Adaptive diversity-based quantum circuit architecture search

Yuhan Huang [1,3], Siyuan Jin [2,3], Bei Zeng,[4,*] and Qiming Shao [1,4,5,†]

[1]*Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, 999077 Hong Kong, China*
[2]*Department of Information Systems, The Hong Kong University of Science and Technology, 999077 Hong Kong, China*
[3]*HSBC: HSBC Lab, HSBC holdings, Hong Kong SAR, China*
[4]*Department of Physics, The Hong Kong University of Science and Technology, 999077 Hong Kong, China*
[5]*IAS Center for Quantum Technologies, The Hong Kong University of Science and Technology, 999077 Hong Kong, China*

(Received 5 September 2023; accepted 30 May 2024; published 8 July 2024)

Quantum variational algorithms (VQAs) are highly promising to realize quantum advantages on near-term quantum devices. Existing VQAs based on a manually fixed quantum *Ansatz* are computationally inefficient due to noise and the limited coupling maps of these devices. Previous work considers various quantum architecture search (QAS) algorithms to autodesign a quantum *Ansatz* based on specific questions to improve the performance of VQAs. Compared to manual design, autodesign can more efficiently explore the large space of a possible *Ansatz* and achieve better performance. However, two main challenges in utilizing QAS to design quantum circuits efficiently are the tremendous amount of space required for candidate quantum circuits, and the disconnection between quantum devices and autodesign in terms of qubit mapping and quantum noise. To address these issues, we propose an adaptive diversity-based quantum *Ansatz* search algorithm to efficiently generate the optimal quantum circuit based on device qubit mapping and noise. By considering the diversity among different candidate circuits and adaptively adding circuit depths, our approach only needs to focus on a small optimization space at each iteration step. In addition, the synchronization of optimizing circuit structure and aligning qubit mapping enables us to generate quantum circuits while avoiding additional mapping overhead. We evaluate the performance of our algorithm on simulators and real quantum devices for quantum eigenvalue problems and classification tasks. Results demonstrate that quantum circuits generated by our method outperform both a fixed hardware-efficient *Ansatz* and randomly generated quantum circuits in terms of final performance and resource-saving. Our algorithm provides a flexible way to efficiently generate excellent quantum circuits for significantly improving the performances of VQAs on near-term quantum devices.

DOI: 10.1103/PhysRevResearch.6.033033

I. INTRODUCTION

Quantum computing has had dramatic progress in recent decades. Starting with Shor's [1] breakthrough and continuing with Grover's algorithm [2] to quantum supremacy [3], it is clear that quantum mechanics offers an apparent advantage for computing. A fully fault-tolerant quantum computer can achieve high computing power, accuracy, and reliability in performing computations. However, since constructing a fully fault-tolerant quantum computer is still a challenge at present, researchers have focused on developing and utilizing devices known as noisy intermediate-scale quantum (NISQ) [4] computers, which are smaller and less reliable than a fully fault-tolerant computer but are still capable of performing certain quantum computations.

Quantum-classical hybrid algorithms [5,6] are the most promising near-term algorithms for achieving a quantum advantage in NISQ devices. Some hybrid algorithms, such as variational quantum classifiers (VQCs) [7], variational quantum eigensolvers (VQEs) [8], and quantum approximation optimization algorithms (QAOAs) [9], are proposed for different scenarios, such as a VQC for credit card fraud detection analysis [10,11] and a VQE for quantum chemistry [12]. However, these algorithms still face certain challenges under near-term quantum devices, such as the fixed coupling map of physical particles, which limits the connectivity of quantum gates, and quantum noise, which destroys the results of algorithms. While general quantum circuits, such as a hardware-efficient *Ansatz* [13] and ZX-XX layout [14], have been proposed, designing specific quantum circuits suitable for near-term quantum devices plays the primary role in implementing efficient and high-performance quantum-classical hybrid algorithms.

