Learning to Learn Variational Quantum Algorithm

Rui Huang, Xiaoqing Tan, and Qingshan Xu

***Abstract*— Variational quantum algorithms (VQAs) use classical computers as the quantum outer loop optimizer and update the circuit parameters to obtain an approximate ground state. In this article, we present a meta-learning variational quantum algorithm (meta-VQA) by recurrent unit, which uses a technique called "meta-learner." Motivated by the hybrid quantum-classical algorithms, we train classical recurrent units to assist quantum computing, learning to find approximate optima in the parameter landscape. Here, aiming to reduce the sampling number more efficiently, we use the quantum stochastic gradient descent method and introduce the adaptive learning rate. Finally, we deploy on the TensorFlow Quantum processor within approximate quantum optimization for the Ising model and variational quantum eigensolver for molecular hydrogen (H_2), lithium hydride (LiH), and helium hydride cation (HeH^+). Our algorithm can be expanded to larger system sizes and problem instances, which have higher performance on near-term processors.**

***Index Terms*— Meta-learning, quantum algorithm, quantum computing, quantum information, quantum machine learning (QML).**

I. Introduction

META-LEARNING is called learning to learn [1], which aims to learn how to optimize the parameters of an algorithm and further formulate it for specific problems. Interestingly, the meta-learner model ensures that the learning can generalize by minimizing the test set error. Besides, it can also apply pretrained neural networks to a new task (perform transfer learning). Existing studies have demonstrated a range of meta-learning applications, such as gradient descent [2], model agnostic [3], and first order [4]. A prominent example of these algorithms is learning how to optimize the objective function parameters.

The research of quantum machine learning (QML) relies on quantum characteristics to accelerate classical machine learning. To some extent, compared with the classical algorithm, QML achieved exponential or quadratic speedup [5]. One key is the ability of QML to perform fast linear algebra in the state space. In quantum neural networks (QNNs), entangled witness [6], nonlinear neuron [7], and tomography [8] have been proposed. Besides, QML algorithms are proposed in both supervised and unsupervised learning, including quantum principal component analysis [9], quantum linear regression [10], and quantum hyperparameters estimation [11]. In this context, these algorithms are based on quantum random access memory [12], in which the time complexity is $O\left[\log_2(N)\right]$.

With the advent of the noisy intermediate-scale quantum (NISQ) processor [13], the quantum algorithm for quantum devices in the near term has emerged. This new research area is called the hybrid quantum-classical algorithm [14], similar to the heuristic algorithm of classical machine learning. By continuously improving the model circuit, we can measure the expectation of the objective function. One key to learning a given task is to adjust the parameters of the circuit, where iterative optimization allows us to circumvent deep circuits. The other is the variational quantum algorithm (VQA), which uses classical computers as the quantum outer loop optimizer. We note that the variational algorithm includes variational quantum eigensolver (VQE) [15], quantum approximate optimization algorithm (QAOA) [16], and variational quantum tensor networks [17]. The goal of these algorithms is to generate specific wave functions or extract nonlocal information.

In the field of VQE and QAOA, such an approach is well suited for the near-term quantum processor. A quantum algorithm based on collective optimization is designed, which solves a group of related Hamiltonian effectively [18]. A novel continuous optimization QAOA is proposed, which encodes the objective function based on the dynamics of the quantum system [19]. However, the current research mainly focuses on the gradient optimizer method without introducing the idea of deep learning to make the model smarter. Inspired by the above algorithm design, we set our sights on the field of meta-learning. This article focuses on the VQA running on near-term devices and demonstrates the solution of molecular simulation and combinatorial optimization problems. Besides, we consider the number of iterations and introduce adaptive learning rate and stochastic gradient descent methods.

Our work has two significant contributions. First, we propose a meta-learning variational quantum algorithm (meta-VQA), aiming to extend to larger system sizes and problem instances by optimizing small-scale quantum algorithms. With the help of the recurrent unit method, an effective

Manuscript received 21 May 2021; revised 1 November 2021 and 17 January 2022; accepted 9 February 2022. Date of publication 28 February 2022; date of current version 30 October 2023. This work was supported in part by the Natural Science Foundation of Guangdong Province of China under Grant 2021A1515011440, in part by the Major Program of Guangdong Basic and Applied Research under Grant 2019B030302008, in part by the National Natural Science Foundation of China under Grant 62032009, in part by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province under Grant SKLACSS-202106, and in part by the Outstanding Innovative Talents Cultivation Funded Programs for Doctoral Students of Jinan University under Grant 2021CXB007. *(Corresponding author: Xiaoqing Tan.)*

Rui Huang and Qingshan Xu are with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China.

Xiaoqing Tan is with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China, and also with the Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610025, China (e-mail: ttanxq@jnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3151127.

Digital Object Identifier 10.1109/TNNLS.2022.3151127

variational algorithm can be implemented. At the same time, we introduced the method of weight sharing to learn translation invariant representation, which is an open question in [20]. The whole algorithm is inspired by a gated recurrent unit, which usually has shown excellent results in natural language processing tasks. Second, we deployed a meta-VQA experiment on TensorFlow Quantum (TFQ) processor [21], which is implemented on near-term processors. The experiment includes recurrent unit construction, evaluation circuit model, variational samples, and adaptive learning rate calculation. We compared and analyzed the target energy and variational parameters, which shows that our algorithm has more advantages than the VQE algorithm in terms of learning ability. Importantly, our experiment shows that it is possible to learn on a larger system scale by optimizing small-scale quantum algorithms.

The remainder of this article is organized as follows. Section II lists the basic concepts of quantum information used in this work. We give a brief overview of the VQA in Section III. In Section IV, the meta-VQA used in VQE and QAOA is presented. Section V puts forward the experiment for meta-VQA on the TFQ processor, and finally, Section VI presents our conclusion.

II. Preliminaries

This section introduces the reader to the necessary quantum information background, which is used in the following.

A. Quantum States

Notation such as "$|\ \rangle$" is called the Dirac notation, representing the standard states in quantum mechanics. The difference between classical bits and qubits is that a 1-qubit can be in a linear combination of $|0\rangle$ and $|1\rangle$, often called superposition, i.e., $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are complex numbers and is normalized $|\alpha|^2 + |\beta|^2 = 1$. Mathematically, given the space $\mathbb{C}^d$, the computational basis states of this system are denoted as $\{\overrightarrow{e}_0, \overrightarrow{e}_1, \ldots, \overrightarrow{e}_{d-1}\}$, where the special states $|0\rangle$ (or $|1\rangle$) form the orthonormal basis for this vector space, i.e., $\overrightarrow{e}_i = (0, \ldots, 1, \ldots, 0)^\dagger$ with the $(i+1)$th term is 1 and the others are 0. Besides, notation "†" denotes the complex conjugate of a vector (or matrix), and $|\psi\rangle^\dagger = \langle\psi|$ in the Hilbert space. An n qubits state can be written as $|\phi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle, \ldots, \otimes |\psi_{n-1}\rangle$, where $\otimes$ is the Kronecker product. Thus, a quantum state of such a system is specified by 2^n amplitudes.

B. Unitary Gates

Recalling the definition of superposition states, operations on qubits need to preserve the l_2-norm. In linear algebra, this is called the unitary transformation, which is $U^\dagger U = I$. Here, the notation U is generally used to denote a unitary operator or matrix. Common single-quantum gates (e.g., Pauli gates) and multiquantum gates (e.g., controlled NOT) are listed as follows:

1) Pauli gates

$$I = \sigma^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$Y = \sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1)$$

2) Rotation gates

$$R_x(\theta) = e^{-i\theta X/2} = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$
$$R_y(\theta) = e^{-i\theta Y/2} = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$
$$R_z(\theta) = e^{-i\theta Z/2} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (2)$$

3) Controlled NOT

$$\mathrm{CNOT}(0,1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3)$$

Here, in order to facilitate readers to calculate the objective function of VQA, an important unitary operator decomposition step is shown as follows.

