Classical variational simulation of the Quantum Approximate Optimization Algorithm

Matija Medvidović
Center for Computational Quantum Physics, Flatiron Institute, New York, NY 10010, USA
Department of Physics, Columbia University, New York, NY 10027, USA
matija.medvidovic@columbia.edu

Giuseppe Carleo
Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
giuseppe.carleo@epfl.ch

Abstract

We introduce a method to simulate parametrized quantum circuits, an architecture behind many practical algorithms on near-term hardware, focusing on the Quantum Approximate Optimization Algorithm (QAOA). A neural-network parametrization of the many-qubit wave function is used, reaching 54 qubits at 4 QAOA layers, approximately implementing 324 RZZ gates and 216 RX gates without requiring large-scale computational resources. our approach can be used to provide accurate QAOA simulations at previously unexplored parameter values and to benchmark the next generation of experiments in the Noisy Intermediate-Scale Quantum (NISQ) era.

1 Introduction

The past decade has seen a fast development of quantum technologies and the achievement of an unprecedented level of control in quantum hardware [1], clearing the way for demonstrations of quantum computing applications for practical uses. However, near-term applications face hardware limitations (dubbed Noisy Intermediate-Scale Quantum-NISQ computers [2]) where qubit count and lack of quantum error correction constrain potential applications. Despite these limitations, hybrid classical-quantum algorithms [3–6] have been identified as candidates for practical quantum advantage [7–10]. The Quantum Approximate Optimization Algorithm (QAOA) [5] is a notable example of variational quantum algorithm with prospects of quantum speedup on NISQ devices. Built to exploit quantum effects to solve combinatorial optimization problems, it has been extensively theoretically [11–16] and experimentally [17] studied on state-of-the-art hardware and proposed as a hardware benchmark [18–21]. In this work, we use a (neural network) variational parametrization [22] of the many-qubit state and extend the method of Ref. [23] to simulate QAOA. This approach trades the need for expensive exact classical simulation with an approximate and accurate description. We successfully simulate the Max-Cut QAOA circuit on a 3-regular 54-qubit graph [5, 17, 11] at depth $p = 4$.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

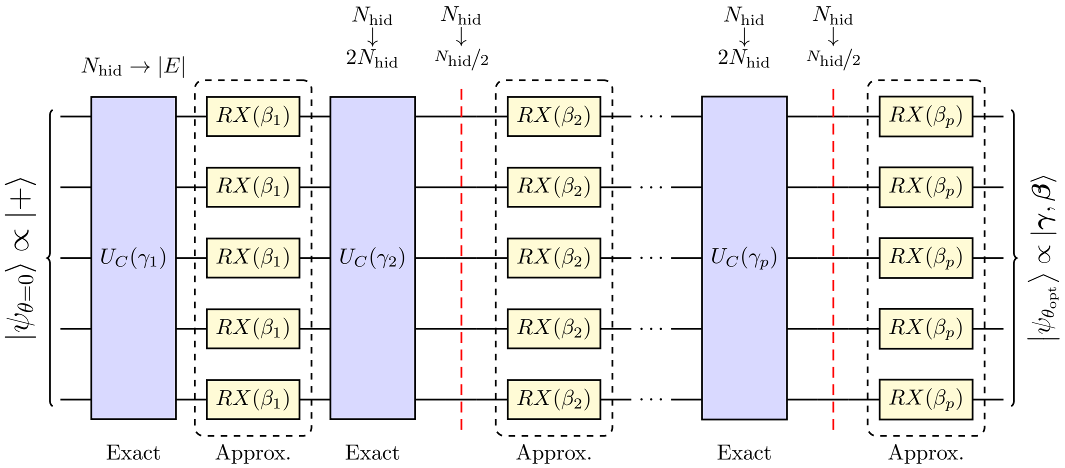
$N_{\text{hid}} \to |E|$
$N_{\text{hid}} \downarrow 2N_{\text{hid}}$
$N_{\text{hid}} \downarrow N_{\text{hid}}/2$
$N_{\text{hid}} \downarrow 2N_{\text{hid}}$
$N_{\text{hid}} \downarrow N_{\text{hid}}/2$
$|\psi_{\theta=0}\rangle \propto |+\rangle$
$U_C(\gamma_1)$
$RX(\beta_1)$
$U_C(\gamma_2)$
$RX(\beta_2)$
$\cdots$
$U_C(\gamma_p)$
$RX(\beta_p)$
$|\psi_{\theta_{\text{opt}}}\rangle \propto |\gamma, \beta\rangle$
Exact
Approx.
Exact
Approx.
Exact
Approx.