There has been a significant amount of research on neural network architecture search algorithms for designing high-performance and low-resource neural networks. These classical architecture search algorithms have considered a variety of factors, such as the limits of the search space [15,16], efficient search strategies [17–19], and performance estimation [20].

*Contact author: zengb@ust.hk
†Contact author: eeqshao@ust.hk

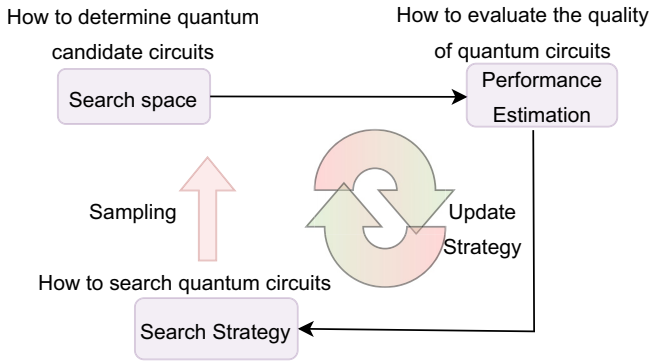

FIG. 1. The quantum architecture search algorithm considers three factors. One is the search space, which determines the set of quantum circuits that are candidates for the search. The second is the search strategy, which determines how to search through the candidate circuits efficiently. The last factor is performance estimation, which determines how to evaluate the quality of the quantum circuits.

Quantum architecture search (QAS) algorithms aim to find the optimal quantum circuit architecture for a given task. Similar to neural network architecture search algorithms, QAS mainly involves three different steps—search strategies, candidate solution space, and performance evaluation—which are shown in Fig. 1. In previous work, a variety of search strategies have been proposed, including evolutionary learning [21–26], reinforcement learning [27,28], the differentiable quantum architecture search algorithm [29], and the Monte Carlo tree searching algorithm [30]. The candidate solution space is usually composed of different permutations of a set of quantum gates, which are determined by specific quantum devices [31]. Usually, the performance estimation of candidate quantum circuits is dependent on the final performance of the algorithm and quantum operator resource requirements. However, there are two primary challenges in QAS: (i) enough shallow quantum circuits must be generated to prevent the results of the algorithms from being destroyed due to the accumulation of quantum noise; and (ii) an efficient search approach must be designed to handle the tremendous amount of solution space.

To address these challenges, this work proposes an adaptive diversity-based quantum architecture search (ADQAS) algorithm that updates the search strategy and restricts the search space to enhance algorithm performance and efficiency. In particular, we adopted an iterative strategy, gradually increasing the number of quantum gates to generate shallow circuits, in order to mitigate the effects of quantum noise accumulation. Further, we consider the combination of qubit mapping and circuit design directly in the search strategy, which can avoid the extra overhead when implementing the generated quantum circuits on a specific device such as the IBM quantum devices Lima, Manila, and Nairobi. In addition, we present an effective approach, a diversity-guided search space, to increase the structural diversity of candidate solutions by calculating the maximum quantum architecture entropy, which can prevent our algorithm from falling into a trap. In the performance estimation stage, the average energy, accuracy, and recall serve as the metrics for evaluating the algorithm.

Here, we consider two different application scenarios of our algorithm, namely variational quantum eigenvalue problems and classification tasks. For the six-qubit VQE task, the adaptive diversity-based quantum architecture search algorithm reduces the classical resources (parameters) and quantum resources (quantum gates) of the generated circuit by 28% and 23%, respectively, compared to a hardware-efficient *Ansatz* (HEA), when subjected to depolarizing noise. Additionally, it achieves lower average energy compared to the HEA under depolarizing noise. Furthermore, the ADQAS algorithm reduces the sampling resources, achieves lower average energy, and takes nearly 100 000 seconds less time than the adaptive random-based quantum architecture search (ARQAS) algorithm. For the four-qubit VQE task, the ADQAS algorithm also reduces the quantum and classical resources and demonstrates better performance compared to the HEA. It also requires less time than the ARQAS algorithm when subjected to the Lima-Simulator and IBM-Lima device.