1) Unitary Operator Decomposition:

$$\begin{aligned} U(\theta) &= e^{i\frac{\theta}{2}\sigma_0^z \otimes \sigma_1^z} \\ &= \begin{bmatrix} e^{i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{-i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{-i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e^{i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{-i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= \mathrm{CNOT}(0,1)\{I_0 \otimes R_z(1, -\theta)\}\mathrm{CNOT}(0,1) \end{aligned} \quad (4)$$

where we know that the quantum circuit of this $U(\theta)$ consists of CNOT and R_z.

2) Hartree Fock: Hartree-Fock is an approximate method to determine the wave functions and energies of quantum many-body systems. In computational physics and chemistry, the Hartree-Fock solution is the most accurate description of multielectron systems. In general, it is assumed first that the N-body wave function of the system is approximated by a single permanent (Boson) or a single Slater determinant (Fermion) of N spin orbitals. Second, N coupled equations of N spin orbitals are derived by the variational method. Finally, a solution of the equations yields the Hartree-Fock wave function and energy of this system. In the experiment, we refer to the related Hartree-Fock states for molecular simulation of quantum chemistry.

3) Slater-Type Orbital nG Basis: In computational chemistry, the basis set is used to represent the electron wave function in the Hartree-Fock method (or density functional). It aims to transform the partial differential equations of the model into algebraic equations, which can be executed effectively on the computer. The slater-type orbital (STO)-nG basis set is the most common minimum basis set, which is derived from the minimum STO basis set, where n is an integer and represents the number of Gaussian primitive functions. The minimum basis set uses enough orbitals to hold all the electrons in a neutral atom. The STO-nG basis set can be obtained by the least square fitting of three Gauss orbitals with a single STO. In the following, STO-3G and STO-6G basis are used for molecular simulation experiments.

III. Review of VQA

In this section, we briefly review the VQA, which includes the QAOA for MaxCut [16] and the VQE for Hubbard model [22]. VQA uses classical computers as the quantum outer loop optimizer and updates the circuit parameters to approximate the ground state. The goal of these algorithms is to generate specific wave functions or extract nonlocal information. The realization of VQA by energy minimization is shown in III-C. We refer curious readers to Section II for a more comprehensive introduction.

A. Quantum Approximate Optimization Algorithm

Let us first introduce the general QAOA ansatz and then focus on the application of the MaxCut problem. QAOA first starts from the initial state $|\psi_0\rangle^{\otimes n}$. A wave function is then produced to calculate the cost Hamiltonian $\hat{H}_C$, where the goal is to have a high probability of being measured in a low energy state. The parameterized quantum circuit ansatz is defined as follows.

1) Initialize the uniform superposition state $|\psi_0\rangle^{\otimes n} = (1/\sqrt{2^n})(\sum_{\boldsymbol{x}\in\{0,1\}^n}|\boldsymbol{x}\rangle)$.
2) Applied onto this initial state with a sequence of exponentials of the form
$$\hat{U}(\boldsymbol{\theta}_m,\boldsymbol{\theta}_c)=\prod_{j=1}^{P} e^{-i\theta_m^{(j)}\hat{H}_M}e^{-i\theta_c^{(j)}\hat{H}_C} \tag{5}$$
where $\hat{H}_M=\sum_{j\in\mathcal{V}}\hat{X}_j$ is known as the mixer Hamiltonian and its eigenstate is $|\psi_0\rangle^{\otimes n}$. Besides, the cost Hamiltonian is noncommuting with the mixer Hamiltonian [16], i.e., $[\hat{H}_C,\hat{H}_M]\neq 0$.
3) Then, the final parameterized quantum state output is given by $|\Psi_{\boldsymbol{\theta}}\rangle=\hat{U}(\boldsymbol{\theta}_m,\boldsymbol{\theta}_c)\,|\psi_0\rangle^{\otimes n}$.

In general, P linearly with the number of gates, so $\hat{H}_M$ and $\hat{H}_C$ are usually easy to compile. Now that we have introduced the general QAOA approach, the energy is to be minimized as $\hat{H}_C\equiv f(\hat{\mathbf{Z}})$, where $\hat{\mathbf{Z}}=\{\hat{Z}_j\}_{j=1}^{n}$ with respect to the output parameterized state. For the MaxCut problem, the $\hat{H}_C$ function f is given as follows:
$$\hat{H}_C=f(\hat{\mathbf{Z}})=\sum_{\{j,k\}\in\mathcal{E}}\frac{1}{2}(\hat{I}-\hat{Z}_j\hat{Z}_k) \tag{6}$$
where $\mathcal{E}$ are the edges of the graph $\mathcal{G}$. We can now construct ansatz according to (5) by choosing the value of P and then substituting $\hat{H}_M$ and $\hat{H}_C$. We then obtain a wave function through variational optimization, which is measured to produce a bit string corresponding to the MaxCut problem.

Algorithm 1 VQA

```
Input: A sequence of exponentials $\hat{U}(\boldsymbol{\theta})$, and a
       measurement operator $\hat{H}$.
Output: A bit string $y \in \{0,1\}^n$.
while some condition holds do
    1. Perform the unitary operation $\hat{U}(\boldsymbol{\theta})$ on the initial
    state $|\psi_0\rangle^{\otimes n}$, namely an ansatz $\boldsymbol{\theta} \to |\Psi_{\boldsymbol{\theta}}\rangle$.
    2. Measured the all terms $\hat{H}$, namely
    $\varepsilon(\boldsymbol{\theta}) = \langle\Psi_{\boldsymbol{\theta}}|\,\hat{H}\,|\Psi_{\boldsymbol{\theta}}\rangle$.
    if Training then
        3. Minimize the target energy and calculate the
        gradient $\nabla J^{(t)}(\boldsymbol{\theta})$.
        4. Update parameters, namely
        $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta^{(t)}\nabla J^{(t)}(\boldsymbol{\theta})$.
    else if Testing then
        5. return $\varepsilon(\boldsymbol{\theta})$ and $\boldsymbol{\theta}$.
    end
end
```

B. Variational Quantum Eigensolver

Here, we describe the VQE ansatz, which estimates the ground state energy of the Hamiltonian by optimizing the wave function. The Hubbard Hamiltonian has a 2-D form as follows:
$$\hat{H}=-T\sum_{\langle i,j\rangle,\sigma}\left(\hat{a}^{\dagger}_{i,\sigma}\hat{a}_{j,\sigma}+\hat{a}^{\dagger}_{j,\sigma}\hat{a}_{i,\sigma}\right)+U\sum_{i}\hat{a}^{\dagger}_{i,\uparrow}\hat{a}_{i,\uparrow}\hat{a}^{\dagger}_{i,\downarrow}\hat{a}_{i,\downarrow} \tag{7}$$
where $\hat{a}_{i,\sigma}$ and $\hat{a}^{\dagger}_{i,\sigma}$ are annihilation and creation operators, respectively, at site i with spin $\sigma\in\{\uparrow,\downarrow\}$. Besides, T is a hopping term, and U is a many-body interaction term.

In general, the VQE ansatz is based on the Trotterization approximation of time evolution, and its form is given as follows:
$$\hat{U}(\boldsymbol{\theta}_t,\boldsymbol{\theta}_u)=\prod_{j=1}^{P} e^{-i\theta_t^{(j)}\hat{T}}e^{-i\theta_u^{(j)}\hat{U}} \tag{8}$$
where P is the Trotter steps. The exponentials at each step are derived using the fermionic swap network [22]. The initial state $|\psi_0\rangle^{\otimes n}$ is a Hartree-Fock state for the quantum chemistry problem.

C. VQA by Energy Minimization

The overall procedure of the traditional approach of VQA is shown in Algorithm 1. It is divided into three stages to implement the VQA, namely initialize ansatz, evaluation quantum circuit, and gradient calculation. To sum up, VQA is a hybrid quantum-classical algorithm, which optimizes the variational circuit parameters through multiple iterations to obtain expectations. However, the caveat is that we need to perform multiple iterations according to different problem scenarios and without prior knowledge.

