entropy

Article

Hybrid Quantum-Classical Neural Network for Calculating Ground State Energies of Molecules

Rongxin Xia [1,2,3] **and Sabre Kais** [1,2,3,*]

[1] Department of Chemistry, Purdue University, West Lafayette, IN 47906, USA; xiar@purdue.edu
[2] Department of Physics and Astronomy, Purdue University, West Lafayette, IN 47906, USA
[3] Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47906, USA
* Correspondence: kais@purdue.edu

Received: 3 July 2020; Accepted: 27 July 2020; Published: 29 July 2020

Abstract: We present a hybrid quantum-classical neural network that can be trained to perform electronic structure calculation and generate potential energy curves of simple molecules. The method is based on the combination of parameterized quantum circuits and measurements. With unsupervised training, the neural network can generate electronic potential energy curves based on training at certain bond lengths. To demonstrate the power of the proposed new method, we present the results of using the quantum-classical hybrid neural network to calculate ground state potential energy curves of simple molecules such as H_2, LiH, and BeH_2. The results are very accurate and the approach could potentially be used to generate complex molecular potential energy surfaces.

Keywords: quantum neural network; quantum machine learning; electronic structure calculation

1. Introduction

Quantum computing has shown its great potential in advancing quantum chemistry research [1]. Many quantum algorithms have been proposed to solve quantum chemistry problems [2–4], such as the Phase Estimation Algorithm; Aspuru-Guzik et al. [5–8] to calculate eigenstate energies of simple molecules; the Variational Quantum Eigensolver (VQE) [9–11] to solve electronic structure problems; quantum algorithms for open quantum dynamics [12]; and benchmark calculations for two-electron molecules conducted on quantum computers [13]. Using quantum computing techniques to perform machine learning tasks [14] has also received much attention recently including quantum data classification [15,16], quantum generative learning [17,18], and quantum neural network approximating nonlinear functions [19]. So far, applying the various quantum machine learning techniques to quantum chemistry is a natural extension [20,21]. However, previous studies focused solely on quantum circuits with only a few nonlinear operations, which are introduced by data encoding [19,22] or repeated measurements until success [23]. Moreover, recently Sim et al. [24] shows increasing the number of layers of the parameterized quantum circuit (PQC) would reach saturation and may not improve the performance when the number of layers is large enough. Furthermore, the nonlinearity is the most important part for the classical neural network [25] which makes neural networks able to produce complex results [23,26,27]. Therefore, quantum machine learning should not solely focus on PQC and nonlinear operations are needed for the quantum neural network.

To solve this problem, here we introduce a new hybrid quantum-classical neural network, by combining quantum computing and classical computing with measurements between the parameterized quantum circuits. In this paper, we first give a detailed description of the whole structure of the hybrid quantum-classical neural network. We then present numerical simulations by using the new hybrid quantum-classical neural network to calculate ground state energies of different molecular systems. The calculated ground state energies are very accurate, which demonstrate the potential of the proposed hybrid quantum-classical neural network to generate potential energy surfaces.

2. Results

We propose a new structure of quantum-classical hybrid neural network by connecting the quantum part (quantum layer) with the classical part (classical layer). For a classical neural network, each artificial neuron is normally constructed by linear connected layers, with nonlinear activation functions connected at the end, as shown in the left part of Figure 1. In this work, we replace the linear part by the quantum circuit as shown in the right side of Figure 1 to take advantage of possible speedup in quantum computation. In the meantime, we use expectation values of operators by measurements, which are nonlinear operations, to serve as the activation function. In this neural network set-up, the quantum circuit can be viewed as the quantum layer and the expectation values by measurements can be viewed as the classical layer. The input data is first encoded into quantum states and calculated by the quantum layer. The outputs are extracted as the expectation values by measurements. The two steps can be repeated several times to construct a hybrid multi-layer neural network. In our construction, the quantum layer is enabled by parameterized quantum circuits (PQC) [28]. We will give details about the hybrid quantum-classical neural network in the following sections.

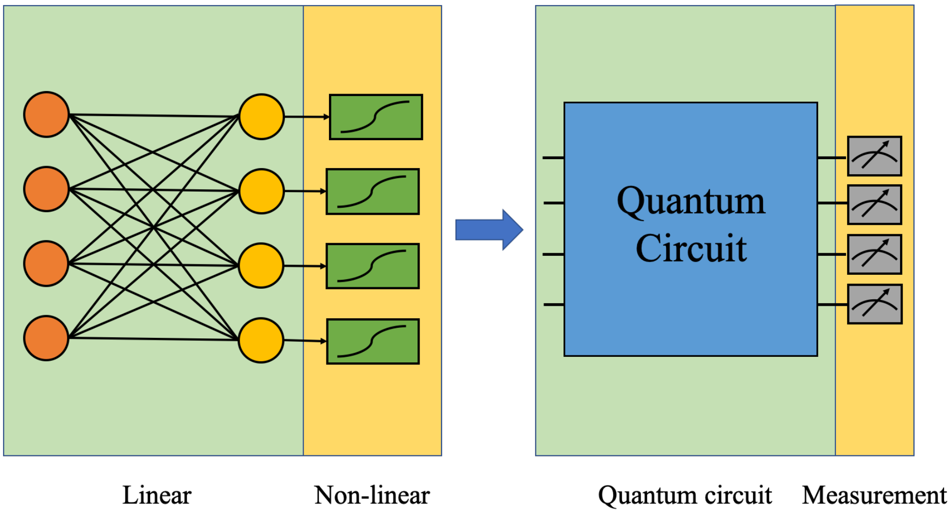


Figure 1. In the proposed quantum-classical hybrid neural network, the linear part in the classical neural network is replaced by the quantum circuits and the nonlinear part is replaced by measurements.

