Quantum circuit structure learning

Mateusz Ostaszewski,[1, 2, *] Edward Grant,[2, 3, †] and Marcello Benedetti[2, 4, ‡]

[1] *Institute of Theoretical and Applied Informatics,*
Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
[2] *Department of Computer Science, University College London, WC1E 6BT London, United Kingdom*
[3] *Rahko Limited, NN14 1UQ Northamptonshire, United Kingdom*
[4] *Cambridge Quantum Computing Limited, CB2 1UB Cambridge, United Kingdom*

(Dated: September 19, 2019)

We propose an efficient method for simultaneously learning both the structure and parameter values of quantum circuits with only a small computational overhead. Shallow circuits trained using structure learning perform significantly better than circuits trained using parameter updates alone, making this method particularly suitable for use on noisy intermediate-scale quantum computers. We demonstrate the method for training a variational quantum eigensolver for finding the ground states of Lithium Hydride and the Heisenberg model in simulation, and for finding the ground state of Hydrogen gas on the IBM Melbourne quantum computer.

INTRODUCTION

Quantum circuit learning is a method for tuning the parameters of a quantum circuit to perform a specific task. Multiple applications have been demonstrated including in chemical simulation, optimization, generative modeling and classification [1–8].

Typically, circuit learning is performed by selecting a fixed circuit structure, parametrizing it using rotation gates, then iteratively updating the parameters to minimize an objective function estimated from measurements. Circuits of this type are known as parametrized quantum circuits (PQCs). Methods based on quantum circuit learning are particularly promising for use with noisy intermediate-scale quantum (NISQ) computers because of their relative tolerance to noise compared to many other quantum algorithms.

In recent years a lot of progress has been made in improving the performance of PQCs including methods for calculating parameter gradients, hardware efficient ansätze, reducing the number of measurements required, and resolving problems with vanishing gradients [9–13]. Despite progress, many challenges remain. One of the most critical challenges is addressing the effects of noise. Generally, the effects of noise increase with the depth of the quantum circuit. For this reason it is highly desirable for a PQC to be as shallow as possible whilst being expressive enough to learn to perform the task at hand.

Few approaches have been proposed for learning the structure of a quantum circuit. In Ref. [14] the authors propose using a genetic algorithm for learning the circuit structure by selecting candidate gates from a set of allowed gates that are not parametrized. In Ref. [15] the authors propose growing the circuit by iteratively adding parametrized gates and re-training the PQC using gradient descent.

In this work we propose a method for efficiently learning the structure of quantum circuits while variationally optimizing an objective function, for example the energy given by a Hamiltonian operator. Gates are defined so that both the canonical direction and angle of rotation are degrees of freedom. Whilst the angle is parametrized in a continuous manner, the direction is chosen from a set of generators, namely tensor products of Pauli matrices. Training is performed by selecting the first gate and finding the direction and angle of rotation that yield minimum energy. This is performed iteratively for all the parametrized gates in the circuit. Once the last gate has been optimized the cycle is repeated until convergence.

In the Methods section we describe the approach in generality and then provide algorithms for the case of parametrized single-qubit gates and fixed two-qubit gates. This special case is interesting because it can be executed on all existing NISQ hardware and can provide a significant advantage as we show in the Results section. Note that for single-qubit rotations about fixed directions, the optimal angle can be found using 3 energy estimations. Independently from our work this property has been used as part of proposed methods for optimizing angles of rotation in quantum circuits [16, 17]. Our method extends this and finds also the optimal direction within a set of canonical ones. This comes with very little overhead, since only 7 energy estimations are required. We shall stress that although we demonstrate the method on parametrized single-qubit gates and fixed entangling gates, the method is not limited to this case. In the Discussion section we briefly present potential extensions and leave the details to the Appendix.

METHOD

Let us consider a variational problem where the objective function is encoded in a Hermitian operator M and the candidate solution is encoded in a parametrized quantum circuit $U = U_D \cdots U_1$ acting on an n-qubit initial state ρ. Each gate is either fixed, e.g., a controlled-Z, or parametrized of the form $U_d = \exp\left(-i \frac{\theta_d}{2} H_d\right)$. Here, $\theta_d \in (-\pi, \pi]$ are angles of rotation and H_d are Hermitian and unitary matrices, e.g., tensor products of Pauli matrices. We collect these parameters into a real vector $\boldsymbol{\theta}$ and a vector of matrices $\boldsymbol{H}$, respectively.

Without loss of generality we consider the task of minimizing the objective function which we simply refer to as *the energy*. Using subscripts to indicate arguments of functions, we can state the variational problem as $(\boldsymbol{\theta}^*, \boldsymbol{H}^*) = \arg\min_{\boldsymbol{\theta},\boldsymbol{H}} \langle M \rangle_{\boldsymbol{\theta},\boldsymbol{H}}$, where $\langle M \rangle_{\boldsymbol{\theta},\boldsymbol{H}} = \operatorname{tr}\left(MU\rho U^\dagger\right)$.

To solve the variational problem we present two methods. The first fixes the circuit structure and optimizes only the angles; the second optimizes structure and angles simultaneously. Both methods rely on the fact that the expectation value as a function of an angle of rotation has sinusoidal form. That is, if we fix $\boldsymbol{H}$ as well as all the degrees of freedom in vector $\boldsymbol{\theta}$ except one, say θ_d, we can express the expectation as $\langle M \rangle_{\theta_d} = A \sin(\theta + B) + C$ where A, B and C are unknown coefficients. A detailed derivation is provided in the Appendix. Clearly, if we were able to estimate these coefficients, we could also characterize the sinusoidal form. This can be exploited to design circuit learning strategies.

Our first method is a coordinate optimization [18–20] algorithm applied to the angles of rotation. It finds the optimal angle for one gate while fixing all others to their current values, and sequentially cycles through all gates. This is rather simple to perform since at each step the energy has sinusoidal form with period 2π. For gate U_d, the optimal angle has a closed form expression

$$\begin{aligned}\theta_d^* &= \arg\min_{\theta_d} \langle M \rangle_{\theta_d} \\ &= \phi - \tfrac{\pi}{2} - \operatorname{arctan2}\left(2\langle M \rangle_{\theta_d=\phi} - \langle M \rangle_{\theta_d=\phi+\frac{\pi}{2}} - \langle M \rangle_{\theta_d=\phi-\frac{\pi}{2}}\,,\ \langle M \rangle_{\theta_d=\phi+\frac{\pi}{2}} - \langle M \rangle_{\theta_d=\phi-\frac{\pi}{2}}\right) + 2\pi k,\end{aligned} \tag{1}$$

for any real ϕ and any integer k. In practice we select k such that $\theta_d^* \in (-\pi, \pi]$. A derivation of this expression is given in the Appendix.