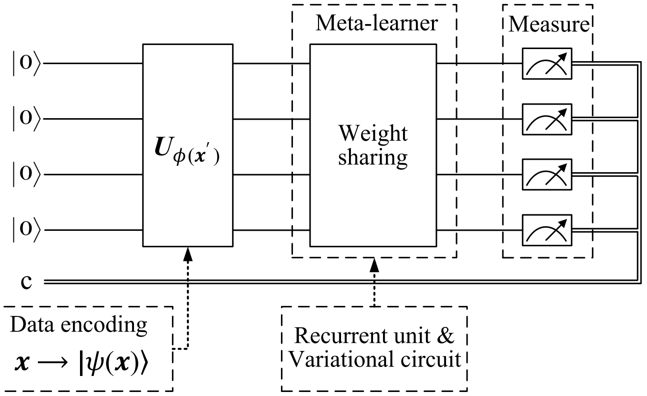

Fig. 1. Illustration of the meta-VQA scheme. $U_{\phi(\boldsymbol{x}')}$ uses encoding to generate quantum datasets, and c is the measured classical data. Inspired by traditional meta-learning, weight sharing is composed of recurrent units and variational circuits. According to specific circuit parameters, the corresponding variables are set in the hidden layer, used to expand to larger system sizes and problems. Then, the recurrent units' neural network layer is used to extract hidden features and improve model accuracy. Finally, the stochastic gradient descent method is used to calculate the model parameters, and an adaptive learning rate is introduced to accelerate the convergence.

IV. Meta-Learning Variational Quantum Algorithm

This section proposes the meta-VQA, which trains classical recurrent units to assist quantum computing, learning to find approximate optima in the parameter landscape. First, we demonstrate our hybrid architecture, in which the classical recurrent units are fed to the previous iterations to assist VQA. The whole process of the meta-VQA scheme is shown in Fig. 1. Second, unrolled the temporal meta-VQA hybrid quantum-classical computational graph, which shares the same parameter set. Besides, the stochastic estimation method is used for the parameterized unitary operator to perform gradient descent. Finally, an adaptive learning rate method is introduced, whose goal is to accelerate the convergence of circuit parameters.

A. Related Work

Verdon *et al.* [23] proposed a meta-learning framework using recurrent neural networks (RNNs), and numerical experiments show that it explores the potential of gradient-free as initialization. Besides, Wilson *et al.* [24] studied a gradient-based long short-term memory meta-learning (LSTM-Meta), which aims to become an independent optimizer. Wang *et al.* [25] proposed a meta-learning algorithm combining LSTM and QNN to solve QAOA problems based on this framework. However, this method of using the QNN to replace ansatz results inconsistent with the traditional QAOA solution.

Our work extends these ideas from five aspects. First, we improved the meta-learning framework and introduced recurrent units combined with VQA to achieve learning to learn. Second, the classical processing unit (CPU) and quantum processing unit (QPU) combine to share the same parameter set. Third, we use stochastic gradient descent to avoid the analysis gradient too costly. Fourth, combined with the adaptive method, the learning rate of circuit parameters is automatically adjusted to accelerate convergence. Finally, a trained meta-VQA is successfully deployed on near-term processors, learning to find approximate optimal in the parameter landscape of VQE/QAOA.

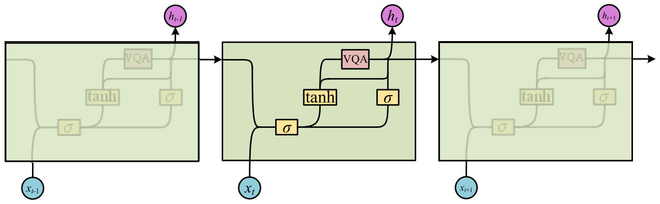

Fig. 2. Unrolled the temporal meta-VQA hybrid quantum-classical computational graph. At the jth iteration, the CPU is fed the previous parameters $\boldsymbol{h}_{t-1}$ and then suggests a new set of parameters $\boldsymbol{\theta}_t$, which is fed to the QPU. Finally, the expected value of the cost Hamiltonian is evaluated by these given parameters. This procedure takes place over several time steps in a process known as unrolling.

To sum up, the similarity between our method and the above work is that the VQA has the ability of meta-learning. The difference is that we study quantum gradients, shared parameter sets, convergence rate, and molecular simulation.

B. Meta-Learning

Meta-learning is the study of how to design machine learning models that learning to learn. It consists of meta-optimization technology, which aims to allow meta-learners to intelligently learn and modify the parameters (or hyperparameters) of the algorithm. In recent years, many new types of research have emerged in meta-learning, including AutoML [26] and Meta-Networks [27]. Besides, meta-learners can train and optimize general function models [1], not limited to machine learning models. Here, we train classical recurrent units to assist quantum computing, learning to find approximate optima in the parameter landscape. The unrolled form of the temporal meta-VQA hybrid quantum-classical computational graph is shown in Fig. 2.

We aim to train classical recurrent units to learn parameter update heuristics to optimize VQA. Here, we introduced recurrent units combined with VQA to achieve learning to learn. We first update the input data $\boldsymbol{h}_{t-1}$ and $\boldsymbol{x}_t$ as follows:

$$r_t = \sigma(W_r \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_r) \tag{9}$$

where σ is a logistic sigmoid function and W_r and b_r are weight and bias, respectively. The activation r_t of the recurrent units at time t can also be a linear combination of the previous activation $\boldsymbol{h}_{t-1}$. Then, the variational parameters $\boldsymbol{\theta}_t$ are calculated by

$$\boldsymbol{\theta}_t = \tanh(W_\theta \cdot r_t + b_\theta) \tag{10}$$

where tanh is the traditional recurrent units. Here, $\boldsymbol{\theta}_t$ is calculated by memory cell that modulates the amount of memory content. Next, a suggested set of parameters $\boldsymbol{\theta}_t$ is fed to the QPU. The wave function through variational optimization, which is measured to produce a bit string output

$y_t = \varepsilon(\boldsymbol{\theta}) = \langle \Psi_{\boldsymbol{\theta}} | \hat{H} | \Psi_{\boldsymbol{\theta}} \rangle$. Traditionally, the update of the recurrent hidden state s_t is implemented as

$$s_t = \sigma(W_s \cdot r_t + b_s) \tag{11}$$

where s_t depends on the previous activation each time. Finally, the output $\boldsymbol{h}_t$ is reconstructed by

$$\boldsymbol{h}_t = g(s_t, \boldsymbol{\theta}_t, y_t) \tag{12}$$

where g is the concatenate function as the input of the next stage. The computational graph of meta-VQA consists of many copies, each sharing the same set of parameters. Now, the meta-VQA model we designed has the ability to learning to learn. Simple examples of this are given in Appendix A.

In classical meta-learning research, optimization models (such as RNN [28] and LSTM [29]) are usually used to perform intuitive gradient calculations. Among them, RNN is used for the learning problem of sequence-to-sequence neural network architecture. LSTM is to mitigate the vanishing or exploding gradient in the RNN architecture. The above method changes the cell state according to the hidden state, gate, and input data at each time step. However, this method independently iterates between the optimization model and the learning model. A comparison of meta-VQA and state-of-the-art methods is presented in Appendix B.

In this context, the meta-VQA model is an improved meta-learning framework that combined recurrent units with VQA to achieve learning to learn.

C. Stochastic Gradient Descent

Now, we introduce the gradient optimization of meta-VQA. The gradients of hybrid quantum-classical computational graphs can be categorized as gradient-free [30] and gradient-based [31]. Among them, we can obtain the gradient through backpropagation methods, such as automatic differentiation [32], black boxes [33], or hybrid quantum-classical [34]. Here, we use gradient-based optimization parameters, which can speed up training convergence in some cases.

Just like classical models, the task of learning an arbitrary wave function is expressed as minimizing the loss function $\mathcal{L}(\boldsymbol{\theta})$, also known as the objective function. The iteration of gradient descent is a common gradient-based method, in which the parameter $\boldsymbol{\theta}$ is updated with gradient descent as

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta \frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}) \tag{13}$$

where η is the learning rate, which is a hyperparameter used to control the update magnitude. We can use the finite difference scheme to calculate the partial derivatives numerically or simultaneous perturbation stochastic approximation. Due to truncation and rounding errors, the calculation results of the finite difference method are less accurate.

