JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

Implementation of Quantum Machine Learning for Electronic Structure Calculations of Periodic Systems on Quantum Computing Devices

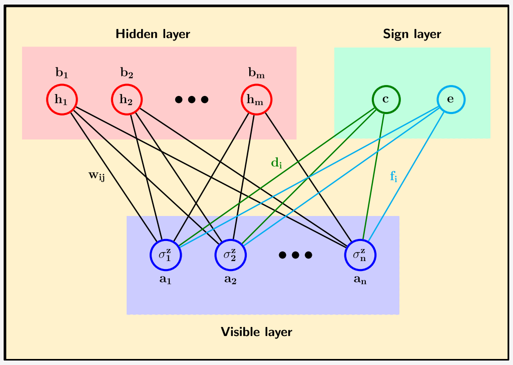
Shree Hari Sureshbabu, Manas Sajjan, Sangchul Oh, and Sabre Kais*

Cite This: *J. Chem. Inf. Model.* 2021, 61, 2667−2674 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

Downloaded via PURDUE UNIV on August 19, 2021 at 13:07:08 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

ABSTRACT: Quantum machine learning algorithms, the extensions of machine learning to quantum regimes, are believed to be more powerful as they leverage the power of quantum properties. Quantum machine learning methods have been employed to solve quantum many-body systems and have demonstrated accurate electronic structure calculations of lattice models, molecular systems, and recently periodic systems. A hybrid approach using restricted Boltzmann machines and a quantum algorithm to obtain the probability distribution that can be optimized classically is a promising method due to its efficiency and ease of implementation. Here, we implement the benchmark test of the hybrid quantum machine learning on the IBM-Q quantum computer to calculate the electronic structure of typical two-dimensional crystal structures: hexagonal-boron nitride and graphene. The band structures of these systems calculated using the hybrid quantum machine learning approach are in good agreement with those obtained by the conventional electronic structure calculations. This benchmark result implies that the hybrid quantum machine learning method, empowered by quantum computers, could provide a new way of calculating the electronic structures of quantum many-body systems.



1. INTRODUCTION

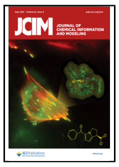
Machine learning (ML) driven by big data and computing power has made a profound impact on various fields, including science and engineering.[1] Remarkably successful applications of machine learning range from image and speech recognition[2,3] to autonomous driving.[4] The recent success of machine learning is mainly due to the rapid increase in classical computing power. This impact of ML has made it a useful tool to solve various problems in physical sciences.[5] Quantum computing is a new way of computation by harnessing the quantum properties such as the superposition and entanglement of quantum states. Some quantum algorithms run on quantum computers could solve the problems which are intractable by classical computers.[6] Recent progress in the development of Noisy Intermediate-Scale Quantum (NISQ) devices[7] makes it possible to run and test multiple quantum algorithms for various practical applications.

Quantum machine learning,[8] the interplay of classical machine learning techniques with quantum computation, provides new algorithms that may offer tantalizing prospects to improve machine learning. At the same time, these techniques aid in solving the quantum many-body problems.[9−14] Using neural networks with a supervised learning scheme, Xu et al.[15] have shown that measurement outcomes can be mapped to the quantum states for full quantum state tomography. Cong et al.[16] have developed a quantum machine learning model motivated by convolutional neural networks, which makes use of $O(\log(N))$ variational parameters for input sizes of N qubits that allows for efficient training and implementation on near term quantum devices.

It is important to solve many-body problems accurately for the advancement of material science and chemistry, as various material properties and chemical reactions are related to quantum many-body effects. Carleo and Troyer[17] introduced a novel idea of representing the many-body wave function in terms of artificial neural networks, specifically restricted Boltzmann machines (RBMs), to find the ground state of quantum many-body systems and to describe the time evolution of the quantum Ising and Heisenberg models. This representation was modified by Torlai et al.[18] for their purpose of quantum state tomography in order to account for the wave function's phase.

Quantum chemistry and electronic structure calculations using quantum computing are considered one of the first real

Received: March 16, 2021
Published: June 16, 2021



© 2021 American Chemical Society

applications of quantum computers.[19−23] Xia and Kais[24] proposed a quantum machine learning method based on RBM to obtain the electronic structure of molecules. The traditional RBM was extended to three layers to take into account the signs of the coefficients for the basis functions of the wave function. This method was applied to molecular and spin−lattice systems. Recently, Kanno et al.[25] have extended the method proposed by Xia and Kais by providing an additional unit to the third layer of an RBM in order to represent complex values of the wave functions of periodic systems.

Since the discovery of graphene, it has sparked a huge interest due to its remarkable properties. Recently, there has been a great deal of interest in studying graphene for quantum computing applications.[26,27] Hexagonal boron nitride (h-BN) gained attention when it was shown that graphene electronics is improved when h-BN is used as a substrate for graphene.[28] Of late, the interest to study h-BN for quantum information has grown since it was discovered that the negatively charged boron vacancy spin defects in h-BN display spin-dependent photon emission at room temperature.[29−31] Hence, in addition to studying graphene, it is important to study h-BN as it is a potential candidate for creating spin qubits that can be optically initialized and read out.

In this paper, we implement the quantum machine learning method with a three-layered RBM along with a quantum circuit to sample the Gibbs distribution[24,25] to calculate the electronic structure of periodic systems. Specifically, the implementation on NISQ devices is shown by modifying this quantum machine learning algorithm to run on an actual quantum computer. As the benchmark test, we demonstrate the performance of this algorithm first through the simulation of tight-binding and Hubbard Hamiltonians of monolayer hexagonal boron nitride and monolayer-graphene, respectively, on the IBM quantum computing processors, which is done using the IBM quantum experience. The valence band of the 2-D honeycomb lattices is calculated using quantum machine learning methods on IBM-Q and the *qasm* simulator. As we shall see, such valence band calculations on IBM-Q after employing a warm start and measurement error mitigation are shown to be in good agreement with the exact calculations.