The optimal angle can be found for all $d = 1, \ldots, D$ in order to complete a cycle. A new cycle is then initialized unless a stopping criterion is met. A number of potential stopping criteria could be used here. For example, one could stop after a fixed budget of K_1 cycles with the caveat that there may still be room for improvement. As another example, one could stop when the energy has not been significantly lowered for K_2 consecutive cycles. We call this algorithm `Rotosolve` which is summarized in Algorithm 1.

The choice of circuit structure is often based on prior knowledge about the problem as well as hardware constraints, e.g., qubit-to-qubit connectivity and gate set. It is reasonable to expect the chosen circuit structures to be suboptimal in the majority of cases. We believe there is large room for improvement here. Our second method relaxes the constraint of the fixed circuit structure and learns it along with the angles. Similarly to the first method, we opt for a greedy approach optimizing one gate at a time.

Recall that in the parametrization considered here, n-qubit gates are generated by Hermitian and unitary matrices such as tensor products of Pauli matrices $H_d \in \{I, X, Y, Z\}^{\otimes n}$. Structure learning aims at finding the optimal set of generators, which is clearly a daunting combinatorial problem. The general approach consists of using the expression in Eq. (1) to find minimizers $\theta_d^*(P) = \arg\min_{\theta_d} \langle M \rangle_{\theta_d=\phi, H_d=P}$ for all possible generators $P \in \{I, X, Y, Z\}^{\otimes n}$. Then, H_d is set to the generator giving lowest energy, and θ_d is set to the corresponding minimizer. This is repeated for $d = 1, \ldots, D$ to complete a cycle and the algorithm iterates until a stop criterion is triggered.

The above is very general and indeed suffers from combinatorial explosion due to the 4^n possible choices for the generator of each gate. However, in practice we shall still comply with the constraints of the underlying NISQ hardware. Since we are not considering compilation of logical gates to physical ones, only a very small subset of generators is available, namely the native gate set of the hardware. For simplicity, we now consider learning the structure of single-qubit gates while employing a fixed layout of two-qubit entangling gates similar to the one in Ref. [21]. Figure 1 illustrates the circuit layer and shows an example of how the algorithm updates gate's generator and angle.

When structure learning is limited to single-qubit gates, the generators can be selected such that $H_d \in \{X, Y, Z\}$. The identity generator is not required because it can be trivially obtained from any other generator by setting the angle to zero. In fact, we can exploit this to reduce the number of circuit evaluations per optimization step. According to Eq. (1), each of the three generators requires three circuit evaluations, for a total of 9 evaluations per optimization step. Recalling that ϕ can be chosen at will, we can use $\phi = 0$ to obtain the identity gate for all three generators. Since $\langle M \rangle_{\theta_d=0,H_d=X} = \langle M \rangle_{\theta_d=0,H_d=Y} = \langle M \rangle_{\theta_d=0,H_d=Z}$, we can estimate this quantity only once, effectively reducing the number of evaluations to 7 per optimization step. We call this algorithm `Rotoselect` and summarize it in Algorithm 2.

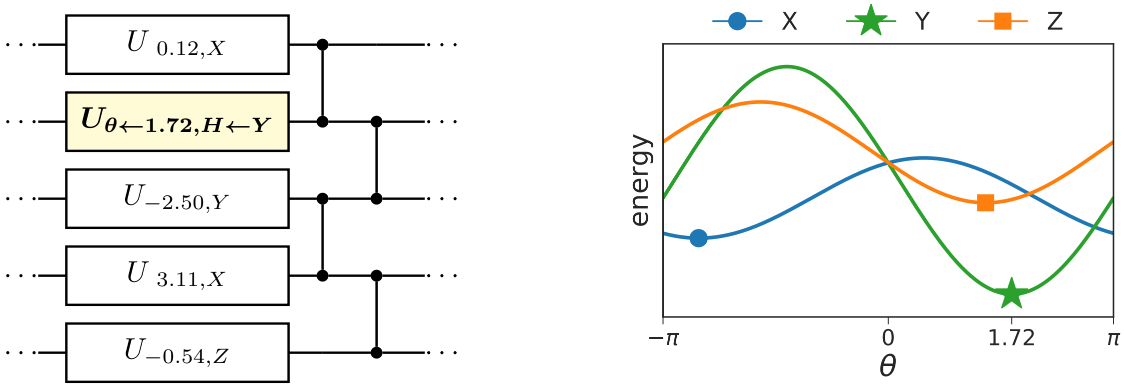

FIG. 1. The gradient-free algorithm `Rotoselect` sequentially adjusts angles of rotation and circuit structure in order to efficiently minimize the energy in a variational problem. For each generator the energy is a sinusoidal function of θ with period 2π. In this example, the single-qubit gate coloured in yellow has been assigned generator Y and angle of rotation 1.72. This is the configuration attaining minimal energy, as shown in the right panel.