For the Iris VQC task, the ADQAS algorithm reduces classical and quantum resources of the generated circuit by 75% and 73%, respectively, compared to HEA, when subjected to depolarizing noise. Additionally, it achieves higher average accuracy compared to the random-based quantum architecture search (RQAS) algorithm under depolarizing noise. Furthermore, the circuits generated by the ADQAS algorithm achieve 81% accuracy and 86% recall ratio on the Iris data set and the credit card fraud detection data set in IBM-Lima.

II. ADAPTIVE DIVERSITY-BASED QUANTUM ARCHITECTURE SEARCH ALGORITHM

The quantum circuit in most quantum algorithms does not consider real physical device environments such as quantum noise and qubit mapping. As a result, the performance of quantum algorithms often falls short of expectations. In this section, we present the ADQAS algorithm, which addresses these challenges by considering the actual noise characteristics of physical devices and their coupling map when designing circuits automatically. For instance, we apply the ADQAS algorithm in the real quantum device Lima, taking into account its device noises such as dephasing and decoherence. In addition, the circuit autodesigned by our algorithm follows the T-shaped qubit connection in Lima as the device's coupling map. By incorporating these factors, the ADQAS algorithm can automatically design more effective circuits, leading to improved performance.

In this study, we propose an adaptive diversity-based quantum architecture search algorithm that aims to improve the performance of the quantum architecture search by using an adaptive algorithm and to improve the efficiency of the search by reducing the search space. The ADQAS algorithm has four stages: the initial stage, the search space sampling stage, the performance and estimation stage, and the adaptive stage, which is shown in Fig. 2. Our algorithm also uses a diversity-guided algorithm, as shown in Fig. 2(b), to limit the search space to a subset of the random-based search space depicted in Figs. 2(c) and 2(d), which samples more evenly and increases the efficiency of the QAS algorithm.

In the initial stage, the ADQAS generates the whole search space with $(\#\text{Rotation}^{n\text{qubits}} * 2^{\#\text{CNOT}})^L$ noisy quantum