2.1. Quantum Layer

The quantum layer is enabled by a parameterized quantum circuit consisting of parameterized quantum gates, which allows the PQC to be optimized by adjusting the parameters to approximate wanted results. PQC has been widely used in many areas of quantum computing and quantum machine learning, such as in VQE [9–11], quantum autoencoder [20], and quantum generative learning [17]. In the following section, we will provide details of the quantum layer including encoding classical data into quantum circuits and parameterized quantum circuits.

2.1.1. Data Encoding

To implement the quantum layer, the first step is to encode the input classical data into a quantum state. Variational encoding [22] has been proposed to reduce the depth of quantum circuits and has been widely used in many quantum machine learning techniques [19,22,29,30]. Variational encoding is used to prepare a set of quantum gates with parameters generated by the input data and then initialize the state from the basic state with all qubits as $|0\rangle$ with these gates. For an array of data $\{a_0, a_1, ...a_{n-1}\}$, an example of variational encoding to encode n qubits is to prepare the gate G as

$$G = \otimes_{i=0}^{n-1} g_i(f_i(a_i)) \tag{1}$$

where g_i is a set of single qubit quantum gates on qubits i and f_i is a classical function to encode a_i as the parameter of g_i. The encoded state would be $G|0\rangle^{\otimes n}$. One simple example is given in our numerical simulations: we take the bond length, a, as the encoding data for each qubit. We choose f_i as the identity function and g_i as $R_y H$, where R_y is the rotation-y gate and H is the Hadamard gate. Thus, the encoded quantum state would be $(\otimes_{i=0}^{n-1} R_y(a)H)|0\rangle^{\otimes n}$.

In most variational encoding the depth of the circuit needed to encode the data would be $O(1)$ [29] for that the number of quantum gates to initialize the quantum state is fixed, which makes variational encoding more suitable for Noisy Intermediate-Scale Quantum (NISQ) devices [31]. Furthermore, recently it has been shown how the variational encoding may help to introduce nonlinearity features in quantum circuits [22,32]. Variational encoding can only be implemented at the beginning of the quantum circuit, but connections between multiple PQC also need to be nonlinear. To enable nonlinear connections, we introduce measurements as connections between multiple PQC. In the numerical simulations, we will be using the variational encoding to perform the simulation and discuss implementing the quantum circuits on NISQ device.

2.1.2. Parameterized Quantum Circuit

A parameterized quantum circuit, also known as a variational quantum circuit [10,28], is a quantum circuit consisting of parameterized gates with fixed depth. This is the main part of the quantum layer to perform the calculation. The parameterized quantum circuit consists of one-qubit gates as well as $CNOT$. Some more complicated gates may also be used in PQC which can be decomposed into one qubit gates and $CNOT$ [33]. In general, an n qubits PQC can be written as

$$U(\vec{\theta})|\psi\rangle = (\prod_{i=1}^{m} U_i)|\psi\rangle \tag{2}$$

where $U(\vec{\theta})$ is the set of universal gates and m is the number of quantum gates. $\vec{\theta}$ is the set of parameters $\{\theta_0, \theta_1 ....\theta_{k-1}\}$, where k is the total number of parameters and $|\psi\rangle$ is the encoded quantum state after data encoding. For each unitary gate U_i, it may be a quantum gate which does not require a parameter or a quantum gate which takes parameters. Examples of the unitary gate taking parameters are rotational gates, $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$, which are given by

$$R_x(\theta) = e^{-i\frac{\theta}{2}\sigma_x} \qquad R_y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} \qquad R_z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} \tag{3}$$

where σ_x, σ_y, and σ_z are Pauli matrices. The operation of U can be modified by changing parameters $\vec{\theta}$. Thus, the output state can be optimized to approximate the wanted state by changing parameters $\vec{\theta}$. By optimizing the parameters used in $U(\vec{\theta})$, PQC approximates the wanted quantum states.

2.2. Classical Layer

The classical layer in our construction of the quantum-classical hybrid neural network is to serve as the activation function connecting different quantum layers. To achieve nonlinearity, the classical layer is enabled by measurements—expectation values of operators on each qubit of the PQC, for example, $\langle\sigma_z^i\rangle$ of each qubit i as the classical layer, which would also serve as nonlinear operations. Expectation values of operators can save complexity because quantum tomography is exponentially hard. Though the expectation values of operators may lose some information compared to quantum tomography, some work used expectation values of operators as connections between quantum computation and classical computation and showed great success [34], which indicates expectation values of operators are capable of extracting useful information from quantum circuits.

2.3. Numerical Simulations

To demonstrate the power of the proposed quantum-classical hybrid neural network, we present results for calculating the ground state energies of simple molecular systems: H_2, LiH, and BeH_2. The inputs for the unsupervised learning are bond lengths and the outputs are the ground state energies. The whole procedure consists of first training the neural network with some bond lengths and then testing the neural network with other bond lengths to generate the whole potential energy curve.

2.3.1. Constructions of the Quantum Layer

The quantum layer consists of two parts: the variational encoding part and PQC part. We choose to use the variational encoding to decrease the depth of the quantum circuit so that it can be implemented on NISQ devices. The construction of the quantum layer follows [29,34]. The input state is initialized as $(\otimes_{i=0}^{n-1} R_y(a)H)|0\rangle^{\otimes n}$, where a is the bond length, H is the Hadamard gate, and R_y is the rotation-y gate. We only have one bond length while the number of qubits of the PQC is n; we decided to follow the variational encoding in [29] to encode each qubit with same value. The number of qubits n is equal to the number of qubits of the corresponding Hamiltonian. The quantum computation part is to use a simple PQC consisting of R_y and $CNOT$ gates, which can be written as