Algorithm 1 `Rotosolve`

```
Input: Hermitian measurement operator $M$ encoding the objective function, parametrized quantum circuit $U$ with
    fixed structure, stopping criterion
Initialize $\theta_d \in (-\pi, \pi]$ for $d = 1, \dots, D$ heuristically or at random
repeat
    for $d = 1, \dots, D$ do
        Select a value $\phi \in \mathbb{R}$ heuristically or at random
        Estimate $\langle M \rangle_{\theta_d=\phi}$, $\langle M \rangle_{\theta_d=\phi+\frac{\pi}{2}}$ and $\langle M \rangle_{\theta_d=\phi-\frac{\pi}{2}}$ from samples
        $\theta_d \leftarrow \phi - \frac{\pi}{2} - \text{arctan2}\left(2\langle M \rangle_{\theta_d=\phi} - \langle M \rangle_{\theta_d=\phi+\frac{\pi}{2}} - \langle M \rangle_{\theta_d=\phi-\frac{\pi}{2}},\ \langle M \rangle_{\theta_d=\phi+\frac{\pi}{2}} - \langle M \rangle_{\theta_d=\phi-\frac{\pi}{2}}\right)$
until stopping criterion is met
```

Algorithm 2 `Rotoselect`

```
Input: Hermitian measurement operator $M$ encoding the objective function, parametrized quantum circuit $U$, stop-
    ping criterion
Initialize $\theta_d \in (-\pi, \pi]$ and $H_d \in \{X, Y, Z\}$ for $d = 1, \dots, D$ heuristically or at random
repeat
    for $d = 1, \dots, D$ do
        Estimate $\langle M \rangle_{\theta_d=\psi,H_d=P}$ for $\psi \in \{-\frac{\pi}{2}, +\frac{\pi}{2}\}$ and $P \in \{X, Y, Z\}$, and $\langle M \rangle_{\theta_d=0}$
        Compute minimizers $\theta_d^*(P)$ for $P \in \{X, Y, Z\}$ using Eq. (1)
        $H_d \leftarrow \arg\min_P \langle M \rangle_{\theta_d=\theta_d^*(P)}$
        $\theta_d \leftarrow \theta_d^*(H_d)$
until stopping criterion is met
```

Note that on a quantum computer the objective function is stochastic since $\langle M \rangle_{\boldsymbol{\theta},\boldsymbol{H}}$ is estimated from samples. For implementation in NISQ computers the operator is typically described as the weighted sum of polynomial terms $M = \sum_i w_i M_i$. In this case the expectation is given by $\langle M \rangle_{\boldsymbol{\theta},\boldsymbol{H}} = \sum_i w_i \langle M_i \rangle_{\boldsymbol{\theta},\boldsymbol{H}}$.

We conclude this section with a note about convergence. Assuming an infinite number of measurements, the objective $\langle M \rangle_{\theta_d}$ is a deterministic function of θ_d. Bertsekas [18] provides convergence results for coordinate minimization algorithms under rather general conditions and including non-convex objective functions. An analysis along these lines could be attempted for Rotosolve. In Rotoselect the analysis is complicated by the fact that multiple generators could lead to equally good minima. That is, at each given optimization step the solution may not be unique. Finally, in practice the objective function is stochastic and this further complicates the analysis.

RESULTS

The impact of structure learning on circuit performance is studied by comparing Rotoselect and Rotosolve, which are identical except that Rotoselect employs structure learning. For this reason any difference in performance can be attributed to structure learning. In the Appendix we compare both algorithms with other state-of-the-art optimizers and find significant improvements in terms of performance and scaling behavior.

First we compare the two algorithms for training a circuit ansatz to find the ground state energy of the 5-qubit Heisenberg model on a 1D lattice with periodic boundary conditions and in the presence of an external magnetic field. The corresponding Hamiltonian reads

$$M = J \sum_{(i,j)\in\mathcal{E}} (X_i X_j + Y_i Y_j + Z_i Z_j) + h \sum_{i\in\mathcal{V}} Z_i, \tag{2}$$

where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the undirected graph of the lattice with 5 nodes, J is the strength of the spin-spin interactions, and h is to the strength of the magnetic field in the Z-direction. For $J/h = 1$ the ground state is known to be highly entangled (see Ref. [10] for VQE simulations on the Heisenberg model). We chose $J = h = 1$.

Circuits were initialized using the strategy described in Ref. [13]. The circuit form consisted of layers of parametrized single-qubit rotations followed by a ladder of controlled-Z gates [21]. An example of this type of layer is shown in Fig. 1. Training was performed for 1000 iterations.

Figure 2 reports mean and variance of the lowest energy encountered during training across 10 trials for circuits with $6, 9, 12$ and 15 layers. Panel (a) shows results when 1000 measurements were used to estimate the expectation of each term in the sum in Eq. (2), while panel (b) shows oracular results in the limit of infinite measurements. For all number of layers, circuits trained using Rotoselect achieved better mean, variance, and absolute lowest energy than circuits trained using Rotosolve. In particular for the lower depth circuits Rotoselect achieves a much smaller variance than Rotosolve. This is likely due to the fact that the generators are initialized at random and Rotosolve is unable to change an initial bad condition. Given the limited capacity of low-depth circuits, a good choice for the generators appears to be particularly important. As the number of layers increases, both algorithms find better approximations to the ground state. Both algorithms exhibit robustness to the finite number of measurement employed here.

The experiment was repeated on the chemical Hamiltonian for Lithium Hydride (LiH) at bond distance, using the coefficients and the Pauli terms given in Ref. [10], Supplementary Material. Figure 3 shows results that are consistent with the previous experiment.

Finally, we compare the performance of Rotoselect on the 14-qubit IBM Melbourne quantum computer against simulations. For this test we chose to find the ground state energy of the 2-qubit Hydrogen Hamiltonian, again using the coefficients and the Pauli terms given in Ref. [10], Supplementary Material. The ansatz circuit had 2 layers and a total of 4 adjustable angles. In contrast to previous experiments where we used controlled-Z entangling gates, here we used CNOTs as they belong to the native gate alphabet of this device. We also characterized the measurement noise using an off-the-shelf method provided by Qiskit Ignis [22], namely the tensored measurement calibration with least squares fitting. In the following analysis, we do not include the overhead from the calibration since it was performed only once at the beginning of each trial.

We ran 5 trials with randomly initialized circuits and taken 1000 measurements for each Hamiltonian term when estimating the energy. Figure 4 shows the mean and one standard deviation of the energy as a function of the number of evaluations for two full cycles of Rotoselect. Despite error mitigation, results on the quantum device (blue dots) are in average slightly worse than those from the simulator (orange crosses). Yet we were able to get close to the ground state within the 56 evaluations required to complete two cycles.