This paper is organized as follows. In section 2, the quantum machine learning method based on RBM is introduced, and implementation details are discussed. Section 3 presents the results of electronic structure calculations using quantum machine learning on the *qasm* simulator and IBM-Q quantum computers. Finally, the summary and discussion will be given in section 4.

2. METHODOLOGY

In this section, we review the basic outline of the machine learning algorithm used and also discuss the implementation details.

Quantum Machine Learning Algorithm. A quantum many-body state $|\Psi\rangle$ can be expanded in terms of the basis $|\mathbf{x}\rangle$, $|\Psi\rangle = \sum \Psi(\mathbf{x})|\mathbf{x}\rangle$ where $\Psi(\mathbf{x})$ is the wave function. Carleo and Troyer's[17] method involved representing the trial wave function $\Psi(\mathbf{x};\, \theta)$ in terms of a neural network with parameters θ and to obtain the ground state by minimizing the expectation value of the Hamiltonian of a quantum many-body system, $E(\theta) = \langle\Psi(\theta)|H|\Psi(\theta)\rangle$. This was shown to use lesser number of parameters compared to tensor networks, indicating the efficiency of using such a representation. More specifically, the ansatz of a trial wave function is given by the marginal probability $P(\mathbf{x};\, \theta)$ of a visible layer of the RBM, $\Psi(\mathbf{x};\, \theta) = \sqrt{P(\mathbf{x};\, \theta)}$. While the learning of conventional RBMs is done by maximizing the likelihood function with respect to training data sets, the ground state of a neural network RBM state is obtained by minimizing the energy $E(\theta)$ using the stochastic optimization algorithm.

Xia and Kais[24] introduced the third layer with a single unit to take into account the signs of the wave function and apply the quantum restricted Boltzmann machine on actual quantum computers rather than the Monte Carlo method on classical digital computers. This quantum machine learning algorithm was further extended to take into account the complex value of the wave function.[25] However, implementation on an actual quantum computing processor was not shown, which would require multiple ancillary qubits as shown in this work.

The RBM we consider here consists of three layers: a visible layer, a hidden layer, and a sign layer, as shown in Figure 1. In contrast with the conventional RBMs with visible and hidden layers, the sign layer is added to take into account the real and imaginary values of the wave function of a quantum state.

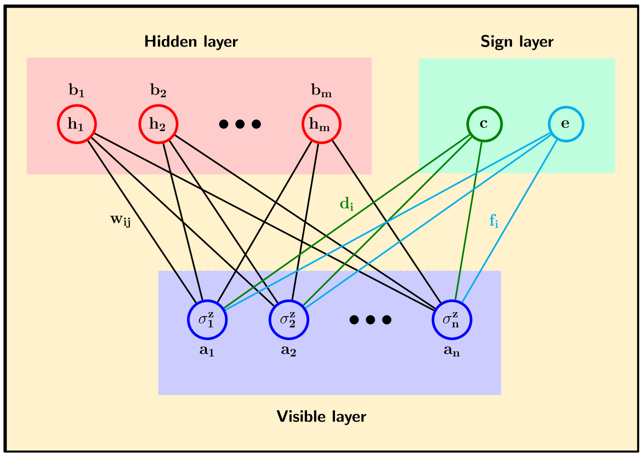


Figure 1. Restricted Boltzmann machine used to calculate the electronic structure of periodic materials. Here, the sign layer consists of two units, one to account for the real part and the other for the imaginary part of the wave function.

The wave function of a periodic system can be expressed as

$$|\Psi\rangle = \sum_{\mathbf{x}} \sqrt{P(\mathbf{x})}\, s(\mathbf{x})|\mathbf{x}\rangle \tag{1}$$

where

$$P(\mathbf{x}) = \frac{\sum_{\{h\}} e^{\sum_i a_i\sigma_i^z + \sum_j b_j h_j + \sum_{ij} w_{ij}\sigma_i^z h_j}}{\sum_{x'} \sum_{\{h\}} e^{\sum_i a_i\sigma_i^{z\prime} + \sum_j b_j h_j + \sum_{ij} w_{ij}\sigma_i^{z\prime} h_j}} \tag{2}$$

$$s(\mathbf{x}) = \tanh\left[\left(c + \sum_i d_i\sigma_i\right) + i\left(e + \sum_i f_i\sigma_i\right)\right] \tag{3}$$

Here σ_i^z is the z-component of the Pauli operators at i, $|\mathbf{x}\rangle = |\sigma_1^z\sigma_2^z\sigma_3^z \dots \sigma_n^z\rangle$ is the basis vector, and the values that σ_i^z and h_j take are $\{+1, -1\}$. a_i, b_j, c, and e denote the trainable bias parameters of the visible units, the hidden units, the unit representing the real part of the sign layer, and the unit

