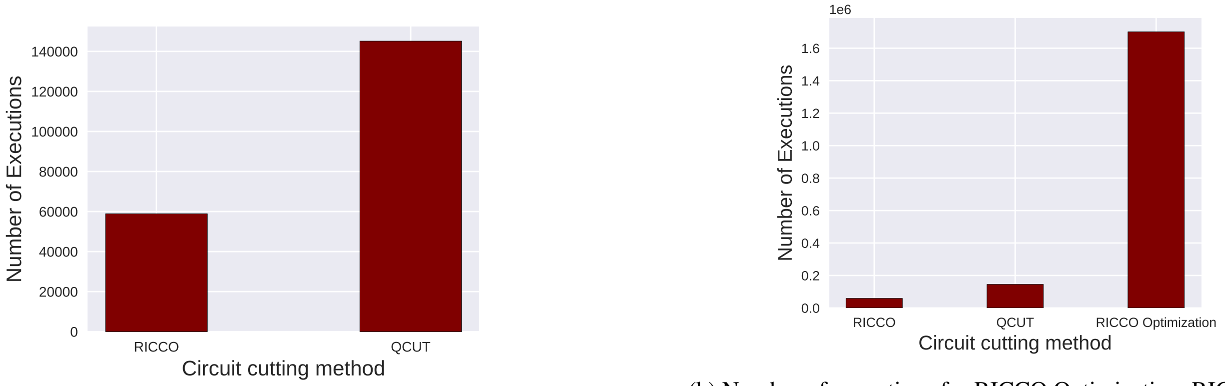
PennyLane was used to build a representation of the electronic Hamiltonian. We used the circuit from the PennyLane demo by Li et al. [9] as our ansatz (see Fig. 5). The parametrized circuit prepares the trial quantum state of H_2 molecule which is trained using a cost function that measures the expectation value of the problem Hamiltonian in the trial state to compute the ground state energy. The paramters of VQE were randomly initialized at the start of optimization with Adam's optimizer.

To apply RICCO to VQE, one of the qubits of the original ansatz was cut which resulted into two subcircuits before training. During the training phase of the VQE for the cut circuit, RICCO was used to reconstruct the expectation value for the cost function (see Algorithm 2).

The Hamiltonian consists of 18 observables which contribute to its expectation value. After cutting the circuit, it is expected that each of these observables are treated as separate entities and optimized separately with RICCO (see Algorithm 1). The results of the expectation value of each observable are then recombined to reconstruct the original expectation value of the uncut circuit. To reduce the number of RICCO optimiza-

Algorithm 2: VQE with RICCO algorithm

```
  Input: upstream and downstream subcircuit,
 RICCO subroutine, and $vqe_{params}$
  Output: $\langle Y \rangle$, Expectation value
$\epsilon \leftarrow 10^{-6}$ set the tolerance of VQE;
$max_{iter} \leftarrow 500$
 set the maximum number of iterations;
$iter \leftarrow 0$ initial iterations;
$t \leftarrow 0$;
Initialize X;
parameters of U Initialize $cost_{previous}$;
Initialize $params$;
while $iter < max_{iter}$ do
    $ricco_{params} \leftarrow$ RICCO optimization
    VQE Optimization step $\leftarrow$ricco$_{params}$;
    $vqe_{params}, vqe_{energy} \leftarrow$ VQE Optimization step
    $t \leftarrow |prev_{energy} - vqe_{energy}\ |$
    $cost \leftarrow vqe_{energy}$;
    if $t \leq \epsilon$ then
        break
    end
end
return $cost$
```



(a) Number of executions for RICCO and QCUT with VQE

(b) Number of executions for RICCO Optimization, RICCO and QCUT with VQE

Fig. 6: Optimization Results for VQE. 6a shows that RICCO requires less than half the number of quantum circuit executions compared to QCUT. 6b shows that RICCO's speedup was at the expense of its optimization

tion and hence, the number of executions for the 18 observables, we grouped them into 6 disjoint sets of observables, $\{\{III, IZI\}$, $\{IIZ, IZZ\}$, $\{XIY, XZY\}$, $\{YIX, YZX\}$, $\{ZII, ZZI\}$, $\{ZIZ, ZZZ\}\}$ for optimal RICCO optimization of the upstream subcircuit. This is an optimal choice because it captures all the observables within the Hamiltonian that belong to the uncut qubits of the upstream subcircuit and avoids repeating optimization for observables that are the same on the uncut qubits. It is important to note that for each observable in a set, only the middle observable changed – this is the cut qubit, $B0$. The first and last observables are the qubit registers $A0$ and $A1$ respectively.