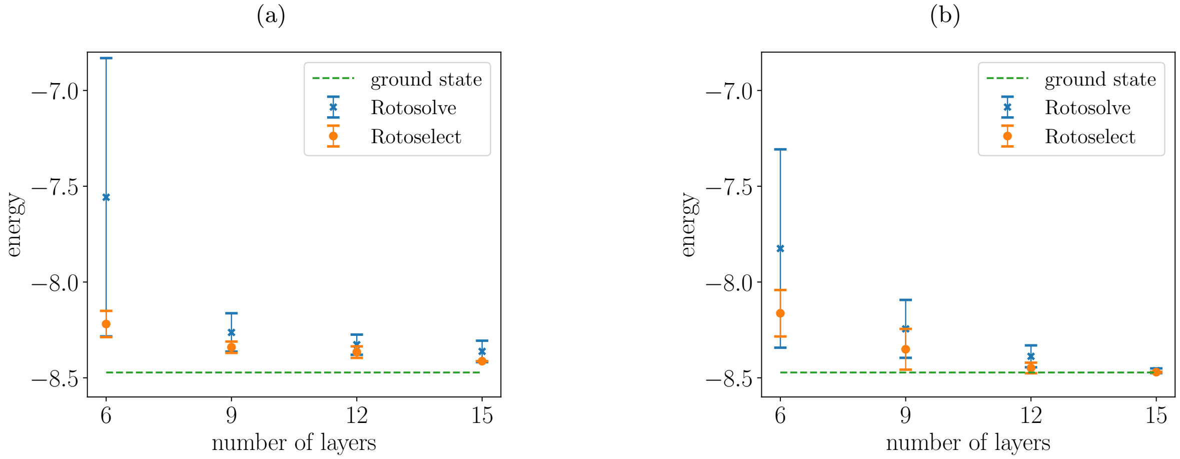

FIG. 2. Mean and standard deviation of energy across trials as a function of number of circuit layers comparing **Rotosolve** and **Rotoselect** for training a VQE to minimize the energy of the cyclic spin chain Heisenberg model on 5 qubits. In (a) 1000 measurements for each Hamiltonian term were used to approximate the energy. In (b) the exact energy was calculated.

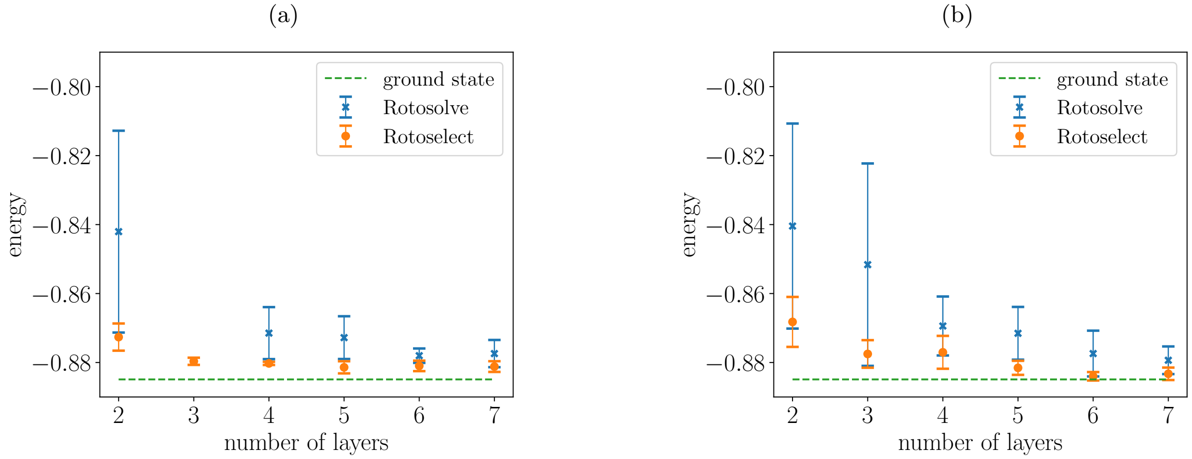

FIG. 3. Mean and standard deviation of energy across trials as a function of number of circuit layers comparing **Rotosolve** and **Rotoselect** for training a VQE to minimize the molecular Hamiltonian for LiH. In (a) 1000 measurements for each Hamiltonian term were used to approximate the energy. The result for **Rotosolve** for 3 layers lies above the plotted range. In (b) the exact energy was calculated.

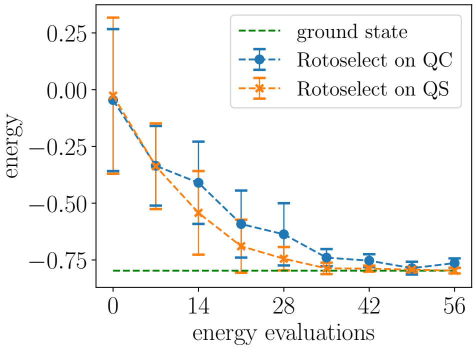

FIG. 4. Mean and standard deviation of the energy as a function of the number evaluations for the Hydrogen Hamiltonian. Blue dots corresponds to experiments made on the IBM Melbourne quantum computer (QC), and orange crosses corresponds to experiments made on quantum simulator (QS).

DISCUSSION

The proposed algorithms can be extended in a number of ways. In the context of circuit learning, Rotosolve can be generalized to find minimizers of K angles of rotation at the same time and at the cost of evaluating 3^K circuits. While this is an exponential cost, for small K this approach is computationally feasible and may provides an advantage. In Ref. [17] the authors explore this idea for $K \in \{1, 2, 3\}$ where subsets of angles are chosen at random. This approach belongs to the class of algorithms called coordinate block minimization [20].

In the context of circuit structure learning, Rotoselect can be generalized to incrementally grow circuits from scratch rather than starting from random initial structures. This is similar in spirit to Adapt-VQE [15], but with the advantage that each new gate is optimized efficiently in closed form rather than using gradient-based optimization. Furthermore, we could efficiently remove gates by assessing whether they are contributing to the solution (e.g., a redundant gate would generate sinusoidal forms that appear to be flat).

Another interesting extension is to use Rotoselect to learn the optimal connectivity layout for the entangling gates. As an example consider trapped ion computers that can implement fully connected layers of Mølmer-Sørensen gates [23]. This choice provides high expressive power in low-depth circuits [4], but must be balanced against the potentially slow clock speed of these entangling gates [24] and potential gate errors. Our algorithm could find this sweet spot. Since with n qubits we must evaluate $n(n-1)/2$ choices for non-directional two-qubit gates in a layer, this approach is also efficient.