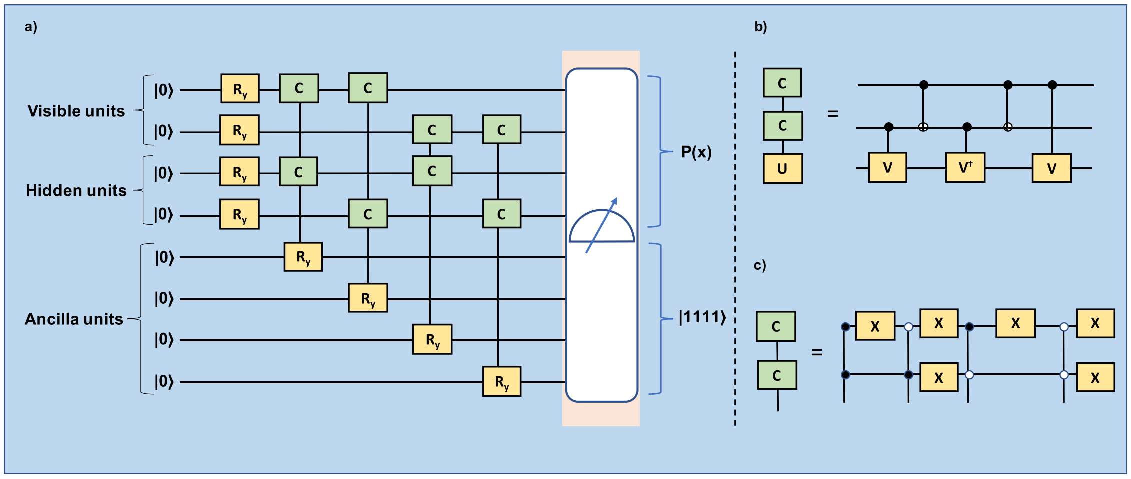


Figure 2. a) Quantum circuit to sample Gibbs distribution. This circuit consists of 2 visible units, 2 hidden units, and 4 ancilla qubits. R_y represents the single qubit rotation, $C - C - R_y$ represents the controlled−controlled rotation, with visible and hidden units being the control qubits and ancilla qubit being the target qubit. After measurement, if the ancilla qubits are in $|1111\rangle$, only then do the qubits corresponding to the visible and hidden units give the distribution $P(\mathbf{x})$. b) Decomposition of the $C - C - R_y$ gate for $|11\rangle$. Here $U = V^2$, and this leads to choosing $V = R_y(\theta/2)$. c) $C - C - R_y$ conditioned by $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ can be achieved by implementing the circuit in this form.

representing the imaginary part of the sign layer, respectively. w_{ij}, d_i, and f_i denote the trainable weights corresponding to the connections between σ_i^z and h_j, σ_i^z and the unit representing the real part of the sign layer, and σ_i^z and the unit representing the imaginary part of the sign layer, respectively. All the parameters are randomly initialized, and the values of these random numbers range from −0.02 to 0.02.

In order to obtain the probability distribution, the quantum circuit (shown in Figure 2) is employed. The quantum circuit consists of a single qubit rotation (R_y) and a controlled−controlled rotation operation ($C - C - R_y$). The angle by which the R_y operation rotates is determined by the visible and hidden bias parameters a_i and b_j. The angle by which the $C - C - R_y$ operation rotates is determined by the weights connecting the visible and hidden layers, w_{ij}. For each combination of visible and hidden units, $y = \{\sigma^z, h\}$, in order to increase the probability of successful sampling, the distribution $Q(y)$ is sampled rather than $P(y)$.[24] The two distribution functions $P(y)$ and $Q(y)$ are given by

$$P(y) = \frac{e^{\sum_i a_i\sigma_i^z + \sum_j b_j h_j + \sum_{ij} w_{ij}\sigma_i^z h_j}}{\sum_{y'} e^{\sum_i a_i\sigma_i^{z\prime} + \sum_j b_j h_j' + \sum_{ij} w_{ij}\sigma_i^{z\prime} h_j'}} \tag{4}$$

$$Q(y) = \frac{e^{1/k(\sum_i a_i\sigma_i^z + \sum_j b_j h_j + \sum_{ij} w_{ij}\sigma_i^z h_j)}}{\sum_{y'} e^{1/k(\sum_i a_i\sigma_i^{z\prime} + \sum_j b_j h_j' + \sum_{ij} w_{ij}\sigma_i^{z\prime} h_j')}} \tag{5}$$

Here, k is taken as $\max\left(\sum_{ij}\frac{|w_{ij}|}{2}, 1\right)$.[24] This is done in order to make the lower bound of the probability of successful sampling a constant. If k is taken to be 1, then the number of measurements required to get successful sampling becomes exponential. (See the Supporting Information.)

The target qubits for the controlled−controlled rotations are the ancilla qubits. Once all the rotations are completed, the ancilla qubits are measured. If the ancilla qubits are in $|1\rangle$, then the sampling is deemed successful. Then, the qubits corresponding to the visible and hidden units are measured to obtain the distribution $Q(y)$. Once the distribution $Q(y)$ is obtained, the probabilities are calculated to the power of k and then normalized to get $P(y)$. With $P(y)$ computed through our QML algorithm and $s(y)$ computed classically, the wave function $|\psi\rangle$ is computed, and through this the energy $E(\theta)$ is obtained. This value of $E(\theta)$ is optimized through gradient descent until the eigenvalue of the Hamiltonian is obtained.

For this algorithm, the number of qubits required scales as $O(nm)$, and the complexity of the gates turns out to be $O(nm)$ for one sampling,[24] where n is the number of visible units and m is the number of hidden units.

Implementation Methods. The developed quantum machine learning algorithm for calculating the band structures of monolayer h-BN and monolayer-graphene is executed using the following tools:

(i) We start with the implementation of the algorithm classically. Classical simulation is performed to ensure the algorithm performs accurately. Here, classical simulation implies that the gates were simulated on a classical computer.

(ii) Having ensured that the algorithm works when implemented classically, we move on to implementing it using Qiskit. Qiskit stands for IBM's Quantum Information Software Kit (Qiskit) and is designed to mimic calculations performed on a real noisy-intermediate scale quantum computing device using a classical computer. Specifically, we implemented the algorithm on the *qasm* back-end, which is a high-performance quantum circuit simulator amenable to treat the errors (noise) associated with the implementation of the quantum circuit with appropriate customizable noise models. Essentially, the *qasm* simulator is designed to replicate an actual noisy quantum device. Even if a custom noise is not chosen, the *qasm* simulator can be made to mimic a particular IBM-Q backend, in which case, the simulator automatically assumes a noise consistent with the hardware of the real device. The $C - C - R_y$ gate can be implemented by using Qiskit's multicontrolled y-rotation (mcry) operation, by specifying the control, target, and ancillary qubits. The circuit is executed multiple times on the simulator, each time culminating in the chosen set of measurements. The