$$\begin{cases} \prod_{j=0}^{n-1}(\otimes_{i=0}^{n-1} R_y(w_{i+n\times j}))(CNOT_{n-3,n-2}...CNOT_{3,4}CNOT_{1,2})(CNOT_{n-2,n-1}...CNOT_{2,3}CNOT_{0,1}) & n \text{ is even} \\ \prod_{j=0}^{n-1}(\otimes_{i=0}^{n-1} R_y(w_{i+n\times j}))(CNOT_{n-2,n-1}...CNOT_{3,4}CNOT_{1,2})(CNOT_{n-3,n-2}...CNOT_{2,3}CNOT_{0,1}) & n \text{ is odd} \end{cases} \quad (4)$$

where w are adjustable parameters, R_y represents rotation-y gate, and $CNOT_{m,n}$ represents $CNOT$ gate with m as the control qubit and n is the target qubit. To achieve better entanglement of the qubits before appending nonlinear operations, the n qubits PQC has n repeated layers in our simulation. By optimizing the parameters, the general PQC tries to approximate arbitrary states so that it can be used for different specific molecules. The construction of the PQC for three qubits is illustrated in the blue part of Figure 2, and the construction of the PQC for four qubits is illustrated in the blue part of Figure 3.

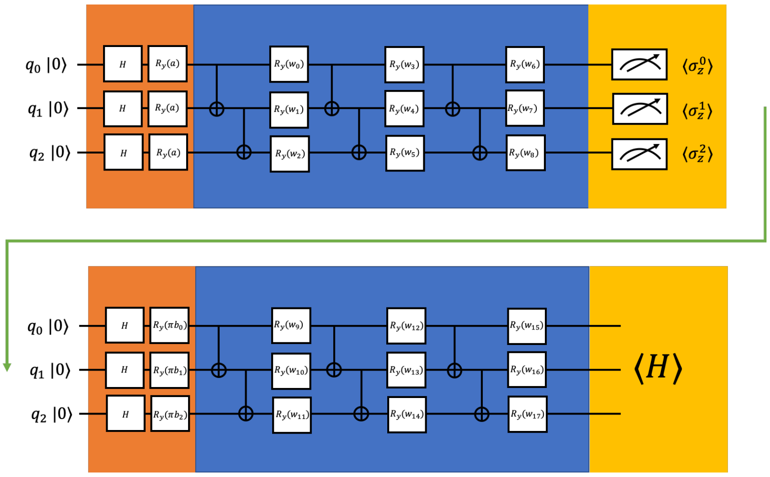


Figure 2. The example constructions of the proposed hybrid quantum-classical neural network for 3 qubits (odd qubits number). The orange parts are the data encoding, the blue parts are parameterized quantum circuits, and the yellow parts are measurements. The first measurements serve as nonlinear operations connecting two PQC. a is the input bond length, bs are the expectation values of σ_z, and ws are adjustable parameters.

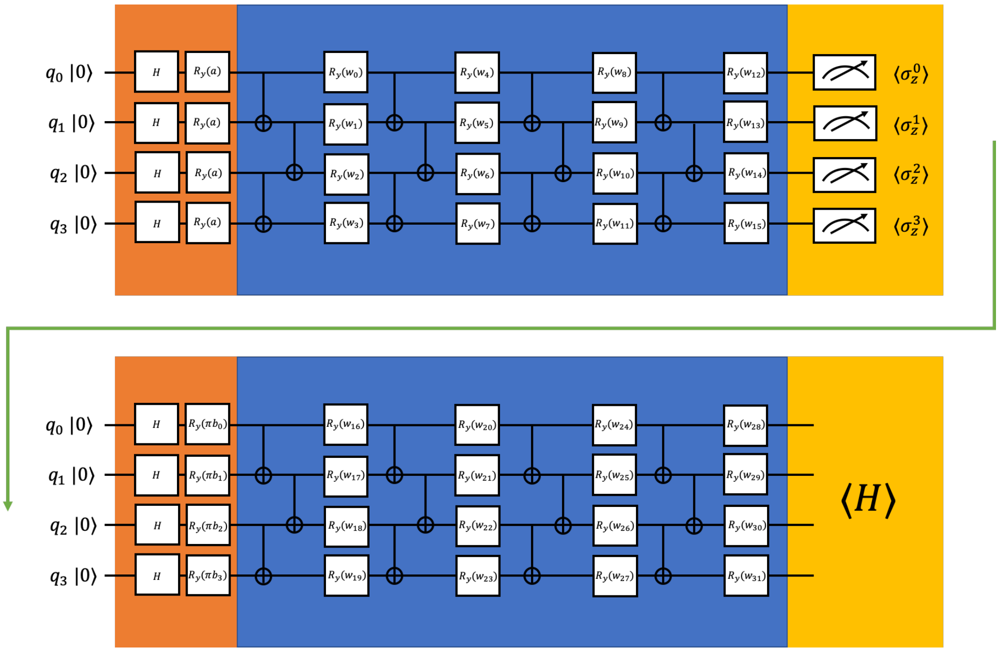


Figure 3. The example constructions we use for the 4 qubits H_2 calculation (even qubits number). The orange parts are the data encoding, the blue parts are parameterized quantum circuits, and the yellow parts are measurements. The first measurements serve as nonlinear operations connecting two PQC. a is the input bond length, bs are the expectation values of σ_z, and ws are adjustable parameters.

2.3.2. Constructions of the Classical Layer

The classical layer is enabled by expectation values of the operators. In our numerical simulations, we are using $\langle\sigma_z^i\rangle$ for qubit i as the classical layer. The outputs from the classical layer will be encoded into another quantum layer. The second quantum layer is the same as the first one except for the data encoding part it would be $\otimes_{i=0}^{n-1} R_y(b_i\pi)H$, where b_i is the measured expectation value from qubit i. We multiply each b_i with π when encoding to change the range of the encoding data from $[-1,1]$ to $[-\pi,\pi]$ [35]. The construction of our proposed hybrid quantum-classical neural network is illustrated in Figure 3.