Beyond quantum circuits, Rotoselect could be used to learn the structure of parametrized tensor networks in both classical and quantum models.

Finally, heuristics can be used to speedup our methods. In our simulations we found that Rotoselect tends to change the structure only in the early stage of training. We could detect this and switch from Rotoselect to Rotosolve hence reducing the number of circuit evaluations.

Quantum circuit structure learning provides an efficient means for improving the expressivity of low-depth quantum circuits with only a small computational overhead. This characteristic makes the described methods particularly suitable for deployment on near-term quantum computers where circuit depth and training time are significant bottlenecks.

ACKNOWLEDGEMENTS

M.B. is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and by Cambridge Quantum Computing Limited (CQC). E.G. is supported by ESPRC [EP/P510270/1]. M.O. acknowledges support from Polish National Science Center scholarship 2018/28/T/ST6/00429. We thank Leonard Wossnig for helpful discussions. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This research was supported in part by PLGrid Infrastructure.

CONTRIBUTIONS

M.B., E.G. and M.O. designed the Rotosolve and Rotoselect algorithms. M.B. performed theoretical justification. E.G. and M.O. wrote the code and performed experiments on the IBM Melbourne quantum computer. M.B., E.G. and M.O. analysed the numerical results.

Sinusoidal form of expectation values

Recall that we consider circuits $U = U_D \cdots U_1$ where each gate is either fixed, e.g., the controlled-Z, or parametrized. We consider parametrized gates of the kind $U_d = \exp\big(-i\frac{\theta_d}{2}H_d\big)$, where $\theta_d \in (-\pi, \pi]$ and the generator H_d is a Hermitian and unitary matrix such that $H_d^2 = I$. Using the definition of matrix exponential we obtain

$$\begin{aligned}
U_d &= \sum_{k=0}^{\infty} \frac{(-i)^k (\frac{\theta_d}{2})^k H_d^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{(-i)^{2k} (\frac{\theta_d}{2})^{2k} H_d^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{(-i)^{2k+1} (\frac{\theta_d}{2})^{2k+1} H_d^{2k+1}}{(2k+1)!} \\
&= \sum_{k=0}^{\infty} \frac{(-1)^k (\frac{\theta_d}{2})^{2k}}{(2k)!} I - i \sum_{k=0}^{\infty} \frac{(-1)^k (\frac{\theta_d}{2})^{2k+1}}{(2k+1)!} H_d \\
&= \cos\left(\tfrac{\theta_d}{2}\right) I - i \sin\left(\tfrac{\theta_d}{2}\right) H_d.
\end{aligned} \tag{3}$$

As an example of generators with this property, we could use tensor products of Pauli matrices $H_d \in \{I, X, Y, Z\}^{\otimes n}$, where n is the number of qubits.

We apply the circuit to a fiducial state $\bar{\rho}$ and then measure a Hermitian operator $\bar{M}$ encoding the objective function of the variational problem. From measurement outputs we can estimate the expectation

$$\langle \bar{M} \rangle = \mathrm{tr}\Big(\bar{M} U_D \cdots U_d \cdots U_1 \bar{\rho} U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger\Big). \tag{4}$$

We want to analyse this expectation as a function of a single parameter θ_d. To simplify the notation we absorb all gates before U_d in the density operator, that is $\rho = U_{d-1} \cdots U_1 \bar{\rho} U_1^\dagger \cdots U_{d-1}^\dagger$. Similarly, we absorb all gates after U_d in the measurement operator $M = U_{d+1}^\dagger \cdots U_D^\dagger \bar{M} U_D \cdots U_{d+1}$. This can be done because unitary transformations preserve the Hermiticity of both density and measurement operators. Using this notation Eq. (4) can be written as $\langle M \rangle = \mathrm{tr}\Big(M U_d \rho U_d^\dagger\Big)$.

In the following discussion we will need to evaluate the expectation at different parameter values for gate U_d. A further change in notation helps. From now on, we drop index d and use subscripts to indicate parameter value. Using the new notation we write

$$\begin{aligned}
\langle M \rangle_\theta &= \mathrm{tr}\left(M U_\theta \rho U_\theta^\dagger\right) \\
&= \mathrm{tr}\left(M \left(\cos\left(\tfrac{\theta}{2}\right) I - i \sin\left(\tfrac{\theta}{2}\right) H\right) \rho \left(\cos\left(\tfrac{\theta}{2}\right) I + i \sin\left(\tfrac{\theta}{2}\right) H\right)\right) \\
&= \cos^2\left(\tfrac{\theta}{2}\right) \mathrm{tr}\left(M\rho\right) + i \sin\left(\tfrac{\theta}{2}\right) \cos\left(\tfrac{\theta}{2}\right) \mathrm{tr}\left(M\left[\rho, H\right]\right) + \sin^2\left(\tfrac{\theta}{2}\right) \mathrm{tr}\left(MH\rho H\right).
\end{aligned} \tag{5}$$

Let us inspect the third line of this equation. First, we note that $\mathrm{tr}(M\rho)$ is simply the expectation when the circuit is evaluated at $\theta = 0$. That is, $\langle M \rangle_0 = \mathrm{tr}(M\rho)$. Second, the term $i\,\mathrm{tr}(M[\rho, H])$ can be written as the difference of two expectations. Indeed, using two independent circuits $U_{\pm\frac{\pi}{2}} = \exp\big(\mp i \frac{\pi}{4} H\big) = \frac{1}{\sqrt{2}}(I \mp iH)$ we have

$$\begin{aligned}
\langle M \rangle_{\frac{\pi}{2}} - \langle M \rangle_{-\frac{\pi}{2}} &= \mathrm{tr}\left(M U_{\frac{\pi}{2}} \rho U_{\frac{\pi}{2}}^\dagger\right) - \mathrm{tr}\left(M U_{-\frac{\pi}{2}} \rho U_{-\frac{\pi}{2}}^\dagger\right) \\
&= \tfrac{1}{2}\left(\mathrm{tr}\left(M\left(I - iH\right)\rho\left(I + iH\right)\right) - \mathrm{tr}\left(M\left(I + iH\right)\rho\left(I - iH\right)\right)\right) \\
&= -i\,\mathrm{tr}\left(MH\rho\right) + i\,\mathrm{tr}\left(M\rho H\right) \\
&= i\,\mathrm{tr}\left(M[\rho, H]\right).
\end{aligned} \tag{6}$$

Third, the last term $\mathrm{tr}\left(MH\rho H\right)$ is the expectation obtained evaluating the circuit at $\theta = \pi$. That is, using circuit $U_\pi = \exp\big(-i\frac{\pi}{2}H\big) = -iH$ we have $\langle M \rangle_\pi = \mathrm{tr}\left(MH\rho H\right)$.

