Collective optimization for variational quantum eigensolvers

Dan-Bo Zhang[1,*] and Tao Yin[2,†]

[1]*Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, GPETR Center for Quantum Precision Measurement, SPTE, and Frontier Research Institute for Physics, South China Normal University, Guangzhou 510006, China*

[2]*Yuntao Quantum Technologies, Shenzhen 518000, China*

(Received 19 December 2019; accepted 14 February 2020; published 10 March 2020)

A variational quantum eigensolver (VQE) optimizes parametrized eigenstates of a Hamiltonian on a quantum processor by updating parameters with a classical computer. Such a hybrid quantum-classical optimization serves as a practical way to leverage classical algorithms in exploiting the power of near-term quantum devices. Here, aiming to solve a group of Hamiltonians more efficiently, we develop an extension of the conventional VQE. A snake algorithm is incorporated to couple optimizing processes for VQEs of different Hamiltonians by gradient descent. Such a so-called collective VQE (CVQE) is applied to simulate molecules with varied bond lengths for demonstration. Numeral simulations show that the CVQE exhibits clear collective behavior in the optimization process of updating parameters. Remarkably, the CVQE tends to avoid a single VQE task to be trapped in the local minimum. The collective optimization utilizes intrinsic relations between related tasks and may inspire advanced hybrid quantum-classical algorithms for solving practical problems with current quantum technologies.

DOI: 10.1103/PhysRevA.101.032311

I. INTRODUCTION

Quantum computing promises to solve some outstanding problems with quantum advantages [1–5] and is influencing a broad range of computationally intensive areas [1,6–9]. A variational approach for quantum computing sets parameters in a quantum circuit and those parameters can be learned through hybrid quantum-classical optimization [10–30]. Such an approach is well suited for the near-term quantum processor and has been intensively studied in recent years [31]. Among those studies, the variational quantum eigensolver (VQE) aims to solve eigenvalues and eigenstates for quantum systems [10,12,17,18,32]. The power of representing an exponentially large wave function on quantum processors and effective hybrid quantum-classical optimization of the VQE enhance the ability to solve hard quantum problems.

In many practical problems, a group of related Hamiltonians needs to be solved, e.g., molecule electronic Hamiltonians at different bond lengths or bond angles and quantum many-body systems with different interacting strengths. The conventional variational quantum eigensolver will solve such a group of Hamiltonians one by one, independently. However, it is straightforward to notice that those tasks are similar and related to each other. For instance, optimal variational parameters will be very close if their corresponding Hamiltonians are close to each other. Exploiting intrinsic relations among those related tasks may lead to more efficient optimization algorithms that employ quantum resources better. This is also related to meta learning that draws on prior experience for new tasks in machine learning [33,34] and its applications in quantum neural networks [35].

In this paper we propose a hybrid quantum-classical algorithm that can provide collective optimization for the VQE to solve a group of related Hamiltonians more efficiently. It evaluates gradients on a quantum processor and updates variational parameters on a classical computer. Remarkably, the process of updating parameters generalizes traditional gradient descent into a collective version, which updates variational parameters of different Hamiltonians simultaneously. This is achieved by a snake algorithm [36,37], originally developed in computer vision [36], which enforces a smooth condition on variational parameters of different Hamiltonians. We call this a collective VQE (CVQE). As a demonstration we use the CVQE to solve ground-state energies for several molecules at different bond lengths. The advantages of collective optimization are investigated and shown through the flow of variational parameters. Remarkably, the snake algorithm gives rise to collective motion of parameters of different tasks that can avoid being trapped in local minimums. Our job thus introduces a different dimension for developing global optimizers for variational methods.

The paper is organized as follows. In Sec. II we review the variational quantum eigensolver and then propose a CVQE using the snake algorithm. In Sec. III we present results of several representative molecules using the CVQE. In Sec. IV we investigate the snake algorithm as a global optimizer. In Sec. V we provide further discussion and a brief summary.

II. OPTIMIZATION FOR VARIATIONAL QUANTUM EIGENSOLVERS

Solving eigenvalues and eigenstates for a given Hamiltonian is a fundamental task. Quantum computers provide an

[*] dbzhang@m.scnu.edu.cn

[†] tao.yin@artiste-qb.net

avenue for effectively solving eigenstate problems of quantum systems. Different quantum algorithms have been developed for tracking this hard problem, such as quantum phase estimation [38], a variational quantum eigensolver [10,12,24], imaginary-time quantum evolution [24,39], and simulating the resonance transition of molecules on quantum processors [40,41]. The VQE approach uses a parametrized quantum circuit to prepare a wave function. The parameters are obtained by optimizing the energy with the hybrid quantum-classical algorithm.

To solve quantum systems on a quantum computer, it is necessary to first map the original Hamiltonian into a qubit (spin-$\frac{1}{2}$) Hamiltonian. For electronic systems, a nonlocal transformation such as Jordan-Wigner transformation [42] or Bravyi-Kitaev transformation [43] is required to transform fermionic operators into Pauli operators. For a quantum system of interest, e.g., molecules, the resulting qubit Hamiltonian typically has many terms,

$$H = \sum_i c_i H_i, \tag{1}$$

where H_i can be written as a tensor product of Pauli matrices $H_i = \otimes_k \sigma_k^{\alpha_k}$. Here $\alpha_k = x, y, z$ and k is the index of qubits. We now discuss how to solve a single Hamiltonian and a group of related Hamiltonians, respectively.