2.3.3. Cost Function

The cost function is defined as

$$f = \sum_j \langle\phi_j|H_j|\phi_j\rangle \tag{5}$$

where j represents the j_{th} input bond length of the training bond lengths. $|\phi_j\rangle$ is the final state of the proposed hybrid quantum-classical neural network with the input as the j_{th} input bond length and H_j is the Hamiltonian corresponding to the j_{th} input bond length. The idea of the cost function is similar to VQE: by optimizing the parameters, the expectation energy of $|\phi_j\rangle$ is minimized to approximate the ground state energy. The evaluation of the Hamiltonian can be done by techniques in [11]. The Hamiltomian can be written as the sum of tensor products of Pauli matrices $H = \sum_i c_i P_i$, where c_i is the coefficient and P_i is the tensor product of Pauli matrices. Instead of evaluating the whole Hamiltonian, we can evaluate each term of the Hamiltonian and the expectation of the Hamiltonian can be obtained by $\langle H\rangle = \sum_i c_i\langle P_i\rangle$, which does not need quantum tomography or take exponential complexity. The whole training procedure is done by taking a set of bond lengths and corresponding Hamiltonian and minimizing the cost function as equation (5). After the training, we test the model with other bond lengths.

2.3.4. Simulation Results

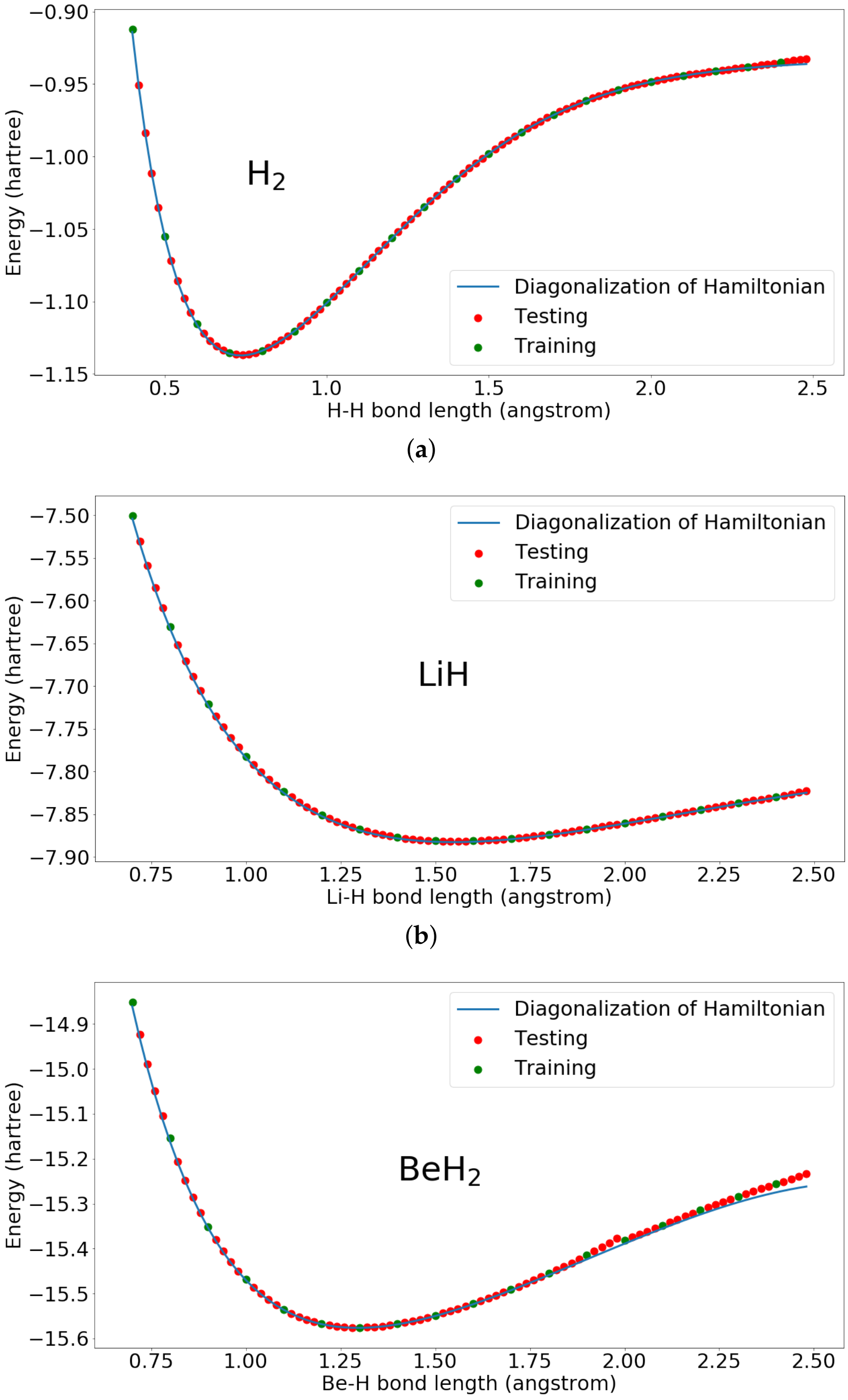
The Hamiltonian of the molecule systems can be derived by transforming the corresponding second quantization Hamiltonian into sum of tensor products of Pauli matrices. For H_2, we use the Jordan–Wigner transformation [36] to get a 4-qubit Hamiltonian. We decided to apply the complete active space (CAS) approach [37,38], which divides the orbitals into inactive orbitals such as always occupied low energy orbitals and always unoccupied high energy orbitals, and active orbitals, to reduce the number of qubits of LiH and BeH_2 Hamiltonian [4,11] and the reduced Hamiltonian is only of the active orbitals. For LiH, we assume the first two lowest energy spin orbitals are always occupied and use the binary code transformation [39] considering spin symmetry to save two qubits. We get an 8-qubit LiH Hamiltonian. For BeH_2, we assume the first two lowest energy spin orbitals are always occupied and the first two highest energy spin orbitals are never occupied, and use the binary code transformation [39] considering spin symmetry to save two qubits. We get an 8-qubit BeH_2 Hamiltonian.