Figure 1: **The QAOA circuit.** A schematic representation of the QAOA circuit and our simulation. The input state is trivially initialized to $|+\rangle$. Next, at each p, the exchange of exactly (U_C) and approximately ($RX(\beta) = e^{-i\beta X}$) applicable gates is labeled (see Sec. 2). As noted in the main text, each (exact) application of the U_C gate increases the number of hidden units by $|E|$ (the number of edges in the graph). In order to keep that number constant, we "compress" the model (see Sec. 2), indicated by red dashed lines after each U_C gate. After the final layer, the RBM is parametrized by θ_{opt}, approximating the final QAOA output state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$.

2 Methods

2.1 The Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm for approximately solving discrete combinatorial optimization problems. Since its inception [5, 12], QAOA has been applied to Maximum Cut (Max-Cut) problems. With competing classical algorithms [24] offering exact performance bounds for all graphs, an open question remains – can QAOA offer quantum advantage by increasing the number of free parameters? In this work, we study a quadratic cost function [25, 26] associated with a Max-Cut problem. If we consider a graph $G = (V, E)$ with edges E and vertices V, the MaxCut of the graph G is defined by the following operator: $\mathcal{C} = \sum_{i,j \in E} w_{ij} Z_i Z_j$ where w_{ij} are the edge weights and Z_i are Pauli operators. The classical bitstring $\mathcal{B}$ that minimizes $\langle \mathcal{B} | \mathcal{C} | \mathcal{B} \rangle$ is the graph partition with the maximum cut. QAOA approximates such a quantum state through a quantum circuit of predefined depth p: $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U_B(\beta_p) U_C(\gamma_p) \cdots U_B(\beta_1) U_C(\gamma_1) |+\rangle$ where $|+\rangle$ is a symmetric superposition of all computational basis states: $|+\rangle = H^{\otimes N} |0\rangle^{\otimes N}$ for N qubits (see Fig. 1). The set of $2p$ real numbers γ_i and β_i for $i = 1 \dots p$ define the variational parameters to be optimized over by an external classical optimizer. The unitary gates defining the parametrized quantum circuit read $U_B(\beta) = \prod_{i \in V} e^{-i\beta X_i}$ and $U_C(\gamma) = e^{-i\gamma\mathcal{C}}$. Optimal variational parameters γ and β are then found by optimization of the following quantum expectation value: $C(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \mathcal{C} | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle$.

It is known that, for QAOA cost operators of the general form $\mathcal{C} = \sum_k \mathcal{C}_k(Z_1, \dots, Z_N)$, the optimal value asymptotically converges to the minimum value as $p \to \infty$. With modern simulations and implementations still being restricted to lower p values, it is unclear how large p has to get in practice before QAOA becomes comparable with its classical competition. In this work we consider 3-regular graphs with all weights w_{ij} set to unity at QAOA depths of $p = 1, 2, 4$.