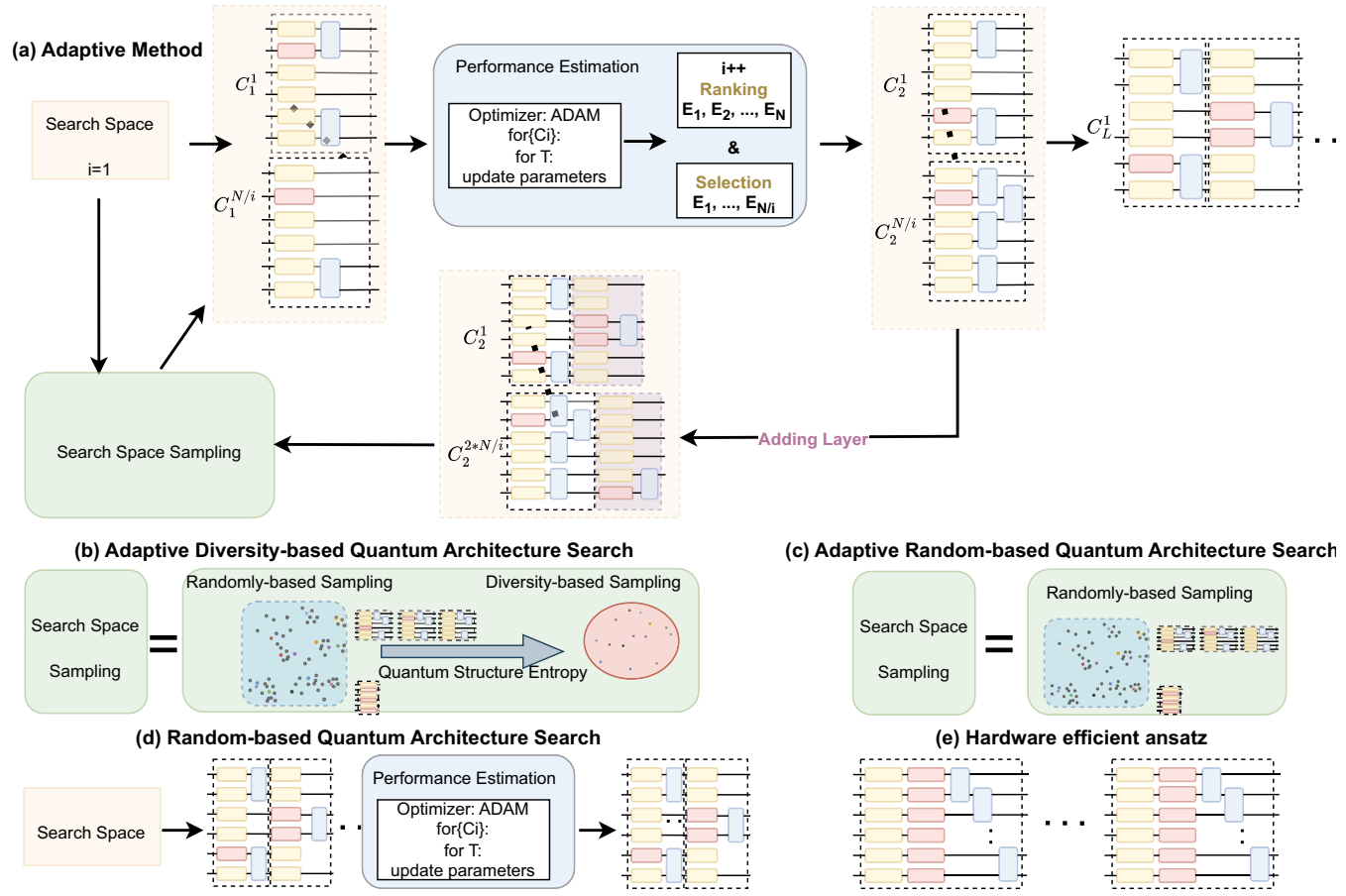

FIG. 2. The ADQAS algorithm and the ARQAS algorithm follow an iterative process. In each iteration i, the algorithms consist of three stages: (1) search space sampling, (2) performance estimation, and (3) adaptation. (a) An adaptive method employed in ADQAS and ARQAS. (b) In ADQAS, at the start of iteration i, search space sampling narrows the circuit search space to a set $C_i^1, \ldots, C_i^{N/i}$ of N/i candidate quantum circuits using quantum architecture entropy. Each C_i^j represents the jth candidate circuit in iteration i. In the performance estimation stage, the top $N/(i+1)$ circuits with the best-estimated performance are retained. Finally, in the adaptive stage, ADQAS adds another layer of complexity to the algorithm. After L generations, the output of the algorithm is the optimal circuits. (c) ARQAS chooses candidate circuits at random in the space sampling stage and uses the same performance estimation and adaptation stages as ADQAS. (d) The RQAS algorithm randomly chooses a quantum circuit from the search space and uses the ADAM optimizer to optimize it. (e) The HEA with R_y, R_z, and CNOT gates as a general set of quantum gates to construct quantum circuits.

circuits with L blocks, where #Rotation refers to the number of noisy quantum rotation gates, and #CNOT refers to the number of noisy CNOT gates. In addition, each block of these noisy quantum circuits consists of a layer of single-qubit gates (the red and yellow rectangles present different basis single-qubit gates) and a layer of two-qubit gates (the blue rectangles present two-qubit gates CNOT, named CX). The CNOT gate connectivity adheres to the specific coupling map of the physical device.

In the search space sampling stage, the ADQAS initializes the candidate noisy quantum circuits and randomly samples M noisy quantum circuits (in the dark blue box) from the whole space. The ADQAS then selects the candidate noisy quantum circuits, called diversity-guided circuits (in red ellipse), from M to N according to the quantum architecture entropy, which ensures the diversity of quantum circuit structures. In a later subsection, we will provide more details on the definition of quantum architecture entropy and how to use it to narrow the search space.