In the simulation, H_2 used four qubits and 32 parameters. LiH and BeH_2 both used eight qubits and 128 parameters. The gate and parameter complexity of the proposed hybrid quantum-classical neural network in this simulation is $O(n^2)$, where n is the number of qubits of the Hamiltonian. Here, we present the results using our proposed hybrid quantum-classical neural network for ground state energies of H_2, LiH, and BeH_2 in Figures 4 and 5. We can see from these figures that the training data points converge very close to the diagonalization results without pre-known ground state information of the transformed Hamiltonian in Pauli matrices format. Furthermore, after training, by inputting the other bond lengths we can also get good approximating ground state energies with optimized parameters. BeH_2 has some deviation when the bond length is large, which may be solved by improving the parameterized quantum circuit. For example, the work in [24], which discusses expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, shows that increasing the depth of PQC will increase the expressibility and different constructions of PQC have also different expressibility.

Furthermore, to show that the intermediate nonlinear measurements improve the performance, we present the comparison of the results of our proposed hybrid quantum-classical neural network and quantum neural network removing intermediate measurements. The setting of the quantum neural network removing intermediate measurements is illustrated in Figure 6.

In Figure 7, we present the comparison of the results of our proposed hybrid quantum-classical neural network and quantum neural network removing intermediate measurements. The proposed hybrid quantum-classical neural network and quantum neural network, removing intermediate measurements, are trained with same set of bond lengths as in Figure 4. We can see without the intermediate nonlinear measurements, the quantum neural network can only achieve bad results. However, by adding the intermediate nonlinear measurements, the results converge closely to the diagonalization results.

The parameters of the proposed hybrid quantum-classical neural network and quantum neural network removing intermediate measurements, are initialized from a Gaussian distribution with standard deviation as 0.1 and mean as 0. Because different initialization of parameters will result in different starting of the optimization and may lead to different final results, to eliminate the effects of parameter initializations, here we present the quantitative comparison of the two constructions with four different parameter initialization from same Gaussian distribution with different random seeds. All are trained with the same set of the training bond lengths as in Figure 4. In the Table 1, we can see that our proposed quantum neural network performs better than the quantum neural network without intermediate measurements. Our simulation results show that adding intermediate nonlinear measurements would help to improve the expressibility of the PQC. Furthermore, adding intermediate measurements would also decrease the circuit depth which makes it more suitable for current NISQ devices.



(**a**)

(**b**)

(**c**)

Figure 4. Ground state energies of H_2, LiH, and BeH_2 calculated by the proposed hybrid quantum-classical neural network. (**a**) Ground state energies of H_2 calculated by the proposed hybrid quantum-classical neural network; (**b**) Ground state energies of LiH calculated by the proposed hybrid quantum-classical neural network; (**c**) Ground state energies of BeH_2 calculated by the proposed hybrid quantum-classical neural network.

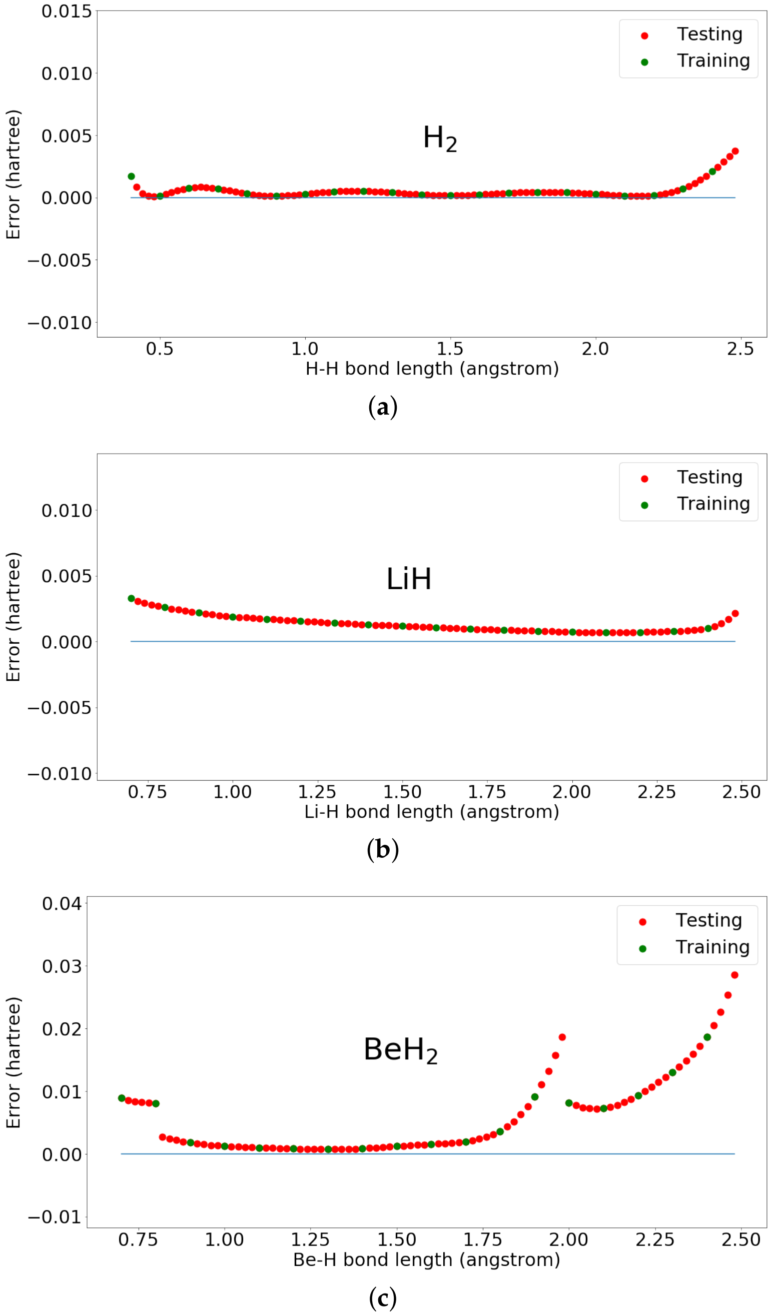


Figure 5. Errors of ground state energies of H_2, LiH, and BeH_2 calculated by the proposed hybrid quantum-classical neural network. (**a**) Errors of ground state energies of H_2 calculated by the proposed hybrid quantum-classical neural network; (**b**) Errors of ground state energies of LiH calculated by the proposed hybrid quantum-classical neural network; (**c**) Errors of ground state energies of BeH_2 calculated by the proposed hybrid quantum-classical neural network.