A. Optimization by gradient descent

To find the eigenstate for a single H, one can use an *Ansatz* $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\psi_0\rangle$ to represent a candidate ground state. Here $|\psi_0\rangle$ usually is taken as a good classical approximation for the ground state of H. In quantum chemistry, for instance, $|\psi_0\rangle$ can be chosen as a Hartree-Fock state. In addition, $U(\boldsymbol{\theta})$ is an unitary operator parametrized with $\boldsymbol{\theta}$, which can take quantum correlation into consideration. As a variational method, the essential task is to find parameters $\boldsymbol{\theta}_0$ that minimize the energy $\mathcal{E}(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$. The optimization is a hybrid quantum-classical one: The quantum processor runs the quantum circuit and performs measurements to evaluate $\mathcal{E}(\boldsymbol{\theta})$; the classical computer updates parameters $\boldsymbol{\theta}$ according to data received from the quantum processor. To obtain a quantum average of H, one can perform measurements for each term H_i, as it is a tensor product of Pauli matrices and thus corresponds to a joint measurement on multiple qubits. Measurements of all terms then are added,

$$\mathcal{E}(\boldsymbol{\theta}) = \sum_i c_i \langle\psi(\boldsymbol{\theta})|H_i|\psi(\boldsymbol{\theta})\rangle. \tag{2}$$

Optimization methods for updating parameters $\boldsymbol{\theta}$ in general can be categorized as gradient-free [18,22], such as the Nelder-Mead method and gradient descent [17,26,32]. Gradient descent methods update parameters using information of gradients. On a quantum processor, calculating gradient with respect to a target cost function [here is $\mathcal{E}(\boldsymbol{\theta})$] can be obtained with the same quantum circuit, using the shift rule [44,45] or numeral differential. Then parameters $\boldsymbol{\theta}$ are updated with gradient descent as

$$\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \eta \frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}^{t-1}), \tag{3}$$

where η is the learning rate or the step size.

B. Collective optimization

In the above, the variational quantum eigensolver solves the eigenvalue problem for a single Hamiltonian. In practice, there may be a group of Hamiltonians to be solved. For instance, what is needed in quantum chemistry usually is an energy potential surface, corresponding to ground-state energies for a molecule at different bond lengths or bond angles. Of course, one can use the VQE to solve Hamiltonians one by one. However, this does not exploit relations between Hamiltonians. Here we develop a more efficient method that can collectively optimize all variational gate parameters for different Hamiltonians at each step of updating parameters.

The motivation behind collective optimization can be presented as follows. Consider calculating the energy potential surface for a molecule at varied bond lengths, which is a standard quantum-chemistry problem. Two Hamiltonians should be close to each other if their underlying molecules are the same and bond lengths vary a little. In such a case, the same *Ansatz* can be applied, and it is expected that optimized parameters of the wave-function *Ansatz* should be very close to each other. With $\lambda \in [\lambda_0, \lambda_T]$ as the bond length and $\boldsymbol{\theta}_0(\lambda)$ as the optimized parameter for the Hamiltonian $H(\lambda)$, then $\boldsymbol{\theta}_0(\lambda) \sim \lambda$ should form a continuous curve in the space of $\boldsymbol{\theta}$ and λ, which we may call an enlarged parameter space. We expect that the optimization of the VQE for one Hamiltonian can help optimize VQEs for other Hamiltonians with nearby system parameters λ. We use gradient descent for the optimization. Instead of updating a single point in the parameter space, the collective optimization updates a sequence of points in the enlarged parameter space, with each point corresponding to a Hamiltonian. At the continuous limit, this is an optimization of a string.

Now let us elaborate on a concrete algorithm. To incorporate a snake algorithm, the cost function should consider the energy of the snake itself, which can be written as follows [36,37]:

$$L(\boldsymbol{\theta}(\lambda)) = \int_{\lambda_0}^{\lambda_T} [\mathcal{L}(\boldsymbol{\theta}(\lambda)) + \mathcal{E}(\boldsymbol{\theta}(\lambda))]. \tag{4}$$

Here $\mathcal{E}(\boldsymbol{\theta}(\lambda)) = \langle\psi(\boldsymbol{\theta}(\lambda))|H_\lambda|\psi(\boldsymbol{\theta}(\lambda))\rangle$ is the local potential the snake experiences. The internal energy of the snake is

$$\mathcal{L}(\boldsymbol{\theta}(\lambda)) = \alpha \left|\frac{\partial \boldsymbol{\theta}(\lambda)}{\partial \lambda}\right|^2 + \beta \left|\frac{\partial^2 \boldsymbol{\theta}(\lambda)}{\partial^2 \lambda}\right|^2. \tag{5}$$

The first term on the right-hand side of Eq. (5) refers to the energy depending on the length of the snake and thus counts for the stretchability, while the second term refers to the energy depending on how curved the snake is and thus counts for the bendability. Two features of the snake are determined by the hyperparameters α and β, respectively. While the α term alone can be related to a collective optimization, the β term will put a very curved snake on high energy and thus punish it, which can make the optimization more efficient.

Solving the snake can be achieved by minimizing Eq. (4), which can be converted to the task of solving a differential equation [see Eq. (A1)]. For this we discretize the snake as a sequence of parameters at different bond lengths $\mathbf{r}_i = (\boldsymbol{\theta}_i(\lambda_1), \boldsymbol{\theta}_i(\lambda_2), \ldots, \boldsymbol{\theta}_i(\lambda_M))$, where i is the ith component for each $\boldsymbol{\theta}_i(\lambda_m)$. Then the discrete snake can be solved iteratively as

$$\mathbf{r}_i^t = (\eta \boldsymbol{A} + \boldsymbol{I})^{-1}\left(\mathbf{r}_i^{t-1} - \eta \frac{dE(\mathbf{r}^{t-1})}{d\mathbf{r}_i}\right), \tag{6}$$

where $E(\mathbf{r}) = \sum_{m=1}^{M} \mathcal{E}(\boldsymbol{\theta}_{\lambda_m})$ and $\boldsymbol{A}$ is a pentadiagonal banded matrix with nonzero elements depending on α and β (see detailed expressions in Appendix A). Compared with Eq. (3), Eq. (6) can be viewed as a collective gradient descent, as the latter is reduced to the former at $\alpha = \beta = 0$.