Putting these three pieces back in Eq. (4) and using well known trigonometric identities, we obtain

$$\begin{aligned}
\langle M \rangle_\theta &= \cos^2\left(\tfrac{\theta}{2}\right) \langle M \rangle_0 + \sin\left(\tfrac{\theta}{2}\right) \cos\left(\tfrac{\theta}{2}\right) \left(\langle M \rangle_{\frac{\pi}{2}} - \langle M \rangle_{-\frac{\pi}{2}}\right) + \sin^2\left(\tfrac{\theta}{2}\right) \langle M \rangle_\pi \\
&= \tfrac{1+\cos(\theta)}{2} \langle M \rangle_0 + \tfrac{\sin(\theta)}{2} \left(\langle M \rangle_{\frac{\pi}{2}} - \langle M \rangle_{-\frac{\pi}{2}}\right) + \tfrac{1-\cos(\theta)}{2} \langle M \rangle_\pi \\
&= \tfrac{\cos(\theta)}{2} \left(\langle M \rangle_0 - \langle M \rangle_\pi\right) + \tfrac{\sin(\theta)}{2} \left(\langle M \rangle_{\frac{\pi}{2}} - \langle M \rangle_{-\frac{\pi}{2}}\right) + \tfrac{1}{2}\left(\langle M \rangle_0 + \langle M \rangle_\pi\right).
\end{aligned} \tag{7}$$

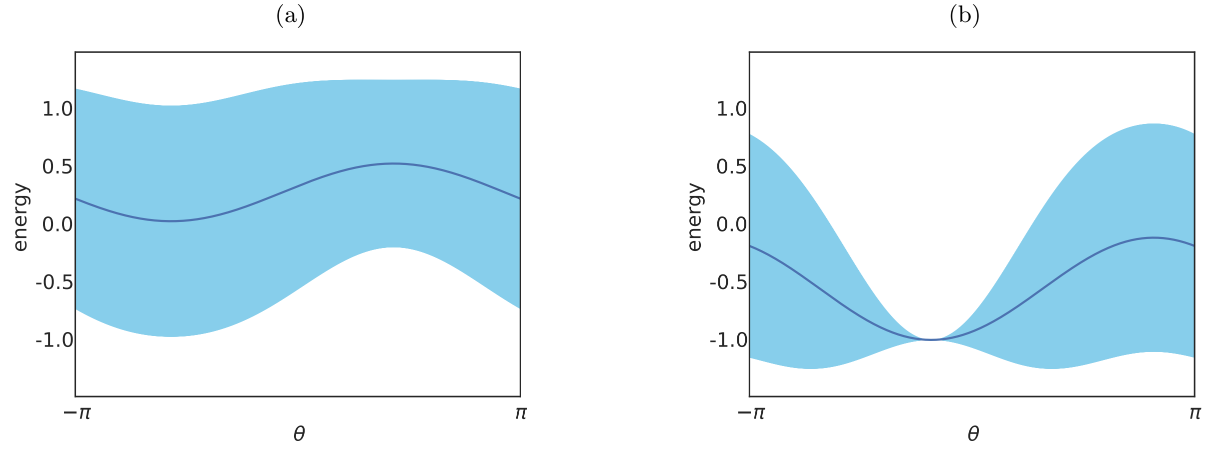

FIG. 5. Expectation and variance of Hermitian observable $ZXZX$ as a function of two different angles of rotation in a random circuit. The expectation value is always of sinusoidal form with period 2π. (a) The variance is large because this angle cannot yield an eigenstate of $ZXZX$. (b) Both expectation and variance are minimized by setting this second angle to $\theta^* \approx -0.5$.

We now use the identity $a\cos(x) + b\sin(x) = \sqrt{a^2+b^2}\sin\big(x + \arctan\big(\frac{a}{b}\big)\big)$ to obtain a compact expression. The expectation value then reads

$$\begin{aligned} \langle M\rangle_\theta &= A\sin(\theta+B)+C, \\ A &= \tfrac{1}{2}\sqrt{\left(\langle M\rangle_0 - \langle M\rangle_\pi\right)^2 + \left(\langle M\rangle_{\frac{\pi}{2}} - \langle M\rangle_{-\frac{\pi}{2}}\right)^2}, \\ B &= \operatorname{arctan2}\left(\langle M\rangle_0 - \langle M\rangle_\pi, \langle M\rangle_{\frac{\pi}{2}} - \langle M\rangle_{-\frac{\pi}{2}}\right), \\ C &= \tfrac{1}{2}\left(\langle M\rangle_0 + \langle M\rangle_\pi\right). \end{aligned} \tag{8}$$

That is, the expectation of M as a function of a single parameter has sinusoidal form with amplitude A, phase B, intercept C, and period 2π. An example is shown in Figure 5.

Note that we must use the arctan2 function in order to correctly handle the sign of numerator and denominator, as well as the case where the denominator is zero. To see why, assume H commutes either with M or ρ. Then, Eq. (6) evaluates to zero and triggers a division by zero in the standard arctan.