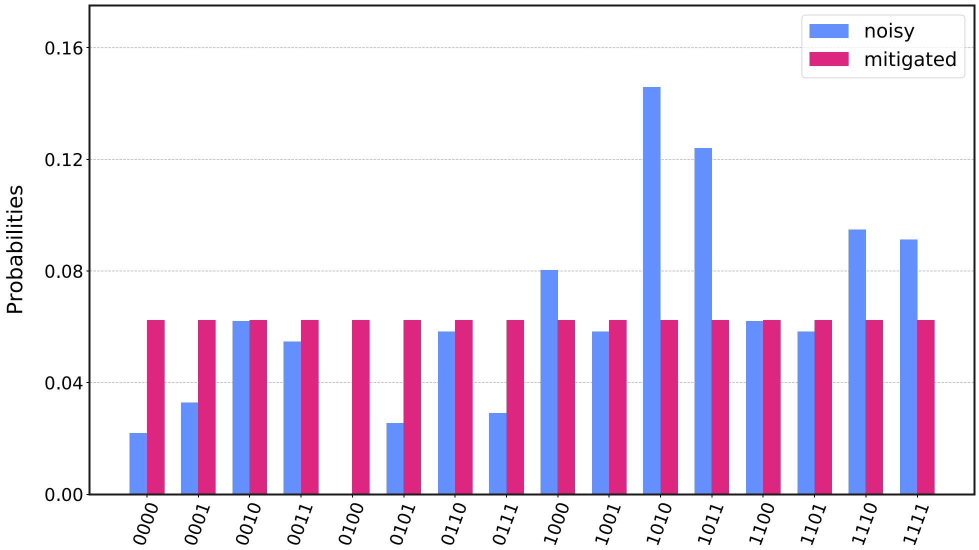


Figure 3. Probabilities of states with ancilla qubits being in |1111⟩ for both the cases with and without measurement error mitigation for the first iteration.

return values are the probabilities for observing the system in measurement basis states with statistical errors due to finite sampling.

(iii) We conclude our discussion by implementing and demonstrating the validity of our results using two actual IBM-Q quantum computers available. Qiskit's results in (i) are compared with those obtained from these real quantum devices.

In the following section, we display the simulation results. The terms "RBM Value" and "Exact Value" stand for the values of valence band energies obtained from our RBM-based approach and from exact diagonalization of the Hamiltonian, respectively.

Initializing the parameters of the RBM randomly can lead to the energies corresponding to certain k points being stuck at local minima. To enhance the generalizing capability of a machine learning model, transfer learning technique can be used. Recently, it has also been extended to the realm of quantum computing.[32] However, in our case, in order to improve the convergence, a method of warm starting is sufficient, wherein the parameters of a previously converged point are used to initialize the parameters of the current point of calculation. Noting that the band structure exists in a 4D space corresponding to energy as a function of k_x, k_y, and k_z, in this case too, if the optimization is performed such that the energy is minimized for every (k_x, k_y, k_z) point, then the parameters of such a point in 4D space can be considered to improve the convergence of the other points.

When implementing the algorithm on NISQ devices, we have to account for the noise that interferes with the accuracy of the results. In this work, we try to mitigate the errors that occur during measurement using *measurement error mitigation*.

The counts corresponding to each state will not be definite as a result of noise. There will be a finite number of counts corresponding to the other basis states even when the measurement outcome is supposed to result in one. So, the counts for each state can be written as a column vector, and a matrix, called the calibration matrix, can be defined corresponding to the concatenation of all column vectors describing the counts for all the basis states. The least-squares method can now be used to get the error mitigated probabilities for each of the states by using the calibration matrix, the ideal state vector, and the noisy result that was obtained. An example of the probability distribution $Q(y)$ obtained with and without measurement error mitigation is shown in Figure 3.

3. RESULTS AND DISCUSSION

As a benchmark test of our quantum machine learning algorithm on existing IBM quantum computers, we calculate the electronic structures of two well-studied two-dimensional periodic systems with hexagonal lattices namely boron-nitride and graphene. In this section, we discuss the results for each of the two systems.

Band Structure of Monolayer h-BN. Hexagonal boron nitride (h-BN) has a unit cell containing one B atom and another N atom. For h-BN, the levels involving the other valence orbitals, the $2s$, $2p_x$, and $2p_y$, are either quite far above or far below the Fermi level. The conduction and valence bands, which are around the Fermi level, are formed from the $2p_z$ orbital, and hence, a tight-binding Hamiltonian using the frontier $2p_z$ orbital and with third-nearest neighbor interaction on each of the two atoms of the unit cell is employed to obtain the electronic structure. Such a treatment affords the requisite dimensionality reduction as the number of qubits available on the IBM quantum computers is limited. Considering spin-degeneracy, the tight-binding Hamiltonian of the h-BN is thus given by a 4 × 4 Hermitian matrix (see the Supporting Information). The number of visible units needed for the simulation is 2, and the number of hidden units is taken to be equal to the number of visible units. For quantum optimization, 2 qubits are used to represent the visible nodes, and 2 qubits are used to represent the hidden nodes. In addition, 4 ancillary qubits are required (see Figure 2). In total, the number of qubits required is equal to 8. The sampling of Gibbs distribution is performed by applying the following sequences of quantum gates: 4 single-qubit rotation gates (R_y), 16 controlled−controlled rotation gates ($C - C - R_y$), and 24 bit-flip (X) gates, as illustrated in Figure 2.