Still, there is an issue for incorporating the snake algorithm into optimizing variational quantum eigensolver. The equilibrium condition (A2) [or Eq. (A1)] is actually not the original one $\frac{d\mathcal{E}(\boldsymbol{\theta}(\lambda_i))}{d\boldsymbol{\theta}(\lambda_i)} = 0$, as there are interactions between neighbors $\boldsymbol{\theta}(\lambda_i)$. As a result, optimization with a gradient flow using Eq. (6) may not give the required optimal results. In practice, nevertheless, this issue may be largely ignored, as explained in the following. For a neighbor λ_i, it is expected that $\boldsymbol{\theta}(\lambda_{i+1}) + \boldsymbol{\theta}(\lambda_{i-1}) \approx 2\boldsymbol{\theta}(\lambda_i)$ and $\boldsymbol{\theta}(\lambda_{i+2}) + \boldsymbol{\theta}(\lambda_{i-2}) \approx 2\boldsymbol{\theta}(\lambda_i)$ once the optimization is good enough and M is large enough. It can be checked that the first term of Eq. (A2) can be approximated as zero, which is consistent with the equilibrium condition for the VQE, namely, by omitting the first term.

In practice, one can introduce a decaying matrix $\boldsymbol{A}(t) = \boldsymbol{A}_0 \exp(-t\Gamma)$ in the optimization process. At the large-t limit, this become the gradient descent of Eq. (3). An analog may be made with the annealing methods that are widely applied for optimization. Internal forces play the role of temperature. Initially, internal forces are large and parameters for different Hamiltonians flow in the space collectively. With decaying internal forces, flows of different parameters become more independent. This may lead us to believe that the snake algorithm may help avoid the optimization to be trapped in a local minimum for a single VQE, which will be investigated in Sec. IV.

III. APPLICATION OF THE CVQE FOR MOLECULES

In this section we apply the CVQE for several representative molecules, including molecular hydrogen, lithium hydride, and a helium hydride cation, and present their results. The numerical simulations are performed by using the Huawei HiQ simulator framework [46]. It is shown that ground-state energies are obtained with great accuracy compared with results using a variational quantum eigensolver for the Hamiltonian at each bond length alone. Remarkably, variational parameters for ground states of Hamiltonians at different bond lengths collectively flow to optimal values. We present the main results and details of the calculation of Hamiltonians for all molecules at different bond lengths as well as their wave-function *Ansatz* in Appendix B.

A. Molecular hydrogen

For H_2, we consider an effective qubit Hamiltonian involving two qubits, following Ref. [18]. The unitary coupled cluster (UCC) *Ansatz* is used; the unitary operator

$$U(\theta) = \exp\left(-i\theta\sigma_0^x\sigma_1^y\right)$$

performing on the Hartree-Fock reference state is $|01\rangle$. We choose 54 points uniformly from bond lengths ranging from 0.25 to 2.85 a.u. Effective Hamiltonians corresponding to those bond lengths are obtained with the open source software OpenFermion [47]. Variational parameters are randomly initialized. We set $\alpha = 0.1$, $\beta = 3$, and $\eta = 0.5$ in Eq. (6) (note that $\boldsymbol{A}$ depends on α and β). Ground-state energies at

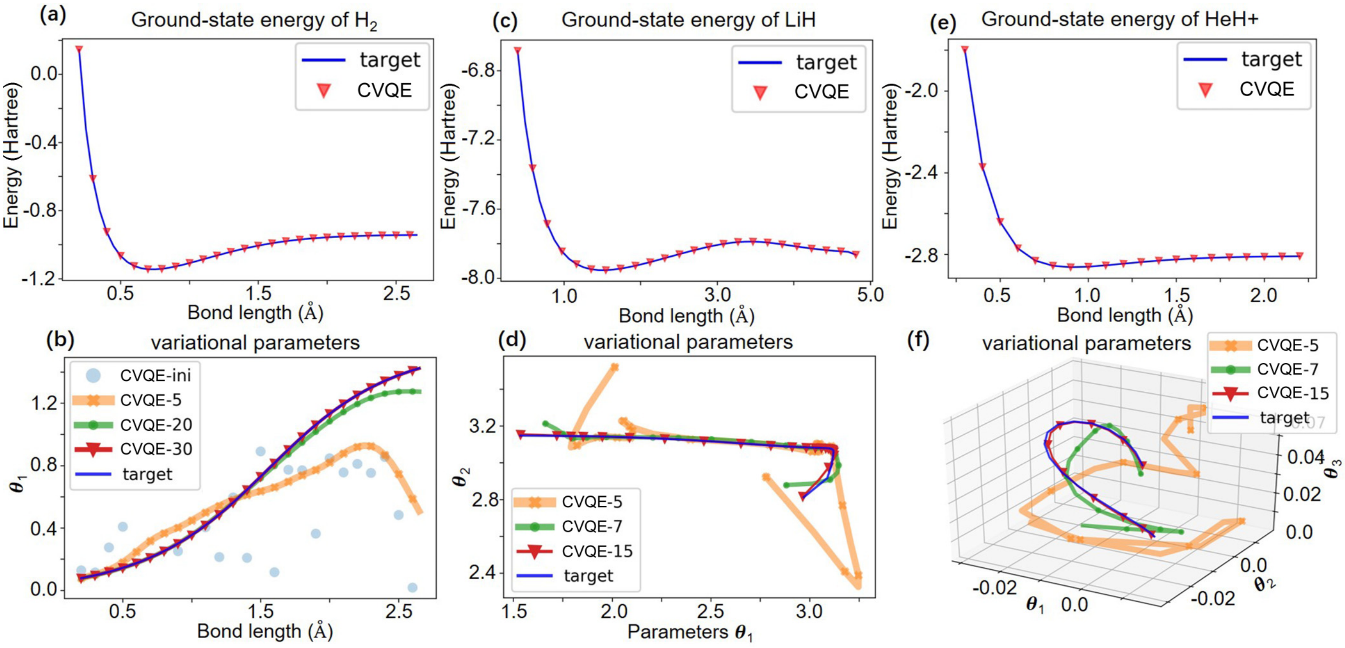

FIG. 1. Collective optimization with the snake algorithm for molecules at different bond lengths. (a), (c), and (e) Target energies and optimization results for the CVQE algorithm and (b), (d), and (f) optimization process of variational parameters for the molecules (a) and (b) H_2, (c) and (d) LiH, and (e) and (f) HeH^+, with one, two, and three variational parameters, respectively.