2.2 Classical Variational Simulation

Consider a quantum system of N qubits. The Hilbert space is spanned by the computational basis $\{|\mathcal{B}\rangle : \mathcal{B} \in \{0,1\}^N\}$ of classical bit strings $\mathcal{B} = (B_1, \dots, B_N)$. A general state can be expanded in this basis as $|\psi\rangle = \sum_{\mathcal{B}} \psi(\mathcal{B}) |\mathcal{B}\rangle$. The convention $Z_i |\mathcal{B}\rangle = (-1)^{B_i} |\mathcal{B}\rangle$ is adopted. We use a neural-network representation of the many-qubit wavefunction $\psi(\mathcal{B})$. Specifically, we adopt a shallow network of the Restricted Boltzmann Machine (RBM) type: [27–29, 22]

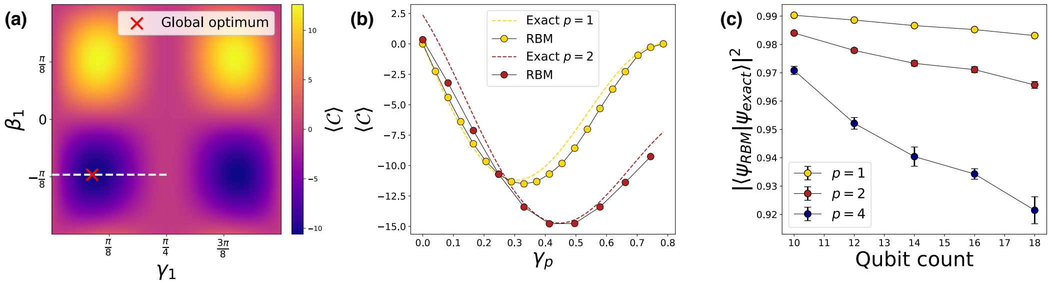

Figure 2: **Benchmarking the cost function for 20 qubits.** **a**: The exact variational QAOA landscape at $p = 1$ of a random 20-qubit instance of a 3-regular graph is presented, calculated using the analytical cost formula. The restricted cut along the constant-β line and at optimal γ is more closely studied in panel b. **b**: RBM-based output wavefunctions are contrasted with exact results. A variational landscape cut is presented at $p = 1, 2$. Optimal QAOA parameters are calculated using analytical derivatives ($p = 1$) or finite differences ($p = 2$) and a gradient-based optimizer. All parameters except the final γ_p are fixed at their optimal values. **c**: The fidelity between exact simulator outputs and RBM outputs for typical smaller graphs.

$$\psi(\mathcal{B}) \approx \psi_\theta(\mathcal{B}) \equiv \exp\left(\sum_{j=1}^{N} a_j B_j\right) \prod_{k=1}^{N_\mathrm{h}} \left[1 + \exp\left(b_k + \sum_{j=1}^{N} W_{jk} B_j\right)\right] . \tag{1}$$

It is parametrized by a set of complex parameters $\theta = \{\mathbf{a}, \mathbf{b}, W\}$ and not explicitly normalized. We note that the initial N-qubit $|+\rangle$ can be exactly implemented by setting all variational parameters to 0. That choice ensures that the wavefunction ansatz given in Eq. 1 is constant across all computational basis states, as required. The advantage of using the ansatz given in Eq. 1 as an N-qubit state is that a subset of one- and two-qubit gates can be exactly implemented as mappings between different sets of variational parameters $\theta \mapsto \theta'$. In general, such mapping corresponding to an abstract gate $\mathcal{G}$ is found as the solution of the following nonlinear equation: $\langle \mathcal{B} | \psi_{\theta'} \rangle = C \langle \mathcal{B} | \mathcal{G} | \psi_\theta \rangle$ (for all bitstrings $\mathcal{B}$ and any constant C, if a solution exists). For example, consider the Pauli Z gate acting on qubit i. In that case, we have $e^{a_i' B_i} = C(-1)^{B_i} e^{a_i B_i}$ after trivial simplification. The solution is $a_i' = a_i + i\pi$ for $C = 1$, with all other parameters remaining unchanged.