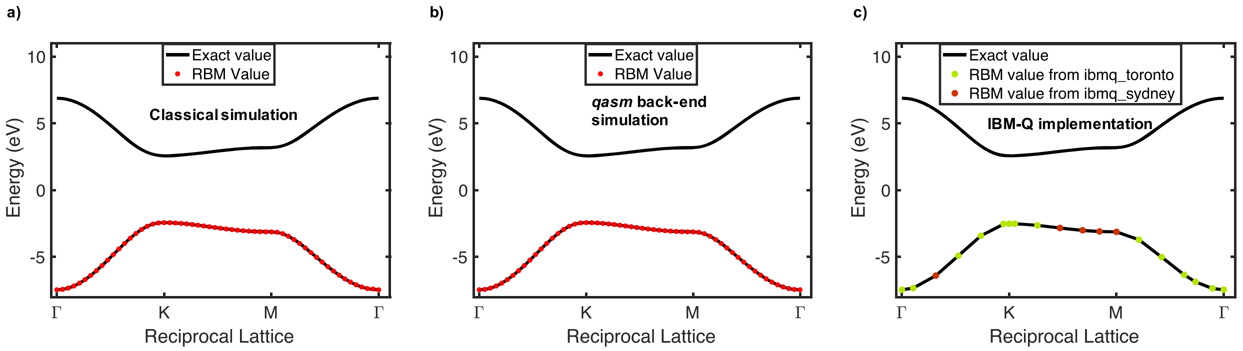


Figure 4. Band structure of h-BN calculated using (a) classical simulation with a warm start (red). The solid black curves show the valence and conduction bands from exact diagonalization. (b) The *qasm* back-end simulation with the aid of a warm start (red). (c) The implementation of the RBM sampling circuit on *ibmq_toronto* (green) and *ibmq_sydney* (red).

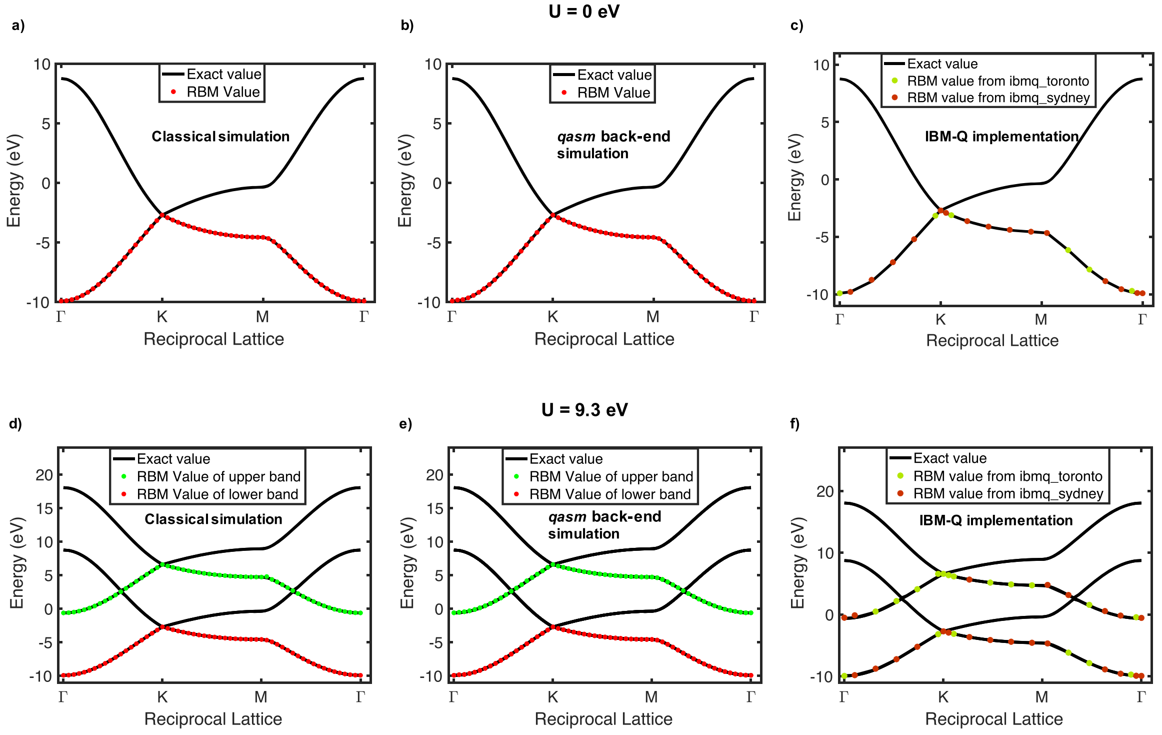


Figure 5. Band structure of the graphene for $U = 0$ eV calculated using (a) classical simulation with a warm start (red). The solid black curves show the valence and conduction bands from exact diagonalization, (b) the *qasm* back-end simulation with the aid of a warm start (red). (c) Implementation on actual IBM quantum computing devices. (d) Same as in (a) with Hubbard on-site interaction $U = 9.3$ eV. The four bands correspond to the two nondegenerate spin states for each of the valences and conduction bands in plot (a). (e) Same as in (b) with Hubbard on-site interaction $U = 9.3$ eV. (f) Same as in (c) with Hubbard on-site interaction $U = 9.3$ eV.

For h-BN band structure calculations, we start with the results from RBM by implementing the gate-set (see Figure 2) classically and then on the Qiskit's quantum simulator, called the *qasm* back-end. Figure 4(a) shows the band structure of h-BN as a function of wave-vector amplitude sampled from the first Brillouin zone. We overlay the valence band energies obtained from our RBM network on a classical computer with the exact diagonalization of the 4 × 4 tight-binding Hamiltonian (black curve). The two results are in excellent agreement. It must be noted that without a warm start results may show deviations from the exact value at certain k-points, as the optimization protocol may get locally trapped. However, the use of the warm starting technique eliminates such convergence issues. Figure 4(b) shows the band structure calculation of h-BN wherein for the RBM, the quantum gates are implemented on the Qiskit's *qasm* back-end. For the sake of our simulations, no noise model was considered, and the results obtained are just with statistical errors. Even in this case, if a warm start is provided, the quantum machine learning algorithm on the Qiskit's *qasm* simulator renders the exact valence band. In Figure 4(c), we show the implementation results for the valence band calculations using RBM wherein the gate-set is implemented on real IBM quantum devices, namely the *ibmq_toronto* and *ibmq_sydney*, both of which are 27 qubit devices. We see the results are in excellent agreement with the exact diagonalization when a warm start is provided along with *measurement error mitigation*.