To leverage gradient-based techniques for the learning of multilayered models, we use the parameter shift method [35] to estimate the analysis gradient of the variational circuit. Recall that the output of the model is the loss function $\mathcal{L}(\boldsymbol{\theta})$, which is an estimate of the expected value $\varepsilon(\boldsymbol{\theta})$. We can use the chain rule to write the derivative as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\langle \hat{H} \rangle_{\boldsymbol{\theta}+\frac{\pi}{2}} - \langle \hat{H} \rangle_{\boldsymbol{\theta}-\frac{\pi}{2}}}{2} \tag{14}$$

where subscript $\boldsymbol{\theta} \pm (\pi/2)$ represents the shifted parameter to be used for evaluation. Note that this method evaluates two circuits for each parameter, which makes the analysis gradient too costly. Besides, another method estimates the partial derivative by adding an auxiliary qubit as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathrm{Im}\Big(\mathrm{tr}\Big(\hat{H} U_{J:j+1} P_j U_{j:1} |0\rangle \langle 0| U_{J:1}^{\dagger}\Big)\Big) \tag{15}$$

where $P_j \in \{I, X, Y, Z\}^{\otimes n}$ is a tensor product of n Pauli matrices and $U_j = \exp(-(i/2)\theta_j P_j)$. It is an indirect measurement method, and we can use the Hadamard test for evaluation [36]. In this context, in order to remedy the extra overhead, Harrow and Napp [37] proposed stochastic gradient descent. It is randomly selected according to the distribution of the coefficient weight of each item in the generator. Here, we follow the approach as given in the following:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{j=1}^{M} \sum_{k=1}^{K} \gamma_k \langle \hat{H} \rangle_{\boldsymbol{\theta}} \tag{16}$$

and introducing random variables, we have

$$\nabla J^{(t)}(\boldsymbol{\theta}) = \sum_{j=1}^{M} \sum_{k=1}^{K} \gamma_k \hat{h}_j(\boldsymbol{\theta}) \tag{17}$$

where γ is the K-term coefficient and $\hat{h}_j(\boldsymbol{\theta})$ is the n-sample mean estimator of $\langle \hat{H} \rangle_{\boldsymbol{\theta}}$.

To sum up, for each iteration t, we first calculate the gradient $\nabla J^{(t)}(\boldsymbol{\theta})$ and select the learning rate η. Then, similar to (13), the parameters are updated by $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta \nabla J^{(t)}(\boldsymbol{\theta})$.

D. Adaptive Learning Rate

At present, the update rate of all parameters depends on a fixed learning rate, but it is difficult to choose an appropriate learning rate. Too small a learning rate will result in slow convergence, while too large a learning rate will hinder convergence and cause the expected value to fluctuate around the minimum. In addition, when the model shows the features of different frequencies under sparse data, we hope to have a different update rate for each parameter. Here, we use a method based on adaptive learning rate to optimize parameters, which speeds up training convergence.

Just like the classic model, stochastic gradient descent has difficulty in navigating the ravine, and it is easy to oscillate on slopes. Momentum is a way to help dampen oscillations by $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - (\gamma m_{t-1} + \eta \nabla J^{(t)}(\boldsymbol{\theta}))$, where m_{t-1} is the momentum term and γ is its decay term. It shows that momentum has taken a step in the direction of the previous momentum vector and the current gradient.

Next, we merge momentum into adaptive moment estimation (Adam) [38]. Recall that the Adam update rule

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\frac{\gamma m_{t-1}}{1 - \gamma_t} + \frac{(1 - \gamma) \nabla J^{(t)}(\boldsymbol{\theta})}{1 - \gamma_t} \right) \tag{18}$$

Algorithm 2 Meta-VQA

```
Input: A sequence of exponentials $\hat{U}(\boldsymbol{\theta})$, and a
       measurement operator $\hat{H}$.
Output: A bit string $y \in \{0, 1\}^n$.
while some condition holds do
    1. Perform the unitary operation $\hat{U}(\boldsymbol{\theta})$ on the initial
    state $|\psi_0\rangle^{\otimes n}$, namely an ansatz $\boldsymbol{\theta} \to |\Psi_{\boldsymbol{\theta}}\rangle$.
    2. Perform meta-learning to shares the same parameter
    set, namely $\boldsymbol{h}_t = g(s_t, \boldsymbol{\theta}_t, y_t)$.
    3. Measured the all terms $\hat{H}$, namely
    $\varepsilon(\boldsymbol{\theta}) = \langle\Psi_{\boldsymbol{\theta}}| \hat{H} |\Psi_{\boldsymbol{\theta}}\rangle$.
    if Training then
        4. Minimize the target energy and stochastic
        gradient descent, namely
        $\nabla J^{(t)}(\boldsymbol{\theta}) = \sum_{j=1}^{M} \sum_{k=1}^{K} \gamma_k \hat{h}_j(\boldsymbol{\theta})$.
        5. Adaptive learning rate and update parameters,
        namely $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\left(\gamma \hat{m}_t + \frac{(1-\gamma)\nabla J^{(t)}(\boldsymbol{\theta})}{1-\gamma_t}\right)$.
    else if Testing then
        6. return $\varepsilon(\boldsymbol{\theta})$ and $\boldsymbol{\theta}$.
    end
end
```

where v_t is the second momentum and $\hat{v}_t = (v_t)/(1 - \gamma_t)$. It is the exponential decay average of the past squared gradients. Now, we introduce the idea of Nesterov momentum, and let $\hat{m}_t = (m_t)/(1 - \gamma_t)$ correct the momentum vector offset of the previous time step. We have

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\left(\gamma \hat{m}_t + \frac{(1-\gamma)\nabla J^{(t)}(\boldsymbol{\theta})}{1-\gamma_t}\right). \quad (19)$$

Besides, the form of momentum can be combined with other algorithms based on adaptive learning rates, such as Nesterov-accelerated adaptive moment estimation (Nadam) [39]. Simple examples of this are given in Appendix B. To sum up, the overall procedure of the meta-VQA is shown in Algorithm 2.

V. Experiments

In this section, we carry out meta-VQA experiments on the TFQ processor for MaxCut and several representative molecules, including H_2, lithium hydride (LiH), and HeH$^+$. Specifically, we perform simulation QPU after designing and training a hybrid quantum-classical model and run the quantum parts of these models online on near-term processors. The hidden state consists of 15 neurons in the experiment, and the recurrent unit is five time steps. Besides, we compared our approach with state-of-the-art methods, including recurrent neural networks meta-learning (RNN-Meta) [23] and LSTM-Meta [24]. These experiments reveal that the optimization based on the meta-learning algorithm is superior to VQA and shows that the meta-learners learn to learn the target energy intelligently. Remarkably, it is used to expand to larger system sizes and problems.

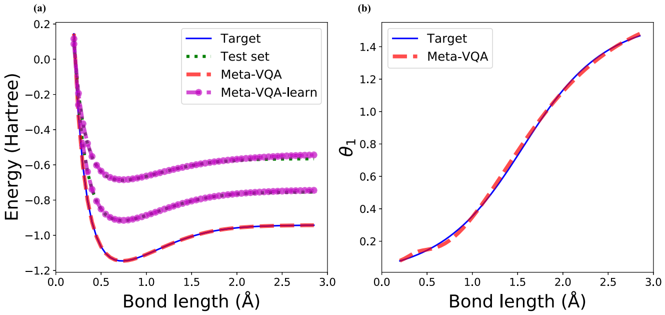

Fig. 3. Learn the information of H_2 with different bond lengths using meta-VQA. In both figures, target, test set, and meta-VQA are indicated by solid, dotted, and dashed lines, respectively. (a) Target energies and optimization results for the meta-VQA. (b) Corresponding variational parameter of each epoch.