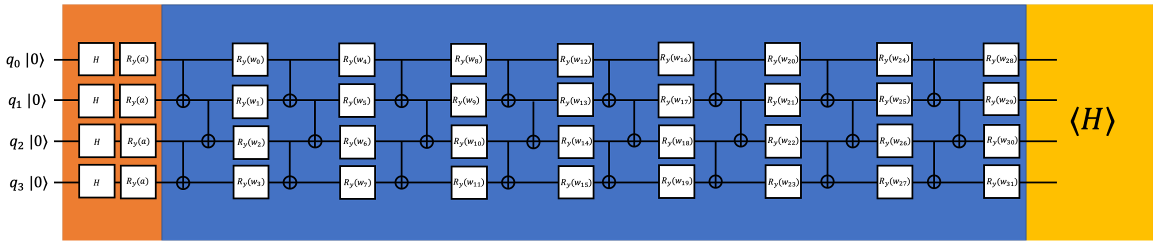
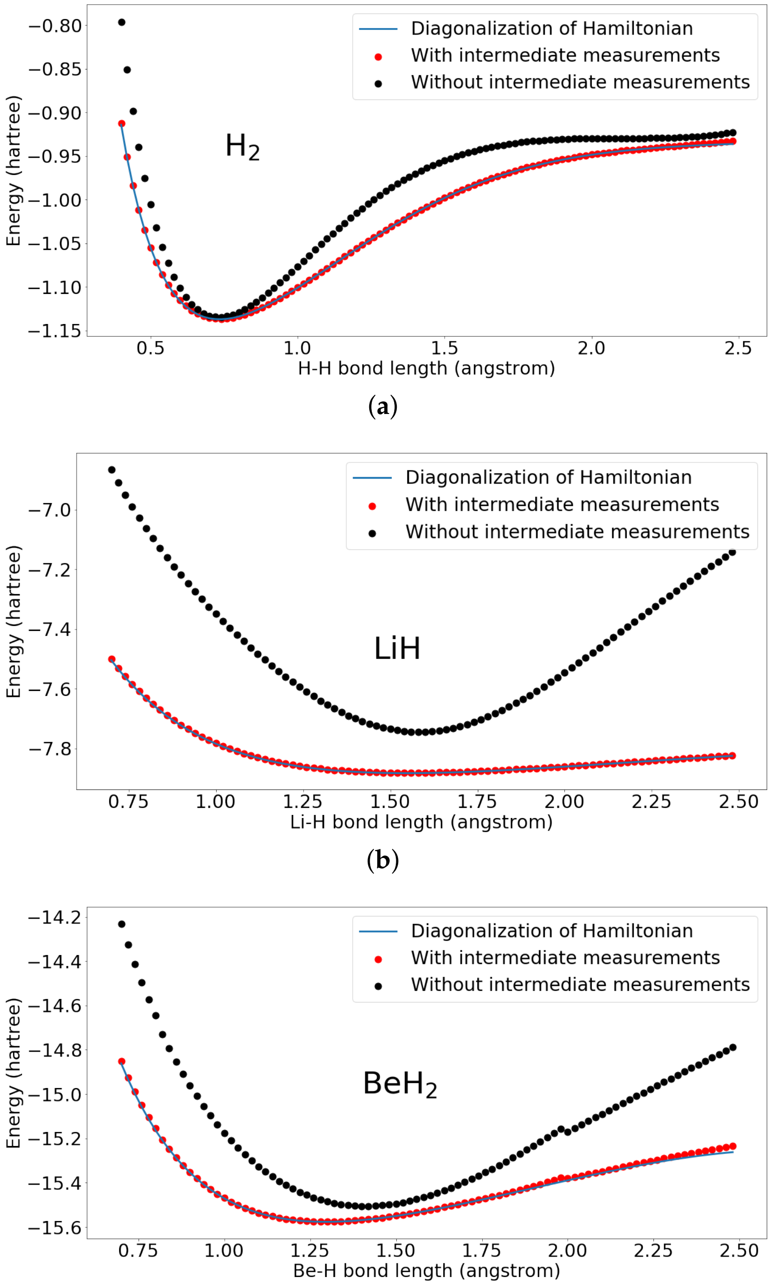

q_0 $|0\rangle$
q_1 $|0\rangle$
q_2 $|0\rangle$
q_3 $|0\rangle$
H
$R_y(a)$
$R_y(w_0)$ $R_y(w_1)$ $R_y(w_2)$ $R_y(w_3)$
$R_y(w_4)$ $R_y(w_5)$ $R_y(w_6)$ $R_y(w_7)$
$R_y(w_8)$ $R_y(w_9)$ $R_y(w_{10})$ $R_y(w_{11})$
$R_y(w_{12})$ $R_y(w_{13})$ $R_y(w_{14})$ $R_y(w_{15})$
$R_y(w_{16})$ $R_y(w_{17})$ $R_y(w_{18})$ $R_y(w_{19})$
$R_y(w_{20})$ $R_y(w_{21})$ $R_y(w_{22})$ $R_y(w_{23})$
$R_y(w_{24})$ $R_y(w_{25})$ $R_y(w_{26})$ $R_y(w_{27})$
$R_y(w_{28})$ $R_y(w_{29})$ $R_y(w_{30})$ $R_y(w_{31})$
$\langle H \rangle$

Figure 6. Constructions of the quantum neural network removing intermediate measurements for H_2.