Band Structure of Monolayer Graphene. Much like h-BN, monolayer graphene also consists of two atoms in its unit cells. However, unlike the previous case, both atomic centers

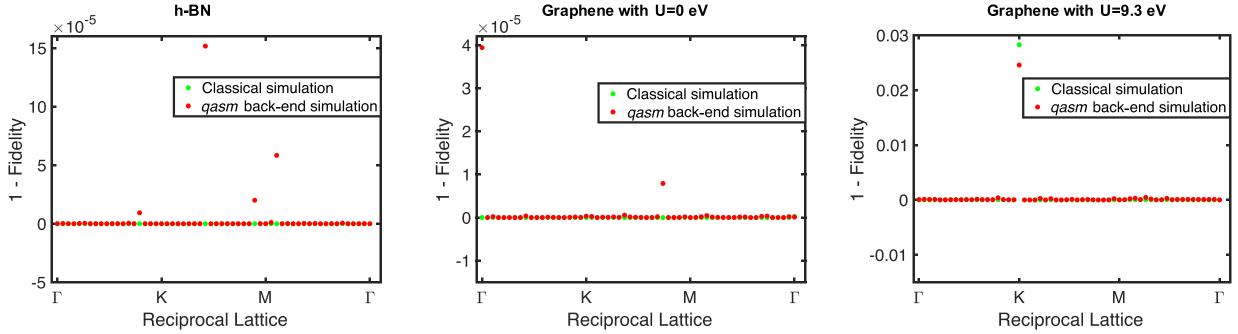


Figure 6. Error in fidelity $(1 - F)$ is plotted as a function of the reciprocal lattice vector (k) for classical simulation and *qasm* back-end simulation.

are made up of carbon. Also, similar to h-BN, in the case of graphene, the levels involving the other valence orbitals, the $2s$, $2p_x$, and $2p_y$, are either quite far above or far below the Fermi level. The orbital responsible for electrical conduction is just the $2p_z$ orbital, and hence, a tight-binding Hamiltonian for the valence and conduction band with third-nearest neighbor interaction is constructed by taking into account the frontier $2p_z$ orbital on each of the carbons. The resultant matrix as before is a 4 × 4 matrix including spin-degeneracy (see the Supporting Information). We introduce electron-electron interaction in graphene using the Fermi-Hubbard model with an onsite repulsion parameter U between opposite spins. In order to simulate graphene, the number of visible units and the number of hidden units is equal to 2. Therefore, 2 qubits to represent the visible nodes, 2 qubits to represent the hidden nodes, and in addition to that, 4 ancilla qubits are required. In total, the number of qubits required is equal to 8. The number of quantum gates required to sample Gibbs distribution is 4 single qubit rotation gates (R_y), 16 controlled−controlled rotation gates $(C - C - R_y)$, and 24 bit-flip (X) gates.

The band structure of monolayer graphene is calculated using the IBM Qiskit's *qasm* simulator and by running the QML algorithm on the IBM-Q quantum computers. Figure 5(a) shows the results for the band structure of graphene at zero U using the classical simulation. As before, the results are overlaid on top of the eigenvalues obtained from exact diagonalization of the 4 × 4 Hamiltonian. In Figure 5(b), we show the band structure of the graphene for $U = 0$ calculated using the Qiskit's *qasm* simulator. Finally, in Figure 5(c), we show the results of the quantum machine learning algorithm for calculation of the band structure of graphene on IBM-Q quantum computers, *ibmq_toronto*, and *ibmq_sydney*. Even for the case of graphene, the results are in good agreement with the exact diagonalization when a warm start is provided along with *measurement error mitigation*.

To show the band splitting for a nonzero on-site repulsion U, the Fermi level is shifted by a chemical potential $\mu = 15$ eV, which controls the filling of electrons. Figure 5(d,e) plots the band structures of graphene for $U = 9.3$ eV obtained using the classical simulation, Qiskit's *qasm* back end, and the actual implementation on an IBM quantum computer. The RBM results are again in good agreement with that from exact diagonalization in all of the cases.

Fidelity. To verify if the eigenstates provided by the QML algorithm match those obtained from exact diagonalization, the fidelity for each k point is calculated. It can be seen from Figure 6 that the error (1-Fidelity) is very small for classical simulation and simulation on the *qasm* back-end for both materials. The fidelity is calculated as follows

$$Fidelity = |\langle \Psi | \Phi \rangle|^2$$

where $|\Psi\rangle$ is the eigenvector obtained from QML, and $|\Phi\rangle$ is the eigenvector obtained from exact diagonalization.

4. CONCLUSION

The primary goal of this study was to examine the performance of an RBM on a NISQ device in order to calculate the electronic structure of materials. In this work, the materials that were taken under consideration were monolayer hexagonal boron nitride (h-BN) and monolayer graphene, both of which are two-dimensional solids. A tight-binding and a Hubbard Hamiltonian were constructed for h-BN and graphene, respectively. By using an RBM and a quantum circuit to sample Gibbs distribution, the valence band energies for each of the two materials were obtained. In the case of graphene, the simulations were performed first for the case when the Hubbard U is equal to 0 and then for the case of nonzero U. The band splitting for the case of nonzero U was also shown. The simulations for both graphene and h-BN were done using IBM's *Qiskit* framework as well as on real IBM quantum computing platforms.