In the performance and estimation stage, the ADQAS optimizes the diversity-guided circuits using an ADAM optimizer. To ensure the efficiency of the ADQAS, the algorithm only optimizes all trainable candidate quantum circuits a few times. To this end, even though the size of quantum circuit space scales in terms of n qubits and L layers, ADQAS can guarantee its applicability toward large-scale problems. To select quantum circuits for the next generation, the trained quantum circuits are ranked based on their performance under the noised simulator, and the top first N/i circuits that are less affected by noise and with low average energy are chosen. We will provide more details on the various performance metrics used in different cases in a later subsection.

In the adaptive stage, the top circuits use the adaptive algorithm to expand the number of blocks to obtain the adapted noisy circuits. Then the adapted noisy circuits use the diversity-guided sampling strategy again. The process is repeated until convergence of the ADQAS. Simultaneously, the

optimal noisy quantum circuit is generated by the ADQAS, which has the best performance under the real physical device.

After identifying the optimal quantum circuit architecture, we combine gates of the same type and remove any nonfunctional quantum gates, and we reoptimize parameters. We will provide more details on the specific process of the adaptive algorithm and how it is used in the ADQAS algorithm in a later subsection.

A. Search space sampling

Improving the efficiency of algorithms in quantum circuit design can reduce the resources and time needed to run the algorithm. One approach is to use ADQAS, which leverages the architecture entropy of quantum circuits to diversify the architecture of candidate quantum circuits. This approach is inspired by the concept of architecture entropy in neural networks [32]. Here, we present three definitions according to architecture entropy in quantum circuits.

The Levenshtein distance (LD) is a measure of the distance between two strings. It is defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another. To calculate the LD between two strings, we can use the following recursive formula:

$$\mathrm{d}_{i,j} = \begin{cases} \mathrm{d}_{i-1,j-1}, & s[i] = t[j], \\ \min(\mathrm{d}_{i-1,j}, \mathrm{d}_{i,j-1}, \mathrm{d}_{i-1,j-1})+1 & \text{otherwise}, \end{cases}$$

where s and t are input strings, and $\mathrm{d}_{i,j}$ is the LD between the substrings $s[0:i]$ and $t[0:j]$. In the context of quantum circuits, the LD can be used to measure the differences between two quantum circuits.

To explore a variety of quantum circuit structures, we establish a notion of the difference between two quantum circuits. In a quantum architecture search, the difference is quantified using the weighted Levenshtein distance (WLD), an extension of the Levenshtein distance. The WLD takes into account the different types of quantum gates and assigns different weights to them based on their relative importance. The definition is as follows:

Definition 1. The weighted Levenshtein distance of quantum circuits is $D(i,j) = \sum_{k=1}^{n\text{layers}} W_R * f_{k,R}(i,j) + W_C * f_{k,C}(i,j)$.

In this definition, $D(i,j)$ represents the WLD between quantum circuits C_i and C_j, $n$layers is the number of layers in the quantum circuit, and $f_{k,R}(i,j)$ and $f_{k,C}(i,j)$ represent the LD between the signal- and two-qubits gates, respectively, of layer k in quantum circuits C_i and C_j. The factors W_R and W_C are the distance weights assigned to the signal- and two-qubit gates, respectively.

To identify candidate quantum circuits with similar structures, we partition quantum circuits into distinct subgroups. Each subgroup $\mathbb{S}_1, \ldots, \mathbb{S}_Q$ consists of quantum circuits that share similar structural characteristics. We define a criterion for assigning a given quantum circuit C_i to the kth subgroup $\mathbb{S}_k$ by utilizing the WLD, a measure of the difference between two quantum circuits.

Definition 2. The criterion for assigning a quantum circuit C_i to the kth subgroup $\mathbb{S}_k$ is based on the averaged WLD between C_i and every circuit $C_{k_j} \in \mathbb{S}_k$. If the averaged WLD is less than or equal to a predetermined threshold value x, C_i is placed in $\mathbb{S}_k$.