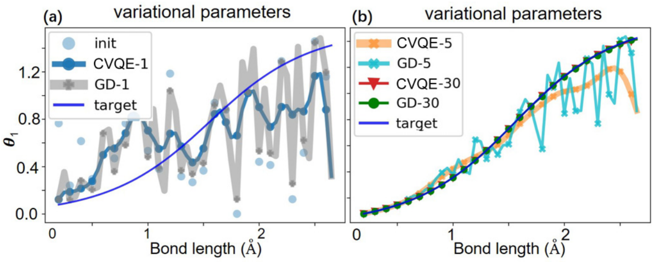

FIG. 2. Comparison between the CVQE and GD, step by step. (a) Starting from circles, the CVQE connects neighboring tasks after the first iteration. (b) The CVQE forms a smooth curve after five iterations, while GD evolves independently. Both methods converge eventually to the target values.

different bond lengths fit perfectly with ideal results. Remarkably, variational parameters for different bond lengths, while initialized randomly, quickly form a smooth curve and evolve to the optimal target values, as shown in Fig. 1. This can be understood as a collective optimization process that exploits intricate relations between VQE tasks for Hamiltonians with different bond lengths.

B. Flow of parameters

To show the collectivity futures of the snake algorithm, in Fig. 2 we compare optimization processes between the CVQE and normal VQE with the gradient descent (GD) method, step by step. In the preceding section we considered the CVQE method for molecular hydrogen H_2 with $\alpha = 0.1$, $\beta = 3$, and $\eta = 0.5$. The method is reduced to the gradient descent result with the parameters $\alpha = \beta = 0$. As is clearly shown, starting from the same randomly initialized parameters, the CVQE updates as a correlated smooth curve while the GD updates independently for each bond length. Though both the CVQE and GD converged to the target values (as expected), the CVQE will obtain the overall system properties more efficiently, which is valuable in the noisy intermediate-scale quantum era [31].

C. Lithium hydride

For LiH, the STO-6G basis is used to construct the electronic Hamiltonian, which is mapped into a qubit Hamiltonian with Bravyi-Kitaev (BK) transformation. Following Ref. [18], three orbitals are chosen so that the final qubit Hamiltonian evolves three qubits. The UCC operator $U(\theta_1, \theta_2) = \exp(-i\theta_1\sigma_0^x\sigma_1^y)\exp(-i\theta_2\sigma_0^x\sigma_2^y)$ performs on an initial state $|111\rangle$. The operator can be taken as two UCCs and each can be decomposed as in Eq. (6). Effective Hamiltonians are calculated with OpenFermion from 50 bond lengths, uniformly chosen from 0.3 to 5.0 a.u. Variational parameters are randomly initialized. We set $\alpha = 0.1$, $\beta = 3$, and $\eta = 0.2$ in Eq. (6). It can be seen in Fig. 1 that the potential surface fits well with ideal results. The evolution of variational parameters turns out to be rather impressive. Unlike the case of molecular hydrogen, there are two parameters for each VQE and thus all points form a curve in the parameter space. The initial curve is random (due to random initialization) and is far away from the target. Nevertheless, the curve flows to the target curve by both shifting and changing its shape. Such a collective optimization process is strikingly reminiscent of the behavior of a crawling snake.

D. Helium hydride cation

We now turn to consider the helium hydride cation, which is a more complicated molecule carrying one positive charge. Under the STO-3G basis, four qubits are required to describe the Hamiltonian [16]. To capture the essential quantum correlation, the UCC *Ansatz* should include a two-particle scattering component [16]. The unitary operator can be written as $U(\theta_1, \theta_2, \theta_3) = \exp(-i\theta_3\sigma_0^x\sigma_1^x\sigma_2^x\sigma_3^y)$ $\exp(-i\theta_2\sigma_1^x\sigma_3^y)\exp(-i\theta_1\sigma_0^x\sigma_2^y)$. Effective Hamiltonians are calculated with OpenFermion from 30 bond lengths, ranging from 0.25 to 2.5 a.u. Hyperparameters for the snake are set as $\alpha = 0.1$, $\beta = 3$, and $\eta = 0.2$ in Eq. (6). As there are three variational parameters, their evolution can be visualized as a crawling snake in a three-dimensional space. Although initialized randomly, the snake becomes more smooth and moves to the target position. This again demonstrates a collective optimization process for the snake algorithm.

IV. NONCONVEX OPTIMIZATION OF THE CVQE

In the above, we have applied the CVQE for solving ground-state energies of several molecules at different bond lengths. The process of optimization shows that parameters for different bond lengths evolve more smoothly, a remarkable feature of the snake algorithm for collective optimization. In this section we further reveal that the snake algorithm trends for a global optimization, avoiding being trapped at local minimums. The key point is to randomly set the parameters of different Hamiltonians or tasks that some parameters fall in traps of global minimums. By collective optimization, those lucky parameters tend to pull other parameters out of local minimums. The strategy is used for optimization for both classical problems of nonconvex functions and variational quantum eigensolvers.