(**a**)

(**b**)

(**c**)

Figure 7. Results of H_2, LiH, and BeH_2 by the proposed hybrid quantum-classical neural network and the quantum neural network removing intermediate measurements. With intermediate measurements represents the results by our proposed hybrid quantum-classical neural network. Without intermediate measurements represents the quantum neural network removing the intermediate measurements. Both are trained with same set of bond lengths as in Figure 4 and same parameter initialization. (**a**) Ground state energies of H_2 calculated by the proposed hybrid quantum-classical neural network and the quantum neural network removing intermediate measurements; (**b**) Ground state energies of LiH calculated by the proposed hybrid quantum-classical neural network and the quantum neural network removing intermediate measurements; (**c**) Ground state energies of BeH_2 calculated by the proposed hybrid quantum-classical neural network and the quantum neural network removing intermediate measurements.

Table 1. Results for the proposed hybrid quantum-classical neural network and the quantum neural network removing intermediate measurements. With intermediate measurements represents the proposed hybrid quantum-classical neural network. Without intermediate measurements represents the quantum neural network removing intermediate measurements. $\sum_{training} Error$ represents the sum of the error of the calculated ground state energies on the training set. $\sum_{testing} Error$ represents the sum of the error of the calculated ground state energies on the testing set. Each result is calculated by 4 different parameter initialization and presented as means and standard deviations. It can be seen that adding intermediate measurements to introduce nonlinear options would help to improve the performance.

Constructions	$\sum_{training} Error$	$\sum_{testing} Error$
With intermediate measurements (H_2)	0.0271 ± 0.0246	0.1178 ± 0.1061
Without intermediate measurements (H_2)	0.6296 ± 0.0151	2.2755 ± 0.0677
With intermediate measurements (LiH)	0.0287 ± 0.0038	0.1178 ± 0.0190
Without intermediate measurements (LiH)	4.7638 ± 1.4444	19.1479 ± 5.7715
With intermediate measurements (BeH_2)	0.1253 ± 0.0552	0.5613 ± 0.2483
Without intermediate measurements (BeH_2)	3.7280 ± 0.6497	14.8440 ± 2.3747

3. Materials and Methods

Orbital integrals in the second quantization Hamiltonian are calculated by STO-3G minimal basis using PySCF [40] and the transformation is done by OpenFermion [41]. The simulation is done by Qiskit [42]. The tensor production orders in OpenFermion and Qiskit are opposite. For a n qubits, the tensor production order in OpenFermion is $q_0 \otimes q_1 ... \otimes q_{n-1}$, while the tensor production order in Qiskit is $q_{n-1} \otimes q_{n-2} ... \otimes q_0$. We decided to follow the tensor production order in OpenFermion. In simulation, we treat the qubit indexed in Qiskit reversely. For n qubits, the qubit indexed as q_0 in Qiskit is treated as q_{n-1}, the qubit indexed as q_1 in Qiskit is treated as q_{n-2}, etc. By doing this, we change the tensor production order in Qiskit same as OpenFermion. The optimization is performed by the Broyden–Fletcher–Goldfarb–Shanno algorithm [43] with maximum 500 iterations and gradient norm tolerance to stop as 10^{-5}. In the simulation, the expectation of the operator is simulated by matrix production of the operator matrix and the Hamiltonian can be treated as a single operator. To save the simulation time, instead of evaluating each $\langle P_i \rangle$ to get $\langle H \rangle = \sum_i c_i \langle P_i \rangle$, we treat H as a single operator and only evaluate once.

4. Conclusions

In this work, we proposed a new hybrid quantum-classical neural network by combing PQC and measurements to achieve nonlinear operations in quantum computing. We have shown that the proposed hybrid quantum-classical neural network can be trained to obtain the electronic energies at certain bond lengths and then generate the whole potential energy curve. The results of H_2, LiH, and BeH_2 are very accurate and demonstrate the power of the proposed hybrid quantum-classical neural network.

Furthermore, we show that the intermediate nonlinear measurements are very important in comparison with quantum neural network removing the intermediate measurements. The intermediate nonlinear measurements can reduce the circuit depth and are more suitable for NISQ devices. Although the method is used to generate one-dimensional potential energy curves, the approach is general and could be generalized to generate multidimensional potential energy surfaces, for example, changing the inputs from the bond lengths to multidimensional coordinates. This will be done in future work.

Author Contributions: S.K. designed the research. R.X. performed the calculations. Both discussed the results and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the financial support by the U.S. Department of Energy (Office of Basic Energy Sciences) under Award No. DE-SC0019215 and the Integrated Data Science Initiative Grant (IDSI F.90000303), Purdue University.

Acknowledgments: The authors would like to thank Zixuan Hu for critical reading and useful discussions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript.