Now, from the graph of the sine function, it is easy to locate the minima at $\theta^* = -\frac{\pi}{2} - B + 2\pi k$ for all $k \in \mathbb{Z}$. The estimator for B in Eq. (8) seems to require four distinct circuit evaluations. Using simple trigonometry we generalize the estimator and show that only three evaluations are required. Let us write

$$\begin{aligned} \langle M\rangle_{\theta+\pi} &= A\sin(\theta+\pi+B)+C \\ &= -A\sin(\theta+B)+C \\ &= -\langle M\rangle_\theta + 2C. \end{aligned} \tag{9}$$

Solving for C we have

$$C = \tfrac{1}{2}\left(\langle M\rangle_{\theta+\pi} + \langle M\rangle_\theta\right). \tag{10}$$

Then, we can write

$$\tan(\phi+B) = \frac{\sin(\phi+B)}{\sin\left(\phi+B-\frac{\pi}{2}\right)} = \frac{\langle M\rangle_\phi - C}{\langle M\rangle_{\phi-\frac{\pi}{2}} - C} = \frac{2\langle M\rangle_\phi - \langle M\rangle_{\phi+\frac{\pi}{2}} - \langle M\rangle_{\phi-\frac{\pi}{2}}}{\langle M\rangle_{\phi+\frac{\pi}{2}} - \langle M\rangle_{\phi-\frac{\pi}{2}}}, \tag{11}$$

where we substituted Eq. (10) evaluated at $\theta = \phi - \frac{\pi}{2}$ for C. Taking the inverse tangent we obtain the general estimator for B requiring only three circuit evaluations

$$B = \operatorname{arctan2}\left(2\langle M\rangle_{\theta=\phi} - \langle M\rangle_{\theta=\phi+\frac{\pi}{2}} - \langle M\rangle_{\theta=\phi-\frac{\pi}{2}},\ \langle M\rangle_{\theta=\phi+\frac{\pi}{2}} - \langle M\rangle_{\theta=\phi-\frac{\pi}{2}}\right) - \phi. \tag{12}$$

Finally, we express the minimizer in closed form

$$\begin{aligned}\theta^* &= \arg\min_{\theta} \langle M\rangle_\theta \\ &= -\tfrac{\pi}{2} - B + 2\pi k \qquad\qquad (13)\\ &= \phi - \tfrac{\pi}{2} - \text{arctan2}\left(2\langle M\rangle_{\theta=\phi} - \langle M\rangle_{\theta=\phi+\frac{\pi}{2}} - \langle M\rangle_{\theta=\phi-\frac{\pi}{2}}\,,\ \langle M\rangle_{\theta=\phi+\frac{\pi}{2}} - \langle M\rangle_{\theta=\phi-\frac{\pi}{2}}\right) + 2\pi k,\end{aligned}$$

where $\phi \in \mathbb{R}$ and $k \in \mathbb{Z}$. Note that in practice we chose k such that $\theta^* \in (-\pi, \pi]$.

Comparison of optimization algorithms and scaling

In this experiment we compare the training time and training time scaling in the system size for `Rotosolve`, `Rotoselect`, `SPSA` [25] and gradient descent with `Adam` [26].

Figure 6 (a) shows the energy and standard deviation as a function of the number of energy evaluations across 5 trials for the 5-qubit Heisenberg cyclic spin chain described in the Results section. The depth of the circuit was 30 layers. The exact energy was used to perform updates. The learning rate for `Adam` was set to 0.05. The hyperparameter for `SPSA` were the same as in Ref. [10]. Both `Rotosolve` and `Rotoselect` converged with significantly fewer energy evaluations than `Adam` or `SPSA`.

Figure 6 (b) shows the mean and standard deviation for the number of evaluations needed to find the ground state of the Heisenberg cyclic spin chain as a function of the size of the system. The ground state was considered found if the candidate solution had energy within 2% from the actual ground state energy. It should be noted, that in the experiments there was an additional stop condition, which stopped algorithm after $100,000$ iterations. 5 trials were performed to estimate each mean and standard deviation. 1000 measurements were performed for each Hamiltonian term in order to approximate the energy for each evaluation. Circuit depth was $3n^2/2 + 2n$ for an even number of qubits and $3(n^2-1)/2 + 2n$ for odd where n was the number of qubits.

Both `Rotosolve` and `Rotoselect` converged faster than `Adam` or `SPSA` for the number of qubits tested, indicating a favorable scaling.

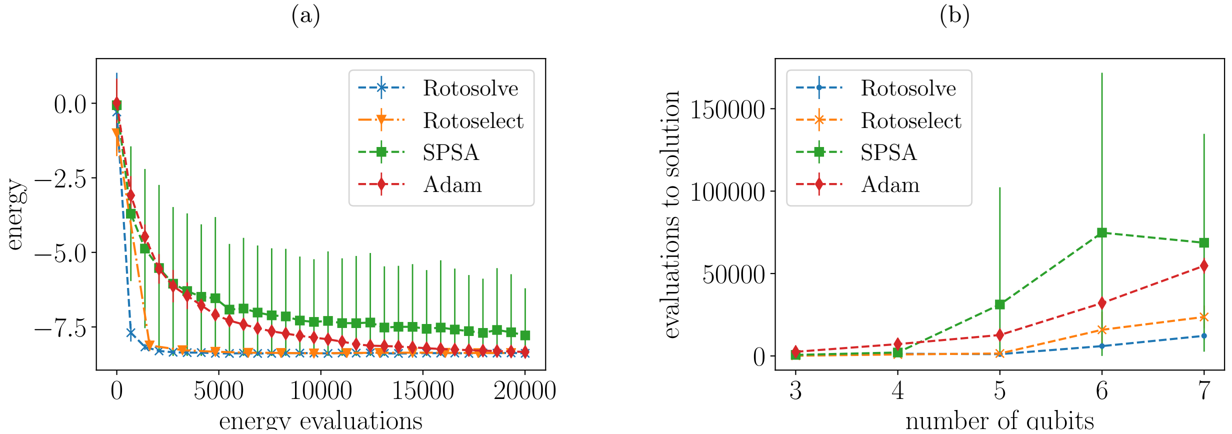

FIG. 6. A comparison of `Rotosolve`, `Rotoselect`, `SPSA` and `Adam`. (a) Energy as a function of number of energy evaluations for the 5-qubit Heisenberg cyclic spin chain. (b) Number of energy evaluations to solution as a function of the number of qubits for the Heisenberg cyclic spin chain.

A few generalizations

We now discuss some generators H_d for which `Rotosolve` applies. Our derivation above is rather general, but in practice Algorithms 1 and 2 rely on the canonical axes of rotation, i.e, tensor products of Pauli matrices.