Implementing RBM classically can either use Maximum-likelihood based gradient descent (which has a time complexity that is exponential in the size of the smallest layer)[33] or Contrastive Divergence using Gibbs sampling, a Markov Chain Monte Carlo (MCMC) method (which is a more efficient approach) to estimate the gradients.[34] The time complexity for training an RBM in the classical case scales as $O(N)$, where N is the size of the training data, while the implementation of RBM on a quantum computer has been shown to have quadratic speed-ups.[35] Also, computing the ground state of a given Hamiltonian using exact diagonalization has a complexity of the order of q^3, where q is the dimension of the column space of a given matrix.[36] However, setting $k = \max\left(1, \frac{|w_{ij}|}{2}\right)$ provides a constant lower bound in the probability of successful sampling, and thus the complexity for one iteration scales as $O(mnN)$, where N is the number of successful samplings required to get the distribution $P(\mathbf{x})$.

The current quantum machine learning method could calculate only on the ground state energy of the periodic systems, i.e., the valence band, an extension needed to treat systems with multiple valence bands[37] or to procure higher order energy bands. This can be done by sampling the orthogonal subspace of the previously computed valence band.[38] Also, to calculate the transition matrix elements, the valence and conduction Bloch wavevectors should be obtained. The expectation value of an operator with respect to the ground state may be calculated using the Hellmann−Feynman method.[39] Here, the effect of noise on quantum machine learning is not fully explored, while the Qiskit's *qasm* simulator and IBM-Q noisy quantum computers show the effect of noise on quantum optimization. With the field of quantum computing developing rapidly, the curiosity of combining machine learning and quantum computing has led to very interesting research. With the development of quantum computers and their capability to scale very fast, quantum machine learning can prove to be useful in not only electronic structure methods but also as a significant tool in developing new materials and understanding complex phenomena.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c00294.

> The supporting information consists of the following sections: General tight-binding Hamiltonian for system of two sublattices, honeycomb lattices: graphene and h-BN, scaling, implementation details, and result (without warm start or measurement error mitigation) (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Sabre Kais − *Department of Chemistry, Department of Physics and Astronomy, and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, Indiana 47907, United States;* orcid.org/0000-0003-0574-5346; Email: kais@purdue.edu

Authors

Shree Hari Sureshbabu − *School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907, United States*

Manas Sajjan − *Department of Chemistry, Purdue University, West Lafayette, Indiana 47907, United States*

Sangchul Oh − *Department of Chemistry, Purdue University, West Lafayette, Indiana 47907, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c00294

Notes

The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team.

The authors declare no competing financial interest.

Data and Model Availability. The input Hamiltonians corresponding to h-BN and graphene can be found in section 2 of the Supporting Information. Data will be made available upon reasonable request to the corresponding author. The codes associated with the classical simulation, simulation on the *qasm* back end, and the implementation on IBM's quantum computing devices will be made available from the corresponding author upon reasonable request.

■ ACKNOWLEDGMENTS

We would like to thank Dr. Ruth Pachter, AFRL, for many useful discussions. AFRL support is acknowledged. We acknowledge the National Science Foundation under award number 1955907. This material is also based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers. We also acknowledge the use of IBM-Q and thank them for the support.

■ REFERENCES

(1) Jordan, M. I.; Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255−260.

(2) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016; pp 770−778, DOI: 10.1109/CVPR.2016.90.

(3) Sak, H.; Senior, A.; Rao, K.; Irsoy, O.; Graves, A.; Beaufays, F.; Schalkwyk, J. Learning acoustic frame labeling for speech recognition with recurrent neural networks. *IEEE international conference on acoustics, speech and signal processing (ICASSP)*; 2015; pp 4280−4284, DOI: 10.1109/ICASSP.2015.7178778.

(4) Bojarski, M.; Testa, D. D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; Zhang, X.; Zhao, J.; Zieba, K. End to End Learning for Self-Driving Cars. 2016, *arXiv preprint arXiv:1604.07316*. https://arxiv.org/abs/1604.07316 (accessed 2021-06-11).

(5) Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborová, L. Machine learning and the physical sciences. *Rev. Mod. Phys.* **2019**, *91*, 045002.

(6) Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F. G. S. L.; Buell, D. A.; Burkett, B.; Chen, Y.; Chen, Z.; Chiaro, B.; Collins, R.; Courtney, W.; Dunsworth, A.; Farhi, E.; Foxen, B.; Fowler, A.; Gidney, C.; Giustina, M.; Graff, R.; Guerin, K.; Habegger, S.; Harrigan, M. P.; Hartmann, M. J.; Ho, A.; Hoffmann, M.; Huang, T.; Humble, T. S.; Isakov, S. V.; Jeffrey, E.; Jiang, Z.; Kafri, D.; Kechedzhi, K.; Kelly, J.; Klimov, P. V.; Knysh, S.; Korotkov, A.; Kostritsa, F.; Landhuis, D.; Lindmark, M.; Lucero, E.; Lyakh, D.; Mandrà, S.; McClean, J. R.; McEwen, M.; Megrant, A.; Mi, X.; Michielsen, K.; Mohseni, M.; Mutus, J.; Naaman, O.; Neeley, M.; Neill, C.; Niu, M. Y.; Ostby, E.; Petukhov, A.; Platt, J. C.; Quintana, C.; Rieffel, E. G.; Roushan, P.; Rubin, N. C.; Sank, D.; Satzinger, K. J.; Smelyanskiy, V.; Sung, K. J.; Trevithick, M. D.; Vainsencher, A.; Villalonga, B.; White, T.; Yao, Z. J.; Yeh, P.; Zalcman, A.; Neven, H.; Martinis, J. M. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505−510.

(7) Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79.

(8) Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195−202.

(9) Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. 2013, *arXiv preprint arXiv:1307.0411*. https://arxiv.org/abs/1307.0411 (accessed 2021-06-11).

(10) Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503.

(11) Neven, H.; Rose, G.; Macready, W. G. Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization. 2008, *arXiv preprint arXiv:0804.4457*. https://arxiv.org/abs/0804.4457 (accessed 2021-06-11).

(12) Neven, H.; Denchev, V. S.; Rose, G.; Macready, W. G. Training a binary classifier with the quantum adiabatic algorithm. 2008, *arXiv preprint arXiv:0811.0416*. https://arxiv.org/abs/0811.0416 (accessed 2021-06-11).