A. Molecular Hydrogen

For H_2, we consider the Hamiltonian involving two qubits, following the scheme in [40]. Using the STO-3G basis to construct Hamiltonian can be written as

$$H_{H_2} = c_0 I + c_1\sigma_0^z + c_2\sigma_1^z + c_3\sigma_0^z\sigma_1^z + c_4\sigma_0^x\sigma_1^x + c_5\sigma_0^y\sigma_1^y \quad (20)$$

where c_i is the coefficient that depends on the bond length and the Hartree-Fock state is $|01\rangle$. Besides, the unitary coupled cluster (UCC) ansatz is as follows: $U(\theta) = \exp(-i\theta\sigma_0^x\sigma_1^y)$.

Effective Hamiltonians are calculated from 54 bond lengths, uniformly chosen from 0.25 to 2.85 a.u. We extracted all samples from the original data as the training set, and two curves were randomly selected as the test set to verify the accuracy of the algorithm. In order to visualize the performance of meta-VQA, the ground state energies and variational parameters are shown in Fig. 3. Remarkably, the target energies at different bond lengths match the actual values, and the variational parameters also evolve to the optimal target values. Besides, two random curves were used to test the learning ability of meta-VQA, as shown by the magenta dashed line in Fig. 3(a). The results showed that meta-VQA could learn the ground state energy of random curves. It can be understood as a learning process, where the meta-learner intelligently learns to modify the parameters (or hyperparameters) of the algorithm to calculate the optimal value.

Table I lists the root-mean-squared error (RMSE) results under different algorithms in comparative studies. The "Algorithm" column describes the RNN-Meta, LSTM-Meta, or Meta-VQA algorithm. First, from the experimental results in column 2 of Table I, we notice that the RMSE of the three algorithms reaches the E-04 accuracy. It shows that these algorithms can adapt to the training dataset of molecular hydrogen. Second, comparing the experimental results of the third and fourth columns, we can see that the RMSE results of the proposed meta-VQA algorithm achieve better performance than those of the model using RNN or LSTM. The reasons for the effectiveness are threefold.

1) Meta-VQA trains recurrent units to assist quantum computation and establishes a weight sharing method,

TABLE I

RESULTS OF THE COMPARISON ANALYSIS OF MOLECULAR HYDROGEN

Algorithm	RMSE		
	Target	Test set 1	Test set 2
RNN-Meta	5.7192E-04	1.8193E-01	4.7549E-02
LSTM-Meta	3.9342E-04	1.9567E-01	5.4051E-02
Meta-VQA	**2.8319E-04**	**3.8888E-03**	**9.2669E-03**

which is deficient in traditional methods RNN-Meta and LSTM-Meta.

2) RNN-Meta concatenates the previous output with the current input, only considering the state of the most recent moment.
3) LSTM-Meta introduces the concept of cell state based on RNN, which is used for information memory and forgetting. Long short-term memory cannot be well established under a single variational parameter.

It reveals that the circuit angle generated by meta-VQA is closer to the target value in the same test set. In this context, the comparison method iterates independently between the optimization and learning models, resulting in a lack of adaptability to the input data.

B. Lithium Hydride

For LiH, we consider the Hamiltonian involving three qubits by the Bravyi–Kitaev (BK) transformation, following the scheme in [40]. Using the STO-6G basis to construct Hamiltonian can be written as

$$\begin{aligned} H_{\mathrm{LiH}} = {} & c_0 I + c_1\sigma_0^z + c_2\sigma_1^z + c_3\sigma_2^z + c_4\sigma_0^z\sigma_1^z + c_5\sigma_0^z\sigma_2^z \\ & + c_6\sigma_1^z\sigma_2^z + c_7\sigma_0^x\sigma_1^x + c_8\sigma_0^x\sigma_2^x + c_9\sigma_1^x\sigma_2^x \\ & + c_{10}\sigma_0^y\sigma_1^y + c_{11}\sigma_0^y\sigma_2^y + c_{12}\sigma_1^y\sigma_2^y \end{aligned} \tag{21}$$

where the Hartree-Fock state is $|111\rangle$. The UCC ansatz is given as follows:

$$U(\theta_1, \theta_2) = \exp\left(-i\theta_1\sigma_0^x\sigma_1^y\right)\exp\left(-i\theta_2\sigma_0^x\sigma_2^y\right). \tag{22}$$

Effective Hamiltonians are calculated from 50 bond lengths, uniformly chosen from 0.3 to 5.0 a.u. We extracted all samples from the original data as the training set, and two curves were randomly selected as the test set. The ground state energies and variational parameters are shown in Fig. 4. The dashed line in Fig. 4(a) indicates that the target energies at different bond lengths match the actual values. Remarkably, variational parameters θ_1 and θ_2 both evolve to the optimal target values, as shown in Fig. 4(b). The results showed that meta-VQA could learn the ground state energy of random curves. In such a learning process, meta-learners can quickly learn the objective function of two variational parameters.

The results using different algorithms are listed in Table II, where the meta-VQA achieved training accuracy of E-04, superior to RNN and LSTM methods. Compared with Table I, meta-VQA showed excellent learning ability in the tasks with two variational parameters. The results show that all three algorithms can learn the ground state energy of the random curve test set. In test set 1, the learning ability of the proposed algorithm is better than that of the comparison method. However, in test set 2, the results of the three algorithms are similar. The reasons for the effects are twofold.

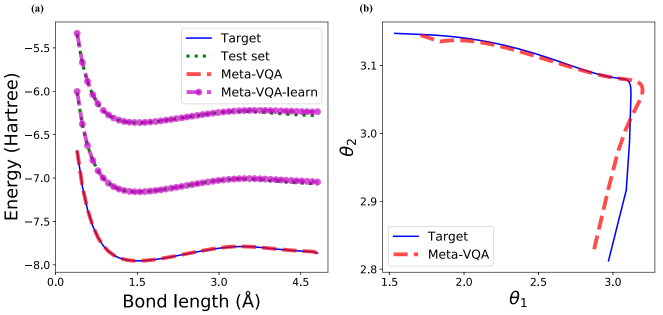

Fig. 4. Learn the information of LiH with different bond lengths using meta-VQA. In both figures, target, test set, and meta-VQA are indicated by solid, dotted, and dashed lines, respectively. (a) Target energies and optimization results for the meta-VQA. (b) Proportional results between two variational parameters. We can see that the meta-VQA calculation results are consistent with the target parameters.

TABLE II

RESULTS OF THE COMPARISON ANALYSIS OF LiH

Algorithm	RMSE		
	Target	Test set 1	Test set 2
RNN-Meta	1.4056E-03	4.2251E-02	3.1489E-02
LSTM-Meta	2.1297E-03	5.4466E-02	4.3791E-02
Meta-VQA	**8.7942E-04**	**2.3046E-03**	**1.3279E-02**

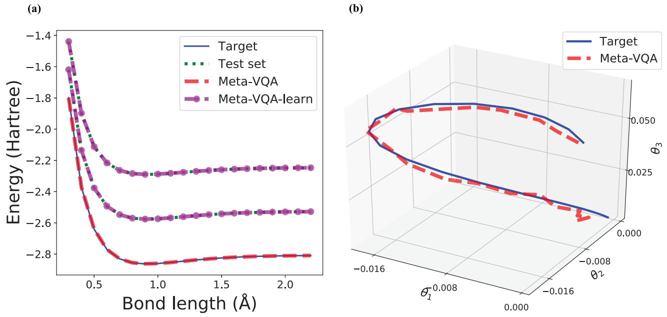

Fig. 5. Learn the information of HeH$^+$ with different bond lengths using meta-VQA. In both figures, target, test set, and meta-VQA are indicated by solid, dotted, and dashed lines, respectively. (a) Target energies and optimization results for the meta-VQA. (b) 3-D structure of variational parameters.

1) The bounds of the target angles are small, and meta-VQA does not demonstrate a decisive learning advantage.
2) In the case of two parameters, LSTM-Meta filters the previous state, weakening the correlation between variational parameters.