A. Snake algorithm for the nonconvex function

We first use a toy example to illustrate how a collective optimization with the snake algorithm can prevent an optimization process from being trapped in local minimums. We consider minimizing the Styblinski-Tang (ST) function [48], a nonconvex function used to benchmark optimization algorithms, defined as $f(x) = \frac{1}{2}\sum_{i=1}^{N} x_i^4 - 16x_i^2 + t_i x_i$. To illustrate the mechanism of the snake algorithm for nonconvex optimization, we take $N = 1$ and consider a group of ST functions, parametrized with t as $f(x;t) = \frac{1}{2}(x^4 - 16x^2 + tx)$, where $t \geqslant 0$. For fixed t, there are two minimums located at $\pm x_0(t)$ and a global one located at $-x_0(t)$ [assuming $x_0(t) > 0$]. However, those traps are sufficiently deep that an optimizer may be easily trapped at local minimums, especially for optimizers based on gradient descents. The snake algorithm, although using gradient descent, can avoid this issue. As seen in Fig. 3, most optimal points for different TS functions are located at global minimums. This is because all points are interconnected and can be optimized collectively. Initially, there

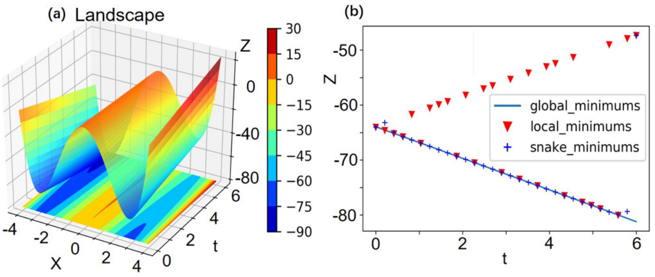

FIG. 3. Optimization for a group of Styblinski-Tang functions parametrized with t ($0 \leqslant t \leqslant 6$). Here $f(x;t) = \frac{1}{2}(x^4 - 16x^2 + tx)$. (a) The landscape for a ST function has two minimums for fixed t. (b) Optimizations with gradient descent and the snake algorithm. It is shown that local minimums are often achieved by gradient descent, while the snake algorithm can mostly achieve global minimums.

are some points located at traps of global minimums with random initialization. Then those points will pull other points out of traps of local minimums, as can be seen in Fig. 4(b). Such a mechanism can explain why the snake algorithm can be used as an optimizer for a nonconvex function.

B. Nonconvex optimization for the VQE

For the variational quantum eigensolver, an expectation of the Hamiltonian with regard to the variational wave-function *Ansatz* is in general a nonconvex function of variational parameters. For illustration, we still consider the hydrogen molecule with the same Hamiltonian as in Eq. (B1), but the wave-function *Ansatz* is changed to

$$U(\theta_1,\theta_2) = \exp\left[-i\theta_2\left(a\sigma_0^x + b\sigma_1^x\right)\right]\exp\left(-i\theta_1\sigma_0^x\sigma_1^y\right).$$

Here a and b are fixed and we set $a = 2$ and $b = 1.5$, for instance. Compared to the original unitary coupled cluster *Ansatz*, there is an extra term $\exp[-i\theta_2(a\sigma_0^x + b\sigma_1^x)]$. As can be seen in Fig. 5(b), the landscape has several different local minimums. The global one is located at the center, corresponding to $\theta_2 = 0$. This is expected as the case of $\theta_2 = 0$ corresponds to the wave function respecting particle conservation, which is required for the system of hydrogen molecule. A simple gradient descent as in Eq. (3) may lead to local minimums, once initially parameters of (θ_1, θ_2) are in traps of local minimums [Fig. 5(a)]. In fact, θ_2 corresponding to those local minimums are far from zero, as can be seen in Fig. 5(c). However, the snake algorithm can perfectly overcome the issue of local minimums for optimizing the VQE. During the optimization process, the θ_1 at different bond lengths evolve collectively. The curve connecting different θ_2 becomes more smooth when approaching the target. Meanwhile, all θ_1 shrink to zero. Those present nice features for nonconvex optimizations that are often encountered in the VQE for quantum-chemistry problems.

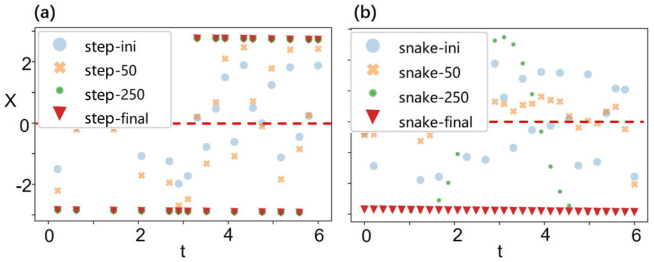

FIG. 4. Optimization processes for a nonconvex function. (a) Optimization by gradient descent. The flow of x at fixed t depends on the sign of the initial value x_0; if x_0 is positive then the optimization goes to the local minimum. (b) Optimization by the snake algorithm. It can be seen that almost all x flow to global minimums collectively, even if they are initialized randomly as positive and negative.

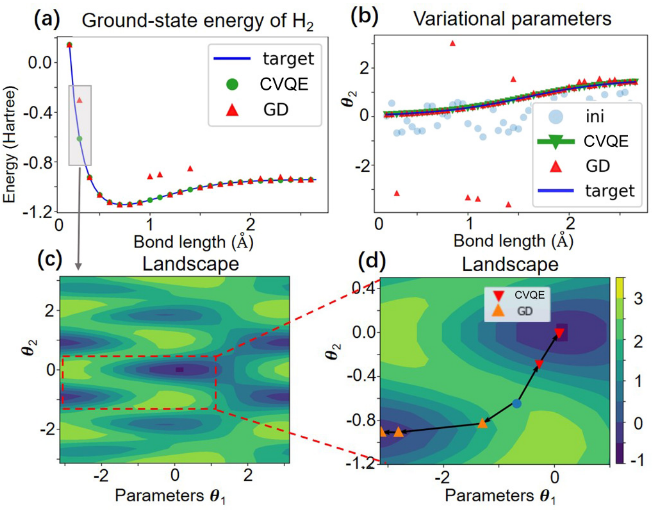

FIG. 5. Nonconvex optimization for hydrogen molecule with the CVQE algorithm. (a) Optimization results using both the CVQE algorithm (marked as CVQE) and gradient descent (marked as GD). (b) Evolution of parameters θ_1 for the optimization process using the CVQE algorithm (green line) and gradient descent (red triangles). (c) and (d) The landscape for the VQE of the hydrogen molecule has several different minimums. The random initial point flows to the global minimum in the CVQE method and to a local minimum in gradient descent.