(13) Neven, H.; Denchev, V. S.; Rose, G.; Macready, W. G. Training a large scale classifier with the quantum adiabatic algorithm. 2009, *arXiv preprint arXiv:0912.0779*. https://arxiv.org/abs/0912.0779 (accessed 2021-06-11).

(14) Das Sarma, S.; Deng, D.-L.; Duan, L.-M. Machine learning meets quantum physics. *Phys. Today* **2019**, *72*, 48−54.

(15) Xu, Q.; Xu, S. Neural network state estimation for full quantum state tomography. 2018, *arXiv preprint arXiv:1811.06654*. https://arxiv.org/abs/1811.06654 (accessed 2021-06-11).

(16) Cong, I.; Choi, S.; Lukin, M. D. Quantum convolutional neural networks. *Nat. Phys.* **2019**, *15*, 1273−1278.

(17) Carleo, G.; Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **2017**, *355*, 602−606.

(18) Torlai, G.; Mazzola, G.; Carrasquilla, J.; Troyer, M.; Melko, R.; Carleo, G. Neural-network quantum state tomography. *Nat. Phys.* **2018**, *14*, 447−450.

(19) Aspuru-Guzik, A.; Dutoi, A. D.; Love, P. J.; Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **2005**, *309*, 1704−1707.

(20) Kais, S. *Quantum Information and Computation for Chemistry*; John Wiley & Sons, Ltd.: 2014; Chapter 1, pp 1−38, DOI: 10.1002/9781118742631.ch01.

(21) Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P. J.; Aspuru-Guzik, A.; O'brien, J. L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213.

(22) Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J. M.; Gambetta, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242−246.

(23) Daskin, A.; Kais, S. Direct application of the phase estimation algorithm to find the eigenvalues of the hamiltonians. *Chem. Phys.* **2018**, *514*, 87−94.

(24) Xia, R.; Kais, S. Quantum machine learning for electronic structure calculations. *Nat. Commun.* **2018**, *9*, 4195.

(25) Kanno, S.; Tada, T. Many-body calculations for periodic materials via restricted Boltzmann machine-based VQE. *Quantum Science and Technology* **2021**, *6*, 025015.

(26) Wang, J. I.-J.; Rodan-Legrain, D.; Bretheau, L.; Campbell, D. L.; Kannan, B.; Kim, D.; Kjaergaard, M.; Krantz, P.; Samach, G. O.; Yan, F.; Yoder, J. L.; Watanabe, K.; Taniguchi, T.; Orlando, T. P.; Gustavsson, S.; Jarillo-Herrero, P.; Oliver, W. D. Coherent control of a hybrid superconducting circuit made with graphene-based van der Waals heterostructures. *Nat. Nanotechnol.* **2019**, *14*, 120−125.

(27) Calafell, I. A.; Cox, J.; Radonjić, M.; Saavedra, J.; de Abajo, F. G.; Rozema, L.; Walther, P. Quantum computing with graphene plasmons. *npj Quantum Information* **2019**, *5*, 37.

(28) Dean, C. R.; Young, A. F.; Meric, I.; Lee, C.; Wang, L.; Sorgenfrei, S.; Watanabe, K.; Taniguchi, T.; Kim, P.; Shepard, K. L.; Hone, J. Boron nitride substrates for high-quality graphene electronics. *Nat. Nanotechnol.* **2010**, *5*, 722−726.

(29) Gottscholl, A.; Kianinia, M.; Soltamov, V.; Orlinskii, S.; Mamin, G.; Bradac, C.; Kasper, C.; Krambrock, K.; Sperlich, A.; Toth, M.; Aharonovich, I.; Dyakonov, V. Initialization and read-out of intrinsic spin defects in a van der Waals crystal at room temperature. *Nat. Mater.* **2020**, *19*, 540−545.

(30) Gottscholl, A.; Diez, M.; Soltamov, V.; Kasper, C.; Sperlich, A.; Kianinia, M.; Bradac, C.; Aharonovich, I.; Dyakonov, V. Room Temperature Coherent Control of Spin Defects in hexagonal Boron Nitride. 2020, *arXiv preprint arXiv:2010.12513*. https://arxiv.org/abs/2010.12513 (accessed 2021-06-11).

(31) Exarhos, A. L.; Hopper, D. A.; Patel, R. N.; Doherty, M. W.; Bassett, L. C. Magnetic-field-dependent quantum emission in hexagonal boron nitride at room temperature. *Nat. Commun.* **2019**, *10*, 222.

(32) Mari, A.; Bromley, T. R.; Izaac, J.; Schuld, M.; Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **2020**, *4*, 340.

(33) Fischer, A.; Igel, C. Training restricted Boltzmann machines: An introduction. *Pattern Recognition* **2014**, *47*, 25−39.

(34) Carreira-Perpinan, M. A.; Hinton, G. E. On contrastive divergence learning. *Aistats*; 2005; pp 33−40 .

(35) Wiebe, N.; Kapoor, A.; Svore, K. M. Quantum deep learning. *Quantum Inf. Comput.* **2016**, *16*, 541−587.

(36) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.

(37) Cerasoli, F. T.; Sherbert, K.; Sławińska, J.; Nardelli, M. B. Quantum computation of silicon electronic band structure. *Phys. Chem. Chem. Phys.* **2020**, *22*, 21816−21822.

(38) Sajjan, M.; Sureshbabu, S. H.; Kais, S. Quantum Machine-Learning for Eigenstate Filtration in Two-Dimensional Materials. 2021, *arXiv preprint arXiv:2105.09488*. https://arxiv.org/abs/2105.09488 (accessed 2021-06-11).

(39) Oh, S. Quantum computational method of finding the ground-state energy and expectation values. *Phys. Rev. A: At., Mol., Opt. Phys.* **2008**, *77*, 012326.