Our first generalization applies to single-qubits gates. Recall that in order to obtain a sinusoidal form of expectation values we need $H_d^2 = I$. For single-qubit gates this condition is met by any $H_d = c_x X + c_y Y + c_z Z$ such that $(c_x, c_y, c_z) \in \mathbb{R}^3$ is a unit vector. To see why,

$$\begin{aligned} H_d^2 &= (c_x X + c_y Y + c_z Z)(c_x X + c_y Y + c_z Z) \\ &= c_x^2 X^2 + c_x c_y XY + c_x c_z XZ + c_y c_x YX + c_y^2 Y^2 + c_y c_z YZ + c_z c_x ZX + c_z c_y ZY + c_z^2 Z^2 \\ &= c_x^2 I + i c_x c_y Z - i c_x c_z Y - i c_y c_x Z + c_y^2 I + i c_y c_z X + i c_z c_x Y - i c_z c_y X + c_z^2 I \\ &= (c_x^2 + c_y^2 + c_z^2) I = I, \end{aligned} \tag{14}$$

where in the third line we used well-known identities for the Pauli matrices. Therefore, $\langle M \rangle_{\theta_d}$ has sinusoidal form for single-qubit gates of the kind $U_d = \exp\bigl(-i \frac{\theta_d}{2}(c_x X + c_y Y + c_z Z)\bigr)$.

This suggests a version of `Rotosolve` where a unit vector of coefficients is sampled at random, and where θ_d is found using Algorithm 1. This version has the potential of finding interesting axes of rotation. Note that a NISQ implementation may require compilation of U_d to lower-level gates.

Our second generalization applies to gates acting on an arbitrary number of qubits. Starting from tensor products of n Pauli matrices, i.e., $H_d \in \{I, X, Y, Z\}^{\otimes n}$, any unitary transformation V provides an alternative valid generator $H_d' = V H_d V^\dagger$. To see why,

$$(H_d')^2 = V H_d V^\dagger V H_d V^\dagger = V H_d^2 V^\dagger = V V^\dagger = I. \tag{15}$$

The result is an n-qubit gate of the form $U_d = \exp\bigl(-i \frac{\theta_d}{2} V H_d V^\dagger\bigr)$, where θ_d is again found using Algorithm 1. We leave the question of how to exploit gates of this kind for future work.

* mostaszewski@iitis.pl
† edward.grant@rahko.ai
‡ m.benedetti@cs.ucl.ac.uk

[1] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L Obrien, “A variational eigenvalue solver on a photonic quantum processor,” Nature communications **5**, 4213 (2014).

[2] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Hartmut Neven, “Quantum algorithms for fixed qubit architectures,” arXiv preprint arXiv:1703.06199 (2017).

[3] Edward Farhi and Hartmut Neven, “Classification with quantum neural networks on near term processors,” arXiv preprint arXiv:1802.06002 (2018).

[4] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz, “A generative modeling approach for benchmarking and training shallow quantum circuits,” npj Quantum Information **5**, 45 (2019).

[5] Marcello Benedetti, Edward Grant, Leonard Wossnig, and Simone Severini, “Adversarial quantum circuit learning for pure state approximation,” New Journal of Physics **21**, 043023 (2019).

[6] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni, “Universal discriminative quantum neural networks,” arXiv preprint arXiv:1805.08654 (2018).

[7] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G Green, and Simone Severini, “Hierarchical quantum classifiers,” npj Quantum Information **4**, 65 (2018).

[8] Marcello Benedetti, Erika Lloyd, and Stefan Sack, “Parameterized quantum circuits as machine learning models,” arXiv preprint arXiv:1906.07682 (2019).

[9] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii, “Quantum circuit learning,” Physical Review A **98**, 032309 (2018).

[10] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” Nature **549**, 242 (2017).

[11] Kosuke Mitarai, Tennin Yan, and Keisuke Fujii, “Generalization of the output of a variational quantum eigensolver by parameter interpolation with a low-depth ansatz,” Physical Review Applied **11**, 044087 (2019).

[12] Artur F Izmaylov, Tzu-Ching Yen, and Ilya G Ryabinkin, “Revising the measurement process in the variational quantum eigensolver: is it possible to reduce the number of separately measured operators?” Chemical Science (2019).

[13] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti, "An initialization strategy for addressing barren plateaus in parametrized quantum circuits," arXiv preprint arXiv:1903.05076 (2019).

[14] Rui Li, Unai Alvarez-Rodriguez, Lucas Lamata, and Enrique Solano, "Approximate quantum adders with genetic algorithms: an ibm quantum experience," Quantum Measurements and Quantum Metrology **4**, 1–7 (2017).

[15] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall, "Adapt-vqe: An exact variational algorithm for fermionic simulations on a quantum computer," arXiv preprint arXiv:1812.11173 (2018).

[16] Ken M Nakanishi, Keisuke Fujii, and Synge Todo, "Sequential minimal optimization for quantum-classical hybrid algorithms," arXiv preprint arXiv:1903.12166 (2019).

[17] Robert M Parrish, Joseph T Iosue, Asier Ozaeta, and Peter L McMahon, "A jacobi diagonalization and anderson acceleration algorithm for variational quantum algorithm parameter optimization," arXiv preprint arXiv:1904.03206 (2019).

[18] D.P. Bertsekas, *Nonlinear Programming* (Athena Scientific, 1999).

[19] Ankan Saha and Ambuj Tewari, "On the finite time convergence of cyclic coordinate descent methods," arXiv preprint arXiv:1005.2146 (2010).

[20] Stephen J Wright, "Coordinate descent algorithms," Mathematical Programming **151**, 3–34 (2015).

[21] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nature communications **9**, 4812 (2018).

[22] Héctor Abraham, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, Thomas Alexander, Eli Arbel, Abraham Asfaw, Carlos Azaustre, Panagiotis Barkoutsos, George Barron, Luciano Bello, *et al.*, "Qiskit: An open-source framework for quantum computing," (2019).

[23] Anders Sørensen and Klaus Mølmer, "Quantum computation with ions in thermal motion," Physical review letters **82**, 1971 (1999).

[24] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe, "Experimental comparison of two quantum computing architectures," Proceedings of the National Academy of Sciences **114**, 3305–3310 (2017).

[25] James C Spall *et al.*, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," IEEE transactions on automatic control **37**, 332–341 (1992).

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).