V. DISCUSSION AND SUMMARY

Optimization is a key component for variational quantum eigensolvers. Here we have incorporated the snake algorithm for optimization of a group of VQEs to find ground-state energies for a molecule at different bond lengths. As the first step for collective optimization for quantum-chemistry and many-body problems, it is expected that the CVQE can be tested on more general wave-function *Ansatz*. We have applied the unitary coupled cluster *Ansatz* for a quantum-chemistry problem and considered only a small number of variational parameters. For many quantum-chemistry and many-body problems, more variational parameters are required and also other wave-function *Ansatz* may be more suitable [32]. The snake algorithm can be studied for such a high-dimensional optimization problem. It is expected that the snake algorithm can help us escape local minimums that often appear in a high-dimensional landscape. Another direction is to generalize this one-dimensional snake to a higher-dimensional one, and thus is applicable for obtaining an energy potential surface of more complicated molecules. We remark that the CVQE can be implemented with multiple quantum devices in parallel.

In summary, we have incorporated the snake algorithm to optimize variational quantum eigensolvers for a group of Hamiltonians. The CVQE has been used to solve for ground states of molecules at different bond lengths simultaneously, which is enhanced by the collective optimization. Remarkably, we have demonstrated that the snake algorithm is a global optimizer, as the collective motion of variational parameters for different tasks can help pull parameters out of traps of local minimums.

ACKNOWLEDGMENTS

The authors thank Peng Cheng Laboratory, where this work was finalized. T.Y. also thanks Dr. Lei Wang for insightful discussion. This work was supported by the Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B030330001), the National Key Research and Development Program of China (Grant No. 2016YFA0301800), the National Natural Science Foundation of China (Grants No. 91636218 and No. U1801661), and the Key Project of Science and Technology of Guangzhou (Grant No. 201804020055).

APPENDIX A: COLLECTIVE GRADIENT DESCENT

In this Appendix we give details of the derivation of Eq. (6). The snake is determined from the least-action principle. This is achieved by minimizing $L(\boldsymbol{\theta}(\lambda))$. By the Euler-Lagrange equation, this leads to a fourth-order differential equation

$$\alpha\frac{\partial^2\boldsymbol{\theta}(\lambda)}{\partial^2\lambda} + \beta\frac{\partial^4\boldsymbol{\theta}(\lambda)}{\partial^4\lambda} + \frac{dE}{d\boldsymbol{\theta}(\lambda)} = 0. \tag{A1}$$

Here $E = \int_{\lambda_0}^{\lambda_T}\mathcal{E}(\boldsymbol{\theta}(\lambda))$. The last term in Eq. (A1) should be evaluated on a quantum processor, which makes Eq. (A1) rather special, and a solution with a hybrid quantum-classical algorithm is expected.

Equation (A1) should be solved numerically as a discrete version. Here M different parameters are chosen uniformly from $[\lambda_0, \lambda_T]$ as $\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$, and $\lambda_T - \lambda_0 = M\delta$. Using a finite difference, the second and fourth orders of differentials turn out to be Eq. (A1),

$$[\boldsymbol{\theta}(\lambda_{i+1}) - 2\boldsymbol{\theta}(\lambda_i) + \boldsymbol{\theta}(\lambda_{i-1})]/\delta^2,$$
$$[\boldsymbol{\theta}(\lambda_{i-2}) - 4\boldsymbol{\theta}(\lambda_{i-1}) + 6\boldsymbol{\theta}(\lambda_i) - 4\boldsymbol{\theta}(\lambda_{i+1}) + \boldsymbol{\theta}(\lambda_{i+2})]/\delta^4,$$

respectively. Then we have

$$\boldsymbol{A}\mathbf{r}_i + \frac{dE(\mathbf{r})}{d\mathbf{r}_i} = 0, \quad i = 1, 2, \ldots, N. \tag{A2}$$

For convenience we also introduce $\mathbf{r}_i = (\boldsymbol{\theta}_i(\lambda_1), \boldsymbol{\theta}_i(\lambda_2), \ldots, \boldsymbol{\theta}_i(\lambda_M))$ and define $E(\mathbf{r}) = \sum_i \mathcal{E}(\boldsymbol{\theta}_{\lambda_i})$. In addition, $\boldsymbol{A}$ is a pentadiagonal banded matrix with nonzero elements (under the periodic condition), $\boldsymbol{A}_{i-2,i} = \boldsymbol{A}_{i,i-2} = \beta$, $\boldsymbol{A}_{i-1,i} = \boldsymbol{A}_{i,i-1} = -\alpha - 4\beta$, and $\boldsymbol{A}_{i,i} = 2\alpha + 6\beta$, where δ^2 and δ^4 are absorbed accordingly. Following Ref. [36], Eq. (A2) can be solved by introducing gradient flow (with an explicit Euler step, so it uses $\boldsymbol{A}\mathbf{r}_i^t$ instead of $\boldsymbol{A}\mathbf{r}_i^{t-1}$)

$$-\frac{\mathbf{r}_i^t - \mathbf{r}_i^{t-1}}{\eta} = \boldsymbol{A}\mathbf{r}_i^t + \frac{dE(\mathbf{r}^{t-1})}{d\mathbf{r}_i}, \tag{A3}$$

which leads to Eq. (6).

APPENDIX B: HAMILTONIANS AND UNITARY CLUSTER *ANSATZ*

Solving for eigenvalues of electronic structures of molecules is one central problem for quantum chemistry. The ground-state energy is especially important as it largely determines the chemical properties of molecules. The electronic Hamiltonian for a molecule consists of nuclear charges and electrons with Coulomb interactions. With the Born-Oppenheimer approximation, the locations of nuclei are fixed. The electronic Hamiltonian is usually reformulated in the second quantized formulation, with a basis of N molecular orbitals that are a linear combination of atomic orbitals. This can reduce the infinite-dimensional space of the original real space into a finite Hilbert space. Solving for eigenvalues and eigenstates can be done in this subspace. The dimensionality N can be adjusted for the sake of precision demanded.