At every step of VQE optimization, RICCO was deployed to reconstruct the original expectation value of the Hamiltonian. The set of 6 unique observables also consists of non-Z observables, $\{XIY, XZY\}$ and $\{YIX, YZX\}$ that were also optimized separately with RICCO at each step of the VQE optimization. It is important to do this to ensure the accurate reconstruction of the original expectation value. This process is repeated until VQE converges to a preset tolerance of about 10^{-6}. It took more than 70 iterations for RICCO to converge to this tolerance as shown in Fig. 3a.

Fig. 3a compares VQE cost function versus the number of iterations for RICCO, QCUT and the conventional VQE optimization methods. Adam optimizer was used for the different methods. Results from the optimization showed that although RICCO coverged, it did not converge to the actual value with the other two methods. This is evident in Fig. 3b which shows that the difference in RICCO's energy compared with the true value was worse (about 10^{-2} Hartree) than QCUT and conventional VQE with no cutting (about 10^{-6} Hartree). RICCO's convergence threshold was set to 10^{-7}. Increasing this threshold indicated some improvements in the energy difference at the expense of increased circuit executions and runtime. We noted that large difference in RICCO's energy could be as a result of the errors incurred during RICCO's optimization and the convergence threshold set for RICCO. Notwithstanding, exploring other causes of this error and how to mitigate it would be the focus of our future work.

V. Discussion

We want to understand how to reduce the number of measurements in cutting procedures, and see if this translates to improvements in actual applications such as VQE. Fig. 6a shows the number of executions required to implement VQE with RICCO and QCUT. RICCO requires less than half the number of quantum circuit executions compared to QCUT. Fig. 6b on the other hand, shows that number of executions required for RICCO optimization was at least $8X$ more than the number of executions for QCUT. This shows the trade-off between the improved runtime for RICCO and number of executions required for its optimization. We observed that this increase in the number of executions was because for each iteration of VQE, RICCO was optimized. One way to solve this problem could be to combine RICCO optimization with VQE optimization for each iteration of VQE instead of running a separate optimization for each. We can also reseed RICCO parameters at each optimization step.

Finally, we observed that the order of optimization of the 6 unique sets of observables could have contributed to the increased number of executions for RICCO. This is because, the optimization of any set of the observables depends on the outcome of the optimized parameters for the previous set of observable that was optimized by RICCO. Alternative distance measures such as trace distance can be explored as an area for future work to determine how they affect optimization convergence

References

[1] T. Peng, A. Harrow, M. Ozols, X. Wu, "Simulating large quantum circuits on a small quantum computer," Physical Review Letters, vol. 125, pp. 150504, 2020.

[2] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, et al., "Pennylane: Automatic differentiation of hybrid quantum-classical computations," arXiv:1811.04968, (2018).

[3] M. Perlin, Z. Saleem, M. Suchara, J. Osborn, C. James, “Quantum circuit cutting with maximum-likelihood tomography,” npj Quantum Information, vol. 7, pp. 1–8, 2021

[4] W. Tang, T. Tomesh, M. Suchara, J. Larson, M. Martonosi, “Cutqc: using small quantum computers for large quantum circuit evaluations,” Proceedings of the 26th ACM International conference on architectural support for programming languages and operating systems, pp. 473–486, 2021

[5] A. Lowe, M. Medvidović, A. Hayes, L. O’Riordan, T. Bromley, M. Arrazola, N. Killoran, “Fast quantum circuit cutting with randomized measurements,” arXiv:2207.14734, (2022)

[6] J. Romero, G. Björk, A. Klimov, L. Sánchez-Soto, “Structure of the sets of mutually unbiased bases for N qubits,” Physical Review A, vol. 72, pp. 062310, 2005

[7] P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980, (2014)

[8] Y. Cao, J. Romero, J. Olson, M. Degroote, P. Johnson, M. Kieferová, I. Kivlichan, T. Menke, B. Peropadre, N. Sawaya, etal. “Quantum chemistry in the age of quantum computing,” Chemical reviews, pp. 10856–10915, 2019

[9] M. Li, L. Bozanic, S. Sim, “Accelerating VQEs with quantum natural gradient,” https://bit.ly/3CzzwxP, (2021)

[10] A. Uhlmann “The “transition probability” in the state space of a $*$-algebra,” Reports on Mathematical Physics, pp. 273–279, 1976

[11] M. Nielsen, I. Chuang “Quantum computation and quantum information,” American Association of Physics Teachers, 2002

[12] S. Flammia, and Y. Liu, “Direct fidelity estimation from few Pauli measurements,” Physical review letters, pp.230501, 2011