VQE	Variantional Quantum Eigensolver
PQC	parameterized quatum circuit
NISQ	Noisy Intermediate-Scale Quantum

References

1. Kais, S. *Quantum Information and Computation for Chemistry*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
2. Cao, Y.; Romero, J.; Olson, J.P.; Degroote, M.; Johnson, P.D.; Kieferová, M.; Kivlichan, I.D.; Menke, T.; Peropadre, B.; Sawaya, N.P.; et al. Quantum chemistry in the age of quantum computing. *Chem. Rev.* **2019**, *19*, 10856–10915. [CrossRef]
3. Xia, R.; Bian, T.; Kais, S. Electronic structure calculations and the ising hamiltonian. *J. Phys. Chem.* **2017**, *122*, 3384–3395. [CrossRef]
4. Bian, T.; Murphy, D.; Xia, R.; Daskin, A.; Kais, S. Quantum computing methods for electronic states of the water molecule. *Mol. Phys.* **2019**, 117, 2069–2082. [CrossRef]
5. Aspuru-Guzik, A.; Dutoi, A.D.; Love, P.J.; Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **2005**, *309*, 1704–1707. [CrossRef] [PubMed]
6. Wang, H.; Kais, S.; Aspuru-Guzik, A.; Hoffmann, M.R. Quantum algorithm for obtaining the energy spectrum of molecular systems. *Phys. Chem. Chem. Phys.* **2008**, *10*, 5388–5393. [CrossRef] [PubMed]
7. Whitfield, J.D.; Biamonte, J.; Aspuru-Guzik, A. Simulation of electronic structure Hamiltonians using quantum computers. *Mol. Phys.* **2011**, *109*, 735–750. [CrossRef]
8. Daskin, A.; Kais, S. Direct application of the phase estimation algorithm to find the eigenvalues of the hamiltonians. *Chem. Phys.* **2018**, *514*, 87–94. [CrossRef]
9. O'Malley, P.J.; Babbush, R.; Kivlichan, I.D.; Romero, J.; McClean, J.R.; Barends, R.; Kelly, J.; Roushan, P.; Tranter, A.; Ding, N.; et al. Scalable quantum simulation of molecular energies. *Phys. Rev.* **2016**, *6*, 031007. [CrossRef]
10. 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. [CrossRef]
11. 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. [CrossRef]
12. Hu, Z.; Xia, R.; Kais, S. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. *arXiv* **2019**, arXiv:1904.00910.
13. Smart, S.E.; Mazziotti, D.A. Efficient two-electron ansatz for benchmarking quantum chemistry on a quantum computer. *Phys. Rev. Res.* **2020**, *2*, 023048. [CrossRef]
14. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195. [CrossRef] [PubMed]
15. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef]
16. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.
17. Dallaire-Demers, P.L.; Killoran, N. Quantum generative adversarial networks. *arXiv* **2018**, arXiv:1804.08641.
18. Lloyd, S.; Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **2018**, *121*, 040502. [CrossRef]
19. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309. [CrossRef]
20. Romero, J.; Olson, J.P.; Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2017**, *2*, 045001. [CrossRef]
21. Xia, R.; Kais, S. Quantum Machine Learning for Electronic Structure Calculations. *Nat. Comuun.* **2018**, *9*, 1–6. [CrossRef]

22. Schuld, M.; Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef] [PubMed]
23. Cao, Y.; Guerreschi, G.G.; Aspuru-Guzik, A. Quantum Neuron: an elementary building block for machine learning on quantum computers. *arXiv* **2017**, arXiv:1711.11240.
24. Sim, S.; Johnson, P.D.; Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quantum Technol.* **2019**, *2*, 1900070. [CrossRef]
25. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
26. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef]
27. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
28. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. The expressive power of parameterized quantum circuits. *arXiv* **2018**, arXiv:1810.11922.
29. Romero, J.; Aspuru-Guzik, A. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *arXiv* **2019**, arXiv:1901.00848.
30. Levine, Y.; Sharir, O.; Cohen, N.; Shashua, A. Bridging Many-Body Quantum Physics and Deep Learning via Tensor Networks. *arXiv* **2018**, arXiv:1803.09780.
31. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
32. Huggins, W.J.; Patil, P.; Mitchell, B.; Whaley, K.B.; Stoudenmire, M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **2019**, *4*, 024001. [CrossRef]
33. Nielsen, M.A.; Chuang, I. Quantum computation and quantum information. *Phys. Today* **2002**, *54*, 60. [CrossRef]
34. Mari, A.; Bromley, T.R.; Izaac, J.; Schuld, M.; Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *arXiv* **2019**, arXiv:1912.08278.
35. Stoudenmire, E.; Schwab, D.J. Supervised learning with tensor networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 4799–4807.
36. Fradkin, E. Jordan-Wigner transformation for quantum-spin systems in two dimensions and fractional statistics. *Phys. Rev. Lett.* **1989**, *63*, 322. [CrossRef]
37. Roos, B.O.; Taylor, P.R.; Sigbahn, P.E. A complete active space SCF method (CASSCF) using a density matrix formulated super-CI approach. *Chem. Phys.* **1980**, *48*, 157–173. [CrossRef]
38. Romero, J.; Babbush, R.; McClean, J.R.; Hempel, C.; Love, P.J.; Aspuru-Guzik, A. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Sci. Technol.* **2018**, *4*, 014008. [CrossRef]
39. Steudtner, M.; Wehner, S. Lowering qubit requirements for quantum simulations of fermionic systems. *arXiv* **2017**, arXiv:1712.07067.
40. Sun, Q.; Berkelbach, T.C.; Blunt, N.S.; Booth, G.H.; Guo, S.; Li, Z.; Liu, J.; McClain, J.D.; Sayfutyarova, E.R.; Sharma, S.; et al. PySCF: The Python-Based Simulations of Chemistry Framework. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2018**, *8*, e1340. [CrossRef]
41. McClean, J.; Rubin, N.; Sung, K.; Kivlichan, I.D.; Bonet-Monroig, X.; Cao, Y.; Dai, C.; Fried, E.S.; Gidney, C.; Gimby, B.; et al. OpenFermion: The electronic structure package for quantum computers. *Quantum Sci. Technol.* **2020**, *5*, 034014. [CrossRef]
42. Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.J.; Carballo-Franquis, J.; Chen, A.; Chen, C.-F.; et al. Qiskit: An Open-source Framework for Quantum Computing. 2019. Available online: https://zenodo.org/record/2562111#.XyDuIecRVPY (accessed on 29 July 2020).
43. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: New York, NY, USA, 2006; Volume 10, p. 978.