In the second quantization, the Hilbert space grows exponentially with the number of orbitals N. It is important to only consider orbitals that contribute significantly to the low-energy state. In practice, only active orbitals are considered, and inactive ones, such as occupied orbitals very close to the nucleus or outside empty orbitals, are ignored. This leads to an effective electronic Hamiltonian that allows for feasible solutions.

The electronic Hamiltonian is fermionic and still cannot be solved on a quantum processor. To map fermionic operators into qubit operators, one can refer to the Jordan-Wigner transformation or Bravyi-Kitaev transformation. Those transformations are nonlocal and may introduce a tensor product of a string of Pauli matrices in the qubit Hamiltonian.

We consider three kinds of molecules: Molecular hydrogen, lithium hydride, and a helium hydride cation. Their qubit Hamiltonians with varying bond lengths are calculated with OpenFermion [47], following setups in Ref. [18] for hydrogen and lithium hydride and Ref. [16] for helium hydride cation.

For H_2 the STO-3G minimal basis adopted and the final effective qubit Hamiltonian involves two qubits, which can be written as

$$\begin{aligned} H_{\mathrm{H}_2}(\lambda) = {} & c_0(\lambda)\mathcal{I} + c_1(\lambda)\sigma_0^z + c_2(\lambda)\sigma_1^z + c_3(\lambda)\sigma_0^z\sigma_1^z \\ & + c_4(\lambda)\sigma_0^x\sigma_1^x + c_5(\lambda)\sigma_0^y\sigma_1^y. \end{aligned} \tag{B1}$$

Here the coefficients $c_i(\lambda)$ depend on the bond length λ and their values can be found in the code. The Hartree-Fock reference state is $|01\rangle$.

For LiH, STO-6G basis is used to construct the electronic Hamiltonian, which is mapped into a qubit Hamiltonian with a BK transformation. Following Ref. [18], three orbitals are chosen so that the final qubit Hamiltonian evolves three qubits,

$$\begin{aligned} H_{\mathrm{LiH}}(\lambda) &\\ = c_0(\lambda)\mathcal{I} + c_1(\lambda)\sigma_0^z + c_2(\lambda)\sigma_1^z + c_3(\lambda)\sigma_2^z + c_4(\lambda)\sigma_0^z\sigma_1^z &\\ + c_5(\lambda)\sigma_0^z\sigma_2^z + c_6(\lambda)\sigma_1^z\sigma_2^z + c_7(\lambda)\sigma_0^x\sigma_1^x + c_8(\lambda)\sigma_0^x\sigma_2^x &\\ + c_9(\lambda)\sigma_1^x\sigma_2^x + c_{10}(\lambda)\sigma_0^y\sigma_1^y + c_{11}(\lambda)\sigma_0^y\sigma_2^y + c_{12}(\lambda)\sigma_1^y\sigma_2^y. & \end{aligned} \tag{B2}$$

The reference state is $|001\rangle$.

For the above two effective qubit Hamiltonians, we adopt a simple unitary coupled cluster *Ansatz* [10,16,18,49,50], which can establish entanglement between different qubits and thus take quantum correlations into account. For H_2, the unitary operator is

$$U(\theta) = \exp\left(-i\theta\sigma_0^x\sigma_1^y\right)$$

and the wave-function *Ansatz* is $U(\theta)|01\rangle$. The parameter θ can characterize the degree of entanglement between the electron and the hole. For LiH, the UCC *Ansatz* is

$$U(\theta_1,\theta_2) = \exp\left(-i\theta_2\sigma_0^x\sigma_2^y\right)\exp\left(-i\theta_1\sigma_0^x\sigma_1^y\right)$$

and the wave-function *Ansatz* is $U(\theta_1,\theta_2)|111\rangle$. Here $U(\theta_1,\theta_2)$ can be decoupled as two entanglers that establish an entanglement of the zeroth and first orbitals and the zeroth and second orbitals, respectively. Two parameters θ_1 and θ_2 characterize the degrees of entanglement, correspondingly.

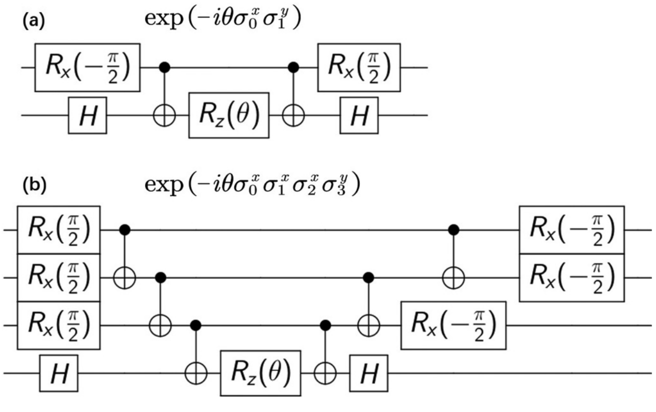

FIG. 6. Decomposition of the basic operator in a unitary coupled-cluster *Ansatz* into a set of universal quantum gates.