In addition, one can exactly implement a subset of two-qubit gates by introducing an additional hidden unit coupled only to the two qubits in question. Labeling the new unit by c, we can implement the RZZ gate relevant for QAOA. The gate is given as $RZZ(\phi) = e^{-i\phi Z_i Z_j} \propto \mathrm{diag}(1, e^{i\phi}, e^{i\phi}, 1)$ up to a global phase. Defining $\mathcal{A}(\phi) = \mathrm{Arccosh}\left(e^{i\phi}\right)$, the replacement rules read:

$$W_{ic} = -2\mathcal{A}(\phi) \ , \quad W_{jc} = 2\mathcal{A}(\phi) \ , \quad a_i \to a_i + \mathcal{A}(\phi) \ , \quad a_j \to a_j - \mathcal{A}(\phi) \ , \tag{2}$$

Not all gates can be treated exactly in this fashion. Most notably, gates that form superpositions belong in this category, including $U_B(\beta) = \prod_i e^{-i\beta X_i}$ required for running QAOA. This happens simply because a linear combination of two or more RBMs cannot be exactly represented by a single new RBM through a simple variational parameter change. To simulate those gates, we employ a variational stochastic optimization scheme. We take $\mathcal{D}(\phi, \psi) = 1 - F(\phi, \psi)$ as a measure of distance between two arbitrary quantum states $|\phi\rangle$ and $|\psi\rangle$, where $F(\phi, \psi)$ is the usual quantum fidelity:$F(\phi, \psi) = \frac{|\langle \phi | \psi \rangle|^2}{\langle \phi | \phi \rangle \langle \psi | \psi \rangle}$. In order to find variational parameters θ which approximate a target state $|\phi\rangle$ well ($|\psi_\theta\rangle \approx |\phi\rangle$, up to a normalization constant), we minimize $\mathcal{D}(\psi_\theta, \phi)$ using Stochastic Reconfiguration (SR) [30–32].

For larger p, extra hidden units introduced when applying $U_C(\gamma)$ at each layer can result in a large number of associated parameters to optimize over that are not strictly required for accurate output state approximations. So to keep the parameter count in check, we insert a model *compression* step which halves the number of hidden units immediately after applying U_C doubles it. Specifically we create an RBM with fewer hidden units and fit it to the output distribution of the larger RBM (output of U_C). Exact circuit placement of compression steps are shown on Fig. 1. As a result of the compression step, we are able to keep the number of hidden units in our RBM ansatz constant, explicitly controlling the variational parameter count.

2.3 Optimization

In this work we use the Stochastic Reconfiguration (SR) [30] algorithm to approximately apply quantum gates to the RBM ansatz. To that end, we write the "infidelity" between our RBM ansatz and the target state ϕ, $\mathcal{D}(\psi_\theta, \phi) = 1 - F(\psi_\theta, \phi)$, in a way that can easily be evaluated using samples $\mathcal{B} \sim |\psi_\theta|^2$ from Markov Chain Monte Carlo (MCMC) techniques: $F(\phi, \psi) = \left\langle \frac{\phi(\mathcal{B})}{\psi(\mathcal{B})} \right\rangle_\psi \left\langle \frac{\psi(\mathcal{B})}{\phi(\mathcal{B})} \right\rangle_\phi$. In addition, we can also express gradients of F using the same set of samples at each optimizer step:

$$\frac{\partial \mathcal{D}}{\partial \theta_l^*} = \left\langle \frac{\phi}{\psi_\theta} \right\rangle_{\psi_\theta} \left\langle \frac{\psi_\theta}{\phi} \right\rangle_\phi \left[\langle \mathcal{O}_k^* \rangle_{\psi_\theta} - \frac{\left\langle \frac{\phi}{\psi_\theta} \mathcal{O}_k^* \right\rangle_{\psi_\theta}}{\left\langle \frac{\phi}{\psi_\theta} \right\rangle_{\psi_\theta}} \right] . \tag{3}$$