Similar to Table I, the RMSE results of the proposed meta-VQA algorithm achieve better performance than those of the model using RNN or LSTM. This conclusion also holds for two variational parameters tasks.

C. Helium Hydride Cation

For a more complicated HeH$^+$ molecule, we consider the Hamiltonian involving four qubits, following the scheme

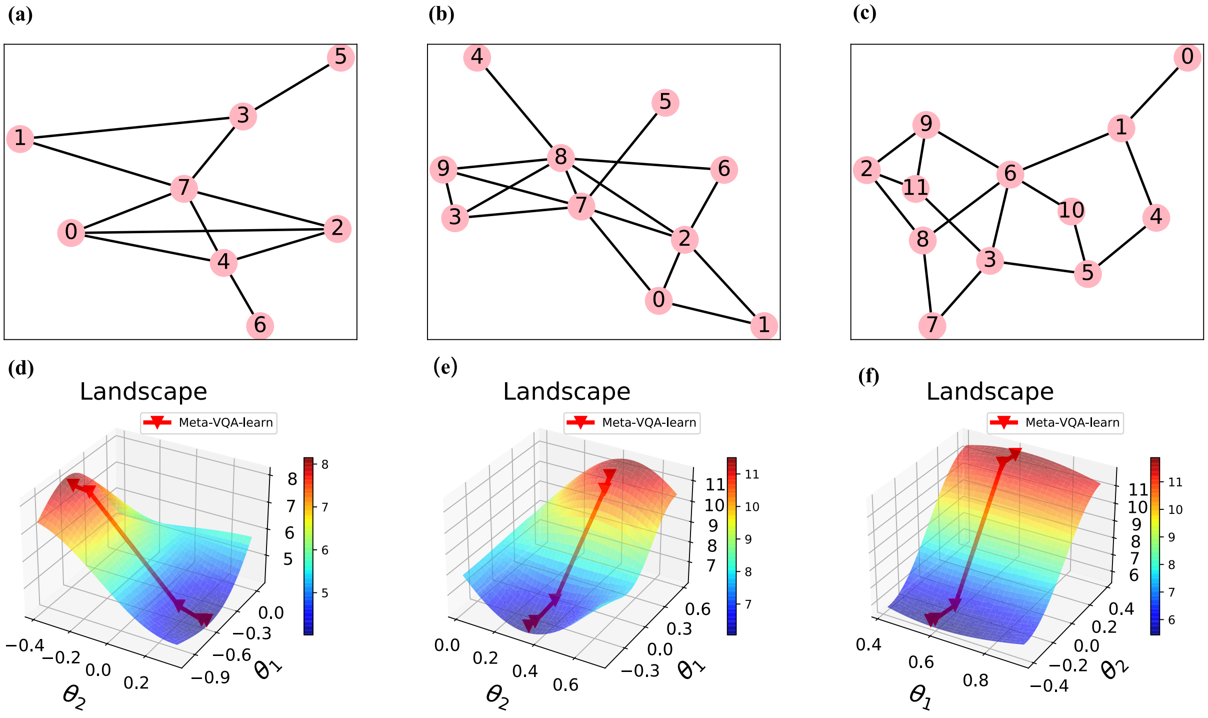

Fig. 6. Use meta-VQA to solve the 8, 10, and 12 qubits MaxCut problem after training the 6, 8, and 10 qubits scene. (a)–(c) Random graph test samples of 8, 10, and 12 qubits. (d)–(f) Evolution of model parameters after training to larger system scale and problem instances.

in [41]. Using the STO-3G basis to construct Hamiltonian can be written as

$$\begin{aligned}H_{\mathrm{HeH}^+} = {} & c_0 I + c_1\sigma_0^z + c_2\sigma_1^z + c_3\sigma_2^z + c_4\sigma_3^z \\ & + \sum_{i,j=0}^{3}\left(c_{z_{ij}}\sigma_i^z\sigma_j^z + c_{x_{ij}}\sigma_i^x\sigma_j^x + c_{y_{ij}}\sigma_i^y\sigma_j^y\right) \\ & + \sum_{i=0}^{1}\left(c_{5_i}\sigma_i^x\sigma_{i+1}^z\sigma_{i+2}^x + c_{6_i}\sigma_i^y\sigma_{i+1}^z\sigma_{i+2}^y\right) \\ & + c_7\sigma_0^x\sigma_1^x\sigma_2^y\sigma_3^y + c_8\sigma_0^y\sigma_1^y\sigma_2^x\sigma_3^x + c_9\sigma_0^x\sigma_1^y\sigma_2^y\sigma_3^x \\ & + c_{10}\sigma_0^y\sigma_1^x\sigma_2^x\sigma_3^y + c_{11}\sigma_0^x\sigma_1^z\sigma_2^x\sigma_3^z + c_{12}\sigma_0^z\sigma_1^x\sigma_2^z\sigma_3^x \\ & + c_{13}\sigma_0^y\sigma_1^z\sigma_2^y\sigma_3^z + c_{14}\sigma_0^z\sigma_1^y\sigma_2^z\sigma_3^y \end{aligned} \tag{23}$$

where c_i is the coefficient that depends on the bond length and the Hartree-Fock reference state is $|0011\rangle$. The UCC ansatz can be written as $U(\theta_1, \theta_2, \theta_3) = \exp(-i\theta_1\sigma_0^x\sigma_2^y)\exp(-i\theta_2\sigma_1^x\sigma_3^y)\exp(-i\theta_3\sigma_0^x\sigma_1^x\sigma_2^x\sigma_3^y)$. Effective Hamiltonians are calculated from 30 bond lengths, uniformly chosen from 0.25 to 2.50 a.u. We extracted all samples from the original data as the training set, and two curves were randomly selected as the test set to verify the accuracy of the algorithm. The target energies at different bond lengths match the actual values, as shown in Fig. 5(a). The results showed that meta-VQA could learn the ground state energy of random curves. In order to visualize the variational performance of meta-VQA, the 3-D structure of variational parameters is shown in Fig. 5(b). Remarkably, variational parameters θ_1, θ_2, and θ_3 evolve to the optimal target values. It again demonstrates the learning ability of the meta-VQA algorithm under more complicated molecular structures.

TABLE III
RESULTS OF THE COMPARISON ANALYSIS OF HELIUM HYDRIDE CATION

Algorithm	RMSE		
	Target	Test set 1	Test set 2
RNN-Meta	1.3632E-04	2.8357E-03	1.1506E-03
LSTM-Meta	1.5426E-04	2.3429E-03	1.1268E-03
Meta-VQA	**4.9045E-05**	**5.7677E-05**	**1.5278E-04**

Table III presents the various modeling performances obtained by different methods. RMSE metrics show that the performance of the proposed meta-VQA algorithm on training and test datasets is better than that of the above RNN and LSTM models. Compared with Table II, meta-VQA showed excellent learning ability in the tasks with three variational parameters. The reasons for the effects are twofold.

1) Recurrent unit assists quantum computing that can learn complex multiparameter quantum circuits. It allows meta-VQA to demonstrate excellent learning abilities.
2) In the case of three variational parameters, RNN-meta relies on the previous output and the current input, resulting in long short-term memory forgetting.

For the LSTM model, the metrics on the test dataset are lower than the RNN model. It indicates that LSTM is more advantageous than RNN under three variational parameters. Yet, RMSE results of the meta-VQA algorithm achieve excellent performance, and this conclusion also holds for three variational parameters tasks.

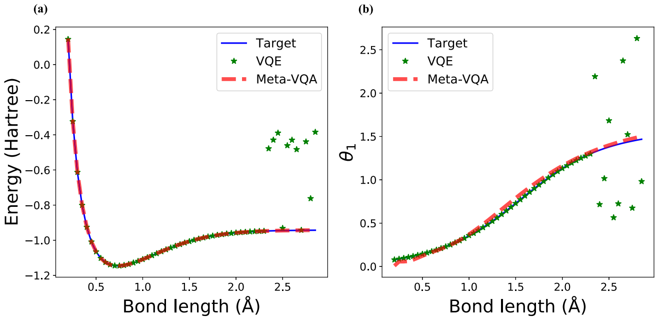

Fig. 7. Illustration of the learning ability of the two algorithms. In both figures, target, VQE, and meta-VQA are indicated by solid, star marker, and dashed lines, respectively. (a) Energy as a function of the Adam epochs. (b) Corresponding variational parameter of each epoch.