C_{k_j} represents the jth quantum circuit belonging to the kth subgroup. Opting for a lower value of x means that circuits need to exhibit a higher level of similarity to be placed in the same subgroup. However, this can result in reduced efficiency. In our study, the choice of x is based on empirical experience, allowing us to strike a balance between precision and efficiency.

After choosing a suitable x, the candidate quantum circuits are divided into different subgroups. Therefore, more diverse quantum circuits can be formed by selecting quantum circuits from different subgroups.

Definition 3. The architecture entropy of quantum circuits is a measure of the diversity of candidate quantum circuits, which is defined as $E_{\mathrm{QAE}} = -\sum_{q=1}^{Q} p_q \log(p_q)$.

E_{QAE} is the architecture entropy, Q is the number of subgroups of candidate quantum circuits, and $p_q = |\mathbb{S}_q|/N$ is the proportion of quantum circuits in subgroup q, where $|\mathbb{S}_q|$ is the number of quantum circuits in subgroup q, $|\cdot|$ presents the number of quantum circuits of the group $\mathbb{S}_q$, and N is the total number of candidate quantum circuits. The value of the architecture entropy belongs to the range $[0, \log N]$. A lower value indicates a lower degree of diversity, and a higher value indicates a higher degree of diversity. The architecture entropy can be used to evaluate the diversity of candidate quantum circuits and optimize their structure by increasing their diversity. It can also be used as a criterion for selecting quantum circuits in ADQAS.

The diversity-guided sample algorithm works for narrowing the search space of ADQAS using the quantum architecture entropy. It involves generating a random-based sample of M candidate quantum circuits and then selecting a diversity-guided sample of N quantum circuits from the random-based sample based on the maximum architectural entropy of the candidate quantum circuits. The value of M is typically greater than N, and there are a total of C_M^N possible combinations of N quantum circuits that can be selected from the random-based sample of M quantum circuits. To improve the efficiency of the algorithm, ADQAS can repeat the selection process $M-N$ times, each time deleting a single quantum circuit from the random-based sample and selecting a new quantum circuit based on the maximum architectural entropy. This helps to reduce redundancy in the random-based sample and improve the diversity of the selected quantum circuits.

B. Adaptive strategy

The random sampling algorithm is commonly used for searching for the optimal solution, but it can be inefficient due to its large variance in performance. To address this issue, Adaptive VQE [33] adds quantum gates one by one based on the derivative of the *Ansatz* to improve quantum architecture search performance. However, this approach can be resource-intensive because of the need to add gates individually.

Adaptive diversity-based quantum architecture search can address this issue by adding quantum layers one by one, using the adaptive layer concept to improve quantum circuits' performance while saving computing resources. In the ADQAS algorithm, N quantum circuits are initially sampled and

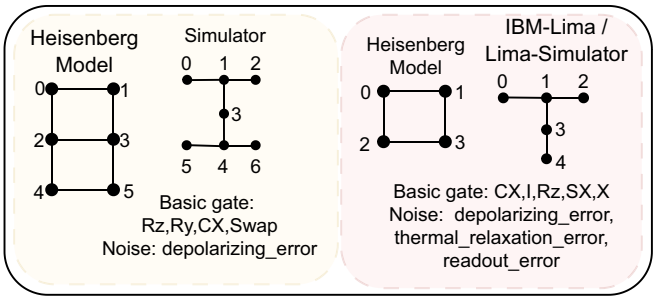

FIG. 3. There are two main types of quantum devices: Simulator and IBM-Lima. Both Simulator and IBM-Lima have a coupling map, which specifies the connectivity of the qubits, and a set of basis quantum gates, which are the basic building blocks for constructing quantum circuits. They may also have a noise model, which describes the sources of noise that can affect the device. Next to devices are the VQE problems corresponding to different devices.

optimized using an optimizer such as ADAM. After T iterations of optimization, the quantum circuits' performance is ranked, and the top N/i performing circuits are chosen as a set of candidate quantum circuits for the next generation, which is shown in Fig. 2(a).