We also consider the helium hydride cation (HeH$^+$). Under the STO-3G basis, its qubit Hamiltonian includes two, three, and four spin interactions:

$$\begin{aligned} H_{\mathrm{HeH}^+}(\lambda) = c_0 I + \sum_{i=0}^{3} c_i Z_i + \sum_{i,j=0}^{3}(z_{ij}Z_iZ_j + x_{ij}X_iX_j + y_{ij}Y_iY_j) + \sum_{i=0}^{1} t_3(X_iZ_{i+1}X_{i+2} + Y_iZ_{i+1}Y_{i+2}) &\\ + f_0X_0X_1Y_2Y_3 + f_1Y_0Y_1X_2X_3 + f_2X_0Y_1Y_2X_3 + f_3Y_0X_1X_2Y_3 + f_4X_0Z_1X_2Z_3 + f_5Z_0X_1Z_2X_3 + f_6Y_0Z_1Y_2Z_3 + f_7Z_0Y_1Z_2Y_3. & \end{aligned} \tag{B3}$$

An UCC *Ansatz* for $H_{\mathrm{HeH}^+}(\lambda)$ should consider both the first and second excitation. Following Ref. [16], we use

$$\begin{aligned} U(\theta_1,\theta_2,\theta_3) = \exp\left(-i\theta_3\sigma_0^x\sigma_1^x\sigma_2^x\sigma_3^y\right) &\\ \times \exp\left(-i\theta_2\sigma_1^x\sigma_3^y\right)\exp\left(-i\theta_1\sigma_0^x\sigma_2^y\right). & \end{aligned} \tag{B4}$$

The wave-function *Ansatz* is $U(\theta_1,\theta_2,\theta_3)|0011\rangle$.

To implement the above *Ansatz* on quantum processors, we need to decompose the Hamiltonian evolution of the one-particle transition and two-particle transition into a set of universal quantum gates involving single-qubit rotations and a two-qubit controlled-NOT gate, as can be seen in Fig. 6 The decomposition makes the UCC operator implementable on quantum chips. Moreover, variational parameters only appear in a single-qubit rotation $R_z(\theta)$. Thus, an analytic gradient can be evaluated using the shift rule.

[1] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[2] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[4] S. Aaronson and A. Arkhipov, Theo. Comput. **9**, 143 (2013).

[5] S. Bravyi, D. Gosset, and R. König, Science **362**, 308 (2018).

[6] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. **79**, 2586 (1997).

[7] I. Buluta and F. Nori, Science **326**, 108 (2009).

[8] A. Trabesinger, Nat. Phys. **8**, 263 (2012).

[9] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).

[10] M. H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano, Sci. Rep. **4**, 3589 (2014).

[11] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, in *Quantum Technology and Optimization Problems, QTOP 2019*. Lecture Notes in Computer Science, edited by S. Feld and C. Linnhoff-Popien, Vol. 11413 (Springer, Cham, 2019).

[12] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New J. Phys. **18**, 023023 (2016).

[13] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding *et al.*, Phys. Rev. X **6**, 031007 (2016).

[14] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).

[15] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Phys. Rev. A **95**, 042308 (2017).

[16] Y. Shen, X. Zhang, S. Zhang, J.-N. Zhang, M.-H. Yung, and K. Kim, Phys. Rev. A **95**, 020501(R) (2017).

[17] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature (London) **549**, 242 (2017).

[18] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Phys. Rev. X **8**, 031022 (2018).

[19] E. R. Anschuetz, J. P. Olson, A. Aspuru-Guzik, and Y. Cao, arXiv:1808.08927.

[20] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).

[21] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, Quantum Sci. Technol. **3**, 030503 (2018).

[22] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, Nature (London) **569**, 355 (2019).

[23] T. Takeshita, N. C. Rubin, Z. Jiang, E. Lee, R. Babbush, and J. R. McClean, Phys. Rev. X **10**, 011004 (2020).

[24] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, npj Quantum Inf. **5**, 75 (2019).

[25] O. Higgott, D. Wang, and S. Brierley, Quantum **3**, 156 (2019).

[26] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, arXiv:1910.01155.

[27] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, arXiv:1909.09083.

[28] B.-X. Wang, M.-J. Tao, Q. Ai, T. Xin, N. Lambert, D. Ruan, Y.-C. Cheng, F. Nori, F.-G. Deng, and G.-L. Long, npj Quantum Inf. **4**, 52 (2018).

[29] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).

[30] N. Yoshioka, Y. O. Nakagawa, K. Mitarai, and K. Fujii, arXiv:1908.09836.

[31] J. Preskill, Quantum **2**, 79 (2018).

[32] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, Phys. Rev. Res. **1**, 023025 (2019).

[33] M. Andrychowicz, M. Denil, S. Gómez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), pp. 3981–3989.

[34] C. Finn, P. Abbeel, and S. Levine, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML17* (JMLR.org, 2017), pp. 1126–1135.

[35] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, arXiv:1907.05415.

[36] M. Kass, A. Witkin, and D. Terzopoulos, Int. J. Comput. Vision. **1**, 321 (1988).

[37] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).

[38] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[39] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan, Nat. Phys. **16**, 205 (2019).

[40] H. Wang, S. Ashhab, and F. Nori, Phys. Rev. A **85**, 062304 (2012).

[41] Z. Li, X. Liu, H. Wang, S. Ashhab, J. Cui, H. Chen, X. Peng, and J. Du, Phys. Rev. Lett. **122**, 090504 (2019).

[42] P. Jordan and E. Wigner, Z. Phys. **47**, 631 (1928).

[43] S. B. Bravyi and A. Y. Kitaev, Ann. Phys. (N.Y.) **298**, 210 (2002).

[44] J. Li, X. Yang, X. Peng, and C.-P. Sun, Phys. Rev. Lett. **118**, 150503 (2017).

[45] M. Schuld and N. Killoran, Phys. Rev. Lett. **122**, 040504 (2019).

[46] Huawei HiQ team, Huawei HiQ: A High-performance Quantum Computing Simulator and Programming Framework, http://hiq.huaweicloud.com.

[47] J. R. McClean, K. J. Sung, I. D. Kivlichan, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, P. Gokhale, T. Häner *et al.*, arXiv:1710.07629.

[48] M. A. Styblinski and T. S. Tang, Neural Netw. **3**, 467 (1990).

[49] G. K.-L. Chan, M. Kállay, and J. Gauss, J. Chem. Phys. **121**, 6110 (2004).

[50] A. G. Taube and R. J. Bartlett, Int. J. Quantum. Chem. **106**, 3393 (2006).