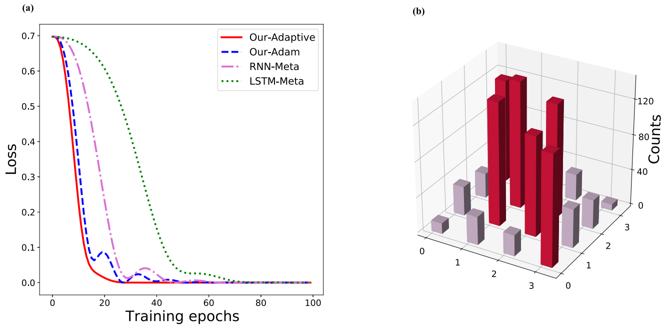

Fig. 8. Illustration of the learning ability of the three algorithms. (a) Training loss of meta-VQA, RNN-Meta, and LSTM-Meta in the three-regular graph dataset. In the figure, Our-Adaptive, Our-Adam, RNN-Meta, and LSTM-Meta are distinguished by solid, dashed, dashed-dotted, and dotted lines, respectively. (b) Simulation of real quantum computer sampling (bit string) using Our-Adaptive model.

D. MaxCut

We now turn to consider the MaxCut problem, which is to solve the bit string through the wave function. First, we randomly generate dataset graph $\mathcal{G}$ with six, eight, and ten qubits, and each dataset contains 500 training task samples. Here, we use 50 samples of random graphs of 8, 10, and 12 qubits as test datasets. Second, we construct the exponentials in the form of (5) for $P = 1$. Then, we deploy the quantum data into the MaxCut circuit and use meta-VQA to optimize the parameters. Finally, we use the test set to calculate the target parameters that are extended to larger system sizes and problems, as shown in Fig. 6.

In landscape parameters, the cost function is optimal when the color of the region is close to deep blue. The red (down) triangle line represents the path of meta-VQA searching for two parameters in a large-scale system through five time steps. Fig. 6 shows that meta-VQA starts guessing near the plateau and calculates the optimal next step around the landscape. In this context, the random initial point flows to the minimum in the meta-VQA method.

The results show that meta-VQA can find the optimal path in random graphs of eight, ten, and twelve qubits, respectively. Remarkably, variational parameters θ_m and θ_c evolve to the optimal target values in the form of (5). In such a learning process, meta-VQA can be expanded to larger system sizes and problem instances by optimizing small-scale quantum algorithms.

VI. Conclusion and Discussion

Here, we combine meta-learning to find the ground state energy of the circuit. As the first step for meta-learning to optimize quantum chemistry and solve approximate quantum problems, it is expected that meta-VQA can be tested on other more general problems. In quantum chemistry, we follow UCC ansatz and only consider a small number of variational parameters. In quantum approximation optimization, we solve the random graph problem. Remarkably, meta-VQA can learn nonlocal information by optimizing small-scale quantum algorithms.

In the following, we compare our proposed meta-learning method with the other three methods, VQA, RNN-Meta, and LSTM-Meta, pointing out their respective advantages and disadvantages. The advantage of VQA is that it can accurately match the target energy during the training epoch. Its drawback is that the number of rounds trained increases with the size of the system and the lack of learning ability for unknown input data. The advantage of RNN-Meta is that it can display learning ability on a simple quantum circuit, while its disadvantage is adaptive learning ability and output accuracy. The advantage of LSTM-Meta is that it can display learning ability on multiparameter complex quantum circuits. Yet, its disadvantage is that it lacks the adaptability to input data and output accuracy. The advantage of our method is that it is not only suitable for quantum circuits with several kinds of variational parameters but also can be extended to more extensive system sizes and problem instances by optimizing small-scale quantum algorithms. However, for some higher dimensional optimization problems, we can further study meta-VQA in the future. The disadvantage of our method is that unknown input learning is not suitable for high RMSE accuracy, yet it can be used as an auxiliary optimal path calculation.

Federated learning [42] allows multiple users, called clients, to collaborate on training a shared global model without sharing data on the local device. Inspired by our previous work [43], model parameters can also be updated by blind quantum computation or delegated quantum computation. Here, we are interested in efficient VQA based on the decentralized data. It is expected that this algorithm can help us reduce expensive high-dimensional gradient resources. Motivated by the federated learning algorithm, meta-VQA can be used for data privacy real applications, such as federated meta-learning.

In the future, we will further study the coordinated implementation of meta-VQA distributed multiquantum devices, which may inspire investigations in the field of secure QML. Furthermore, our method can be considered for the barren plateau problem [44], where the gradient along any reasonable direction is exponentially small in the training of parameterized quantum circuits.

We have shown that the proposed meta-VQA algorithm has an excellent performance in variational quantum tasks. The

TABLE IV

COMPARISON OF THE SIMULATION SAMPLING RESULTS OF META-VQA, RNN-META, AND LSTM-META IN REAL QUANTUM COMPUTERS. IN THIS CONTEXT, QUANTUM STATES ARE REPRESENTED BY $|q_3q_2q_1q_0\rangle$, AND ACCURACY IS THE MEAN ACCURACY AND STANDARD DEVIATION (%)

Algorithm	$\|0011\rangle$	$\|0101\rangle$	$\|0110\rangle$	$\|1001\rangle$	$\|1010\rangle$	$\|1100\rangle$	Accuracy
RNN-Meta	117 ± 14	128 ± 10	121 ± 11	131 ± 9	129 ± 8	128 ± 11	73.50 ± 0.01
LSTM-Meta	118 ± 14	129 ± 10	121 ± 11	131 ± 10	129 ± 9	130 ± 12	73.96 ± 0.01
Meta-VQA	**119** ± 14	**130** ± 10	**122** ± 11	**133** ± 10	**130** ± 9	**131** ± 12	**74.69** ± 0.01

meta-VQA has been used to solve for ground states of MaxCut and several representative molecules, including H_2, LiH, and HeH$^+$. Remarkably, we have demonstrated that the meta-VQA can be expanded to larger system sizes and problem instances by optimizing small-scale quantum algorithms.

APPENDIX A
META-VQA PERFORMANCE

In this appendix, we demonstrate the molecular hydrogen two qubits VQE task, in which the meta-learning has excellent performance. First, we have uniformly chosen 54 samples in the H_2 from 0.25 to 2.85 a.u. Of all the samples, 80% are for training and 20% are for testing. Then, we deploy the quantum data into the corresponding circuit and use stochastic gradients to optimize the parameters. Finally, we use the bond lengths of the test samples to calculate the target energies and variational parameters. To show the learning futures of the meta-VQA algorithm, in Fig. 7, we compare the optimization process between our algorithm and the conventional VQE algorithm.

Fig. 7 shows the target energies and variational parameters of the H_2 using the above two algorithms. Both algorithms can match the target energy during the training epochs. The dashed line in Fig. 7(a) indicates that the target energy can be accurately calculated during the test epochs; however, the star marker is far from the target value. In addition, the dashed line in Fig. 7(b) shows that the variational parameter can accurately match the target values. Therefore, the results show that for the same dataset, the performance of the meta-VQA is better than that of the conventional VQE. Notably, the reason is that meta-learner can learn new samples based on prior knowledge.

APPENDIX B
COMPARISON OF META-VQA WITH RELATED METHODS

In this appendix, we compared our approach with state-of-the-art methods, including RNN-Meta [23] and LSTM-Meta [24]. We demonstrate the four qubits QAOA task for solving the MaxCut problem. We first randomly generate a three-regular graph $\mathcal{G}$ as the training dataset. Then, we construct the exponentials in the form of (5) for $P = 1$ and deploy the quantum data into the corresponding circuit. In this work, we also compare our adaptive learning rate method (for more details, see Section IV-D).