Then, the optimizer update is performed using $\theta_k^{(t+1)} = \theta_k^{(t)} - \eta \sum_l S_{kl}^{-1} \frac{\partial \mathcal{D}}{\partial \theta_l^*}$ where S is the Quantum Geometric Tensor (QGT) [33] and η is the learning rate. The bulk of computational complexity at each optimization step comes from solving the linear system associated with the update. The matrix form of S can be ill-conditioned because it is only estimated from MCMC samples. We employ a standard regularization technique from Variational Monte Carlo by just adding a small constant ϵ to the diagonal of S. We find that using the QGT with the bare gradients significantly improves reached fidelities, enabling this RBM-based approach to consistently reach >90%.

3 Results

3.1 Simulation results for 20 qubits

QAOA angles $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$ are required as an input of our RBM-based simulator. At $p = 1$, we base our parameter choices on the position of global optimum that can be computed exactly [11]. For $p > 1$, we resort to direct numerical evaluation of the cost function as given in Sec. 2 from either the complete state vector of the system (number of qubits permitting) or from importance-sampling the output state as represented by a RBM. For all p, we find the optimal angles using Adam [34] with either exact gradients or their finite-difference approximations.

We begin by studying the performance of our approach on a 20-qubit system corresponding to the Max-Cut problem on a 3-regular graph of order $N = 20$. In that case, access to exact numerical wavefunctions is not yet severely restricted by the number of qubits. That makes it a suitable test-case. In Fig. 2, we present the cost function for several values of QAOA angles, as computed by the RBM-based simulator. Panel b shows cost functions of one typical random 3-regular graph instance. We observe that cost landscapes, optimal angles and algorithm performance do not change appreciably between different random graph instances. We can see that our approach reproduces variations in the cost landscape associated with different choices of QAOA angles at both $p = 1$ and $p = 2$. At $p = 1$, an exact formula [11] is available for comparison of cost function values. We report that, at optimal angles, the overall final fidelity (overlap squared) is consistently above 94% for all random graph instances we simulate.

3.2 Simulation results for 54 qubits

Our approach can be readily extended to system sizes that are not easily amenable to exact classical simulation. To show this, in Fig. 3 we show the case of $N = 54$ qubits. For the system of $N = 54$ qubits, we closely reproduce the exact error curve at $p = 1$. At $p = 4$, we exactly implement 324 RZZ gates and approximately implement 216 RX gates in total. This circuit size and depth is such that there is no available experimental or numerically exact result to compare against. The accuracy of our approach can nonetheless be quantified using intermediate variational fidelity estimates. These fidelities are exactly the cost functions (see Sec. 2) we optimize, separately for each qubit. In Fig. 3 b we show the optimal variational fidelities found when approximating the action of RX