In the next generation, the previous layer is fixed, and the search space in ADQAS is reduced. This process continues until the optimal circuits with L layers are obtained. The search space's reduction helps optimize the circuit search while still guaranteeing an acceptable performance level.

In this stage, the open system needs to be considered. In IBM's quantum cloud, the real physical systems, such as Lima, Manila, and Nairobi, are open systems with noise channels that continuously interact with the environment and have qubit layouts. This work considers two types of devices: a quantum simulator with depolarizing noise and an IBM-Lima with gate noises (depolarizing error and amplitude damping error) and measurement noises.

The quantum simulator has a depolarizing noise model given by

$$\epsilon(\rho) = \frac{pI}{2} + (1-p)\rho, \tag{1}$$

where ρ is the density matrix that represents the quantum state, and p is the probability that the qubit is depolarized. The qubit layout of the quantum simulator is shown in Fig. 3, Simulator.

The IBM-Lima not only has a depolarizing error, but it also has an amplitude damping error described by a Kraus form:

$$\begin{aligned} \epsilon(\rho) &= E_0(\rho)E_0^\dagger + E_1\rho E_1^\dagger, \\ E_0 &= |0\rangle\langle 0| + \sqrt{1-\lambda}|1\rangle\langle 1|, \\ E_1 &= \sqrt{\lambda}|0\rangle\langle 1|, \end{aligned} \tag{2}$$

where λ is the probability of the qubit relaxing; the IBM-Lima also has measurement noises and a "T" qubit layout, as shown in Fig. 3, IBM-Lima.

C. Performance estimation

The performance estimation depends on the different tasks, such as variational quantum eigensolver and variational quantum classification.

1. Variational quantum eigensolver

The variational quantum algorithm is a quantum-classical hybrid algorithm for solving the quantum many-body ground energy problem in the NISQ era. The essence of its mathematics is the Rayleigh-Ritz algorithm [34], which minimizes the value of the function by varying the parameters of the wave function:

$$E(x) = \min_x \frac{\langle\phi_0|U^\dagger(x)HU(x)|\phi_0\rangle}{\langle\phi_0|U^\dagger(x)U(x)|\phi_0\rangle}. \tag{3}$$

In a quantum processing unit (QPU), the initial state $|\phi_0\rangle$ is generally initialized $|000\cdots0\rangle$ in the VQE algorithm. Then, through the unitary evolution $U(x)$, the final state is evaluated as $|\phi_x\rangle$. Next, through the measurement operation, the classical result is $\langle\phi_x|H|\phi_x\rangle$. Finally, VQE puts the classical result into the optimizer in a central processing unit (CPU), such as ADAM, or gradient descent, to optimize the parameters x of the quantum evolution. Then it returns to the QPU for evolution.

2. Variational quantum classifier

A variational quantum classifier, which belongs to quantum supervised learning, is another quantum-classical hybrid algorithm to deal with classification problems. It has a learning process to learn the existing observations, and a predicting process to estimate the new observations.

The variational quantum classifier assumes the existing observations (data set) are $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ is the vector of data features, and $y_i \in \{0, \ldots, k\}$ is the data label, which means the data $\boldsymbol{x}_i$ belong to class y_i. VQC can encode the data features $\boldsymbol{x}_i$ into a quantum state $|\boldsymbol{x}_i\rangle$ by using the amplitude encoding method [35] or the angle encoding method [14]. After data encoding, VQC designs a quantum classifier $U(\theta)$ to evaluate. Note that the data encoding and evolution process $U(\theta)$ can be viewed together as a classifier $G(\boldsymbol{x};\theta)$. Then, according to the total of classes k, VQC uses a measurement operator with $\log K$-dimension. According to the measurement result, the data set can be classified into y_i'. Next, using a classical optimizer to optimize the parameters θ,

$$L(\theta) = \sum -y_i \log\left(\langle\phi(\boldsymbol{x}_i,\theta)|O|\phi(\boldsymbol{x}_i,\theta)\rangle\right), \tag{4}$$

where $|\phi(\boldsymbol{x}_i,\theta)\rangle = G(\boldsymbol{x}_i;\theta)|0\rangle$. Then, VQC is evolved again by classical parameters until the VQC converges. After training the classifier with a parametrized quantum circuit $G(\boldsymbol{x};\theta^*)$ in existing observations, the VQC could predict the class of new observations.