In order to visualize the performance of training, the loss of meta-VQA, RNN-Meta, and LSTM-Meta algorithms is shown in Fig. 8(a). As the training epoch increases, the final loss of the four models converges to a specific scalar. The loss of the Our-Adam model tends to be stable after 50 epochs, which is better than 62 epochs of RNN-Meta and 75 epochs of LSTM-Meta. After introducing the adaptive learning rate in Section IV-D, i.e., Our-Adaptive, it can reach the optimal value after only 25 epochs. It indicates that Our-Adaptive can effectively speed up the model training and avoid overshoot phenomenon similar to other models, as shown in Fig. 8(a).

Besides, the quantum states of the $\mathcal{G}$ solution are $|0011\rangle$, $|0101\rangle$, $|0110\rangle$, $|1001\rangle$, $|1010\rangle$, and $|1100\rangle$. Fig. 8(b) shows the bit string sampling results of a real quantum computer simulated by the Our-Adaptive model repeated 1024 times. The simulation sampling results of the three algorithms in real quantum computers are shown in Table IV. The experimental data are based on the results of five experiments, in which bold values indicate the best results for each quantum state. The reasons for the effects are twofold.

1) Our-Adaptive can effectively speed up the model training and avoid overshoot phenomenon similar to other models.
2) The comparison method iterates independently between the optimization and learning models, resulting in a lack of adaptability to the input data.

It shows that our proposed method achieves excellent performance in six optimal solutions and is superior to RNN-Meta and LSTM-Meta in model accuracy.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and editors for their suggestions to improve the quality of this article.

REFERENCES

[1] Y. Chen *et al.*, "Learning to learn without gradient descent by gradient descent," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 748–756.

[2] M. Andrychowicz *et al.*, "Learning to learn by gradient descent by gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3981–3989.

[3] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.

[4] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999v3*. [Online]. Available: https://arxiv.org/abs/1803.02999v3

[5] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Burlington, MA, USA: Elsevier, 2014.

[6] N. H. Nguyen, E. C. Behrman, M. A. Moustafa, and J. E. Steck, "Benchmarking neural networks for quantum computations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2522–2531, Jul. 2020.

[7] F. M. de Paula Neto, T. B. Ludermir, W. R. de Oliveira, and A. J. da Silva, "Implementing any nonlinear quantum neuron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3741–3746, Sep. 2020.

[8] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. H. Liew, "Reconstructing quantum states with quantum reservoir networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3148–3155, Jul. 2021.

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Phys.*, vol. 10, pp. 631–633, Jul. 2014.

[10] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," *Phys. Rev. A, Gen. Phys.*, vol. 94, no. 2, Aug. 2016, Art. no. 022342.

[11] R. Huang, X. Tan, and Q. Xu, "Quantum algorithm for hyperparameters estimation," *Quantum Sci. Technol.*, vol. 5, no. 4, Aug. 2020, Art. no. 045008.

[12] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Phys. Rev. Lett.*, vol. 100, no. 16, Apr. 2008, Art. no. 160501.

[13] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[14] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Sci. Technol.*, vol. 4, no. 4, Nov. 2019, Art. no. 043001.

[15] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014.

[16] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*. [Online]. Available: https://arxiv.org/abs/1411.4028

[17] R. Huang, X. Tan, and Q. Xu, "Variational quantum tensor networks classifiers," *Neurocomputing*, vol. 452, pp. 89–98, Sep. 2021.

[18] D.-B. Zhang and T. Yin, "Collective optimization for variational quantum eigensolvers," *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 3, Mar. 2020, Art. no. 032311.

[19] G. Verdon, J. M. Arrazola, K. Bradler, and N. Killoran, "A quantum approximate optimization algorithm for continuous problems," 2019, *arXiv:1902.00409*. [Online]. Available: https://arxiv.org/abs/1902.00409

[20] E. Grant *et al.*, "Hierarchical quantum classifiers," *NPJ Quantum Inf.*, vol. 4, no. 1, p. 65, Dec. 2018.

[21] M. Broughton *et al.*, "Tensorflow quantum: A software framework for quantum machine learning," 2020, *arXiv:2003.02989v2*. [Online]. Available: https://arxiv.org/abs/2003.02989v2

[22] I. D. Kivlichan *et al.*, "Quantum simulation of electronic structure with linear depth and connectivity," *Phys. Rev. Lett.*, vol. 120, no. 11, Mar. 2018, Art. no. 110501.

[23] G. Verdon *et al.*, "Learning to learn with quantum neural networks via classical neural networks," 2019, *arXiv:1907.05415*. [Online]. Available: https://arxiv.org/abs/1907.05415

[24] M. Wilson, S. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, and E. Rieffel, "Optimizing quantum heuristics with meta-learning," 2019, *arXiv:1908.03185*. [Online]. Available: https://arxiv.org/abs/1908.03185

[25] H. Wang, J. Zhao, B. Wang, and L. Tong, "A quantum approximate optimization algorithm with metalearning for MaxCut problem and its simulation via TensorFlow quantum," *Math. Problems Eng.*, vol. 2021, pp. 1–11, Mar. 2021.

[26] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Red Hook, NY, USA: Curran Associates, 2015, pp. 2962–2970.

[27] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2554–2563.

[28] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019v4*. [Online]. Available: https://arxiv.org/abs/1506.00019v4

[29] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 459–468.

[30] C. Kokail *et al.*, "Self-verifying variational quantum simulation of lattice models," *Nature*, vol. 569, no. 7756, pp. 355–360, May 2019.

[31] A. Kandala *et al.*, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, Sep. 2017.

[32] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 5595–5637, 2017.

[33] C. Audet and M. Kokkolaras, "Blackbox and derivative-free optimization: Theory, algorithms and applications," *Optim. Eng.*, vol. 17, no. 1, pp. 1–2, Mar. 2016.

[34] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Phys. Rev. A, Gen. Phys.*, vol. 99, no. 3, Mar. 2019, Art. no. 032331.

[35] J. Li, X. Yang, X. Peng, and C.-P. Sun, "Hybrid quantum-classical approach to quantum optimal control," *Phys. Rev. Lett.*, vol. 118, no. 15, Apr. 2017, Art. no. 150503.

[36] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018, *arXiv:1802.06002v2*. [Online]. Available: https://arxiv.org/abs/1802.06002v2

[37] A. Harrow and J. Napp, "Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms," *Phys. Rev. Lett.*, vol. 126, no. 14, Apr. 2021, Art. no. 140502.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980v9*. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[39] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: https://arxiv.org/abs/1609.04747

[40] C. Hempel *et al.*, "Quantum chemistry calculations on a trapped-ion quantum simulator," *Phys. Rev. X*, vol. 8, no. 3, Jul. 2018, Art. no. 031022.

[41] Y. Shen, X. Zhang, S. Zhang, J.-N. Zhang, M.-H. Yung, and K. Kim, "Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure," *Phys. Rev. A, Gen. Phys.*, vol. 95, no. 2, Feb. 2017, Art. no. 020501.

[42] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, vol. 54. Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[43] Q. Xu, X. Tan, R. Huang, and M. Li, "Verification of blind quantum computation with entanglement witnesses," *Phys. Rev. A, Gen. Phys.*, vol. 104, no. 4, Oct. 2021, Art. no. 042412.

[44] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, p. 4812, Nov. 2018.

Rui Huang received the B.E. degree from Huizhou University, Huizhou, China, in 2015, and the M.E. degree from the Guilin University of Technology, Guilin, China, in 2017. He is currently pursuing the Ph.D. degree with Jinan University, Guangzhou, China.

His research interests include secure quantum computing and quantum machine learning.

Xiaoqing Tan received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2004.

She is currently a Professor with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou, China. Her research interests include secure quantum computing and quantum machine learning.

Qingshan Xu received the B.E. degree from China Jiliang University, Hangzhou, China, in 2018, and the M.Sc. degree from Jinan University, Guangzhou, China, in 2021, where he is currently pursuing the Ph.D. degree.

His research interest includes secure quantum computing.