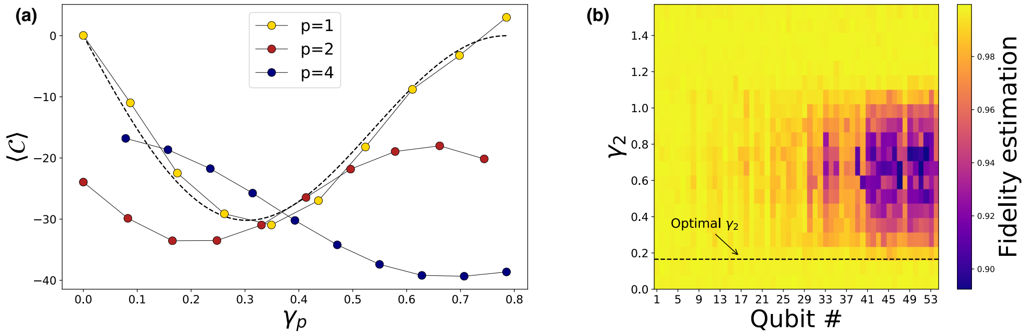

Figure 3: **Simulating 54 qubits.** **a**: Randomly generated 3-regular graphs with 54 nodes are considered at $p = 1, 2, 4$. At each p, all angles were fixed except for the final γ_p. The dashed line represents the exact cost at $p = 1$. Error bars were too small to be visible on the plot. **b**: An array of final stochastic estimations of single-qubit fidelities in the course of optimizer progress on 54 qubits at $p = 2$ where exact state vectors are intractable for direct comparison. Optimal γ_2 value (for given β_1, β_2, γ_1) is shown with a dashed line.

gates. At optimal γ_4 (minimum of $p = 4$ curve at Fig. 3 a), the lowest variational fidelity reached was above 98%, for a typical random graph instance shown at Fig. 3.

We remark that the stochastic optimization performance is sensitive to choices of QAOA angles away from optimum (see Fig. 3 b). The fidelity between the RBM state (Eq. 1) and the exact N-qubit QAOA output state decreases as one departs from the optimal $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. For larger values of QAOA angles, the associated optimization procedure is more difficult to perform, resulting in a lower fidelity (see the dark patch in Fig. 3 b). We find that optimal angles were always small enough not to be in the low-performance region. Therefore, this model is less accurate when studying QAOA states away from the variational optimum. However, even in regions with lowest fidelities, RBM-based QAOA states are able to approximate cost well, as can be seen in Fig. 2 and Fig. 3.

4 Conclusion

Using a novel approximate variational method, we successfully explore previously unreachable regions in the QAOA parameter space. The method is introduced as complementary to established numerical methods of classical simulation of quantum circuits because it is constrained by choices QAOA angles more than qubit counts or circuit depths. Classical variational simulations of quantum algorithms provide a natural way to both benchmark and understand the limitations of near-future quantum hardware. On the algorithmic side, our approach can help answer an open question in the field - whether QAOA can outperform classical optimization algorithms [35–37].

A Limitations

A discussed in the main text, our model is constrained by QAOA parameter choices (see Fig. 3 and accompanying discussion. An additional assumption is stated in Sec. 1 – we tested our approach only on 3-regular graphs, following earlier literature. Model performance may depend on graph structure in an unknown way.

B Computational resources

The code needed to reproduce results in this work is available by clicking on the following URL: github.com/QubitRBM. Numerical simulations were performed using NumPy [38] (BSD 3), SciPy [39] (BSD 3), Google Cirq [40] (Apache License 2.0). Random graph generation was done with NetworkX [41, 42] (BSD 3). Plots were generated using Matplotlib [43] (PSF).

The amount compute used is difficult to accurately report because the data wasn't collected during the process itself. In total, we estimate that the results presented in this paper used approximately

50 CPU-hours. These resources were used at a local cluster belonging to the authors' institution/s. No GPUs were used.

C Societal impact statement

This research develops a classical simulation technique for an algorithm designed for quantum computers. As such, it has very few and far-removed connections with immediate societal impacts which are difficult to quantify. Any impact is going to have to be related to the development of quantum hardware and all dangers it presents itself – especially in areas like cryptography in relation to Shor's algorithm.

As quantum hardware nears the limit of what is possible to simulate classically, it will become important to have reference results from large quantum systems. Our approach (or more general variational many-qubit wavefunctions) could help steer the way in the search for new algorithms. As an addition, it could help reduce the carbon footprint of all HPC resources needed to perform quantum simulations with more traditional methods.

D Paper checklist

D.1 For all authors...

- *Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?* **Yes**

  The authors believe that the main claims made in the abstract and introduction do accurately reflect the paper's contributions and scope.
- *Have you read the ethics review guidelines and ensured that your paper conforms to them?* **Yes**
- *Did you discuss any potential negative societal impacts of your work?* **Yes**

  Societal impact has been discussed in Sec. C.
- *Did you describe the limitations of your work?* **Yes**

  Limitations were described in Sec. A.

D.2 If you are including theoretical results... N/A

- *Did you state the full set of assumptions of all theoretical results?* **N/A**
- *Did you include complete proofs of all theoretical results?* **N/A**

D.3 If you ran experiments...

- *Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?* **Yes**

  The (anonymized) code code repository and has been linked in Sec. B.
- *Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?* **No**

  All of the training details could not be included as they relate to some of the finer details of Monte Carlo sampling and optimization not discussed in the main text for the sake of meeting the four page limit. However, algorithms used in this work are standard in the field and our hyperparameter choices do not differ from the literature nor does changing them affect the results appreciably. Relevant literature is cited where those choices are discussed.
- *Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?* **Yes**

  Error bars were reported where applicable.
- *Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?* **Yes**

  The amount compute was roughly estimated in Sec. B.

D.4 If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

- *If your work uses existing assets, did you cite the creators?* **Yes**
  Existing assets were used and cited in the main text in Appendix B.
- *Did you mention the license of the assets?* **Yes**
- *Did you include any new assets either in the supplemental material or as a URL?* **Yes**
  The (anonymized) code repository and has been linked in Sec. B.
- *Did you discuss whether and how consent was obtained from people whose data you're using/curating?* **N/A**
- *Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?* **N/A**

D.5 If you used crowdsourcing or conducted research with human subjects... N/A

- *Did you include the full text of instructions given to participants and screenshots, if applicable?* **N/A**
- *Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?* **N/A**
- *Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?* **N/A**

References

[1] Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).

[2] Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018). URL `https://quantum-journal.org/papers/q-2018-08-06-79/`. `1801.00862`.

[3] Peruzzo, A. *et al.* A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 1–7 (2014).

[4] Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors (2018). URL `http://arxiv.org/abs/1802.06002`. `1802.06002`.

[5] Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm (2014). URL `http://arxiv.org/abs/1411.4028`. `1411.4028`.

[6] Grant, E. *et al.* Hierarchical quantum classifiers. *npj Quantum Inf.* **4**, 1–8 (2018). `1804.03680`.

[7] Aspuru-Guzik, A., Dutoi, A. D., Love, P. J. & Head-Gordon, M. Chemistry: Simulated quantum computation of molecular energies. *Science* **309**, 1704–1707 (2005). URL `https://science.sciencemag.org/content/309/5741/1704`.

[8] O'Malley, P. J. *et al.* Scalable quantum simulation of molecular energies. *Phys. Rev. X* **6**, 031007 (2016). URL `https://journals.aps.org/prx/abstract/10.1103/PhysRevX.6.031007`. `1512.06860`.

[9] Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017). URL `https://www.nature.com/articles/nature23474`. `1611.09347`.

[10] Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).

[11] Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **97**, 022304 (2018).

[12] Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem (2014). URL `http://arxiv.org/abs/1412.6062`. `1412.6062`.

[13] Lloyd, S. Quantum approximate optimization is computationally universal (2018). URL http://arxiv.org/abs/1812.11075. 1812.11075.

[14] Jiang, Z., Rieffel, E. G. & Wang, Z. Near-optimal quantum circuit for Grover's unstructured search using a transverse field. *Phys. Rev. A* **95**, 062317 (2017). 1702.02577.

[15] Hadfield, S. *et al.* From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34 (2019). URL http://www.mdpi.com/1999-4893/12/2/34. 1709.03489.

[16] Zhou, L., Wang, S. T., Choi, S., Pichler, H. & Lukin, M. D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X* **10**, 21067 (2020). URL https://link.aps.org/doi/10.1103/PhysRevX.10.021067. 1812.01041.

[17] Harrigan, M. P. *et al.* Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**, 332–336 (2021). URL http://arxiv.org/abs/2004.04197. 2004.04197.

[18] Pagano, G. *et al.* Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci. U. S. A.* **117**, 25396–25401 (2020). URL http://arxiv.org/abs/1906.02700. 1906.02700.

[19] Bengtsson, A. *et al.* Improved success probability with greater circuit depth for the quantum approximate optimization algorithm. *Phys. Rev. Appl.* **14** (2020). URL http://arxiv.org/abs/1912.10495. 1912.10495.

[20] Willsch, M., Willsch, D., Jin, F., De Raedt, H. & Michielsen, K. Benchmarking the quantum approximate optimization algorithm. *Quantum Inf. Process.* **19**, 1–24 (2020). 1907.02359.

[21] Otterbach, J. S. *et al.* Unsupervised machine learning on a hybrid quantum computer (2017). URL http://arxiv.org/abs/1712.05771. 1712.05771.

[22] Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).

[23] Jónsson, B., Bauer, B. & Carleo, G. Neural-network states for the classical simulation of quantum computing. *arXiv* (2018). URL http://arxiv.org/abs/1808.05232. 1808.05232.

[24] Goemans, M. X. & Williamson, D. P. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* **42**, 1115–1145 (1995). URL https://dl.acm.org/doi/10.1145/227683.227684.

[25] Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 1–14 (2014). URL http://journal.frontiersin.org/article/10.3389/fphy.2014.00005/abstract. 1302.5843.

[26] Barahona, F. On the computational complexity of ising spin glass models. *J. Phys. A. Math. Gen.* **15**, 3241–3253 (1982). URL https://iopscience.iop.org/article/10.1088/0305-4470/15/10/028.

[27] Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **14**, 1771–1800 (2002).

[28] Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).

[29] Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

[30] Sorella, S. Green function monte carlo with stochastic reconfiguration. *Phys. Rev. Lett.* **80**, 4558–4561 (1998). 9803107.

[31] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953).

[32] Hastings, W. K. Monte carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).

[33] Stokes, J., Izaac, J., Killoran, N. & Carleo, G. Quantum Natural Gradient. *Quantum* **4**, 269 (2020). URL https://quantum-journal.org/papers/q-2020-05-25-269/. 1909.02108.

[34] Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. In *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* (International Conference on Learning Representations, ICLR, 2015). 1412.6980.

[35] Gomes, J., Eastman, P., McKiernan, K. A. & Pande, V. S. Classical quantum optimization with neural network quantum states (2019). URL http://arxiv.org/abs/1910.10675. 1910.10675.

[36] Zhao, T., Carleo, G., Stokes, J. & Veerapaneni, S. Natural evolution strategies and variational Monte Carlo. *Mach. Learn. Sci. Technol.* **2**, 2–3 (2020). URL https://doi.org/10.1088/2632-2153/abcb50.

[37] Hibat-Allah, M., Inack, E. M., Wiersema, R., Melko, R. G. & Carrasquilla, J. Variational Neural Annealing (2021). URL http://arxiv.org/abs/2101.10154. 2101.10154.

[38] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020). URL http://www.nature.com/articles/s41586-020-2649-2. 2006.10256.

[39] Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020). URL https://doi.org/10.1038/s41592-019-0686-2. 1907.10121.

[40] Gidney, C., Bacon, D. & The Cirq Developers. quantumlib/Cirq: A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits. (2018). URL https://github.com/quantumlib/Cirq.

[41] Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In *7th Python Sci. Conf. (SciPy 2008)*, 11–15 (2008). URL http://conference.scipy.org/proceedings/SciPy2008/paper_2/.

[42] Steger, A. & Wormald, N. C. Generating Random Regular Graphs Quickly. *Comb. Probab. Comput.* **8**, 377–396 (1999). URL https://www.cambridge.org/core/journals/combinatorics-probability-and-computing/article/abs/generating-random-regular-graphs-quickly/49CE556102BCF231EBC489CB2C2A7E65.

[43] Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **9**, 99–104 (2007).