In the following two sections, we will present the results of using the simulator and IBM-Lima device for VQE and VQC problems. The training was performed using an AMD Ryzen 9 5950X CPU with 16 cores.

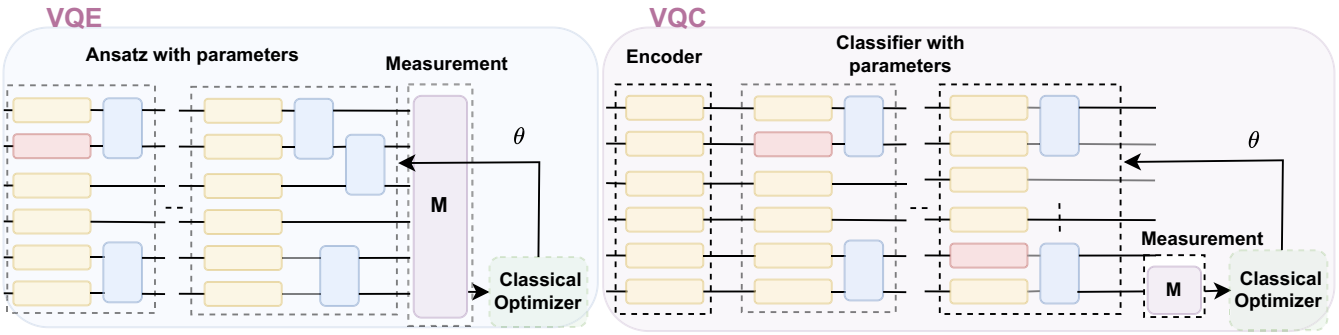

FIG. 4. The VQE and VQC are algorithms that utilize the concept of quantum-classical hybridization to solve problems by combining a quantum computer with a classical computer. The VQE consists of two quantum blocks (*Ansatz* and measurement) and a classical optimizer. Each *Ansatz* block includes one layer of single-qubit gates, which are parametrized by variables that can be modified by the classical optimizer, and one layer of two-qubit gates such as CNOT gates. The measurement block measures the output of the *Ansatz* block, and the classical optimizer adjusts the parameters of the *Ansatz* block to improve the approximation of the ground energy through iterative adjustment. The VQC comprises three quantum blocks (encoder, *Ansatz*, and measurement) and a classical optimizer. The encoder block loads the data into the quantum circuit, and the *Ansatz* block processes the data with a parametrized quantum circuit. The measurement block measures the output of the *Ansatz* block, and the classical optimizer adjusts the parameters of the *Ansatz* block to increase classification accuracy.

III. RESULTS FOR VQE PROBLEMS

In this section, we consider the variational quantum eigensolver, as shown in Fig. 4, under both the Simulator and the IBM-Lima device. We aim to solve the ground energy of the two-dimensional (2D) lattice of a six-qubit and a four-qubit Heisenberg system. The Heisenberg model is a mathematical model of a system of interacting spin-1/2 particles, widely used to describe the behavior of magnetic materials. To demonstrate the efficiency of the ADQAS, in Figs. 2(a) and 2(b), for designing optimal quantum circuits with high performance and low resource requirements, we conducted a comparison with three other algorithms: the HEA shown in Fig. 2(e), the RQAS shown in Fig. 2(d), and the ARQAS shown in Fig. 2(c). The HEA [13], with R_y, R_z and CNOT gates, is a general quantum circuit that can be used for both VQE and VQC problems, and it is designed to be efficient in terms of its use of quantum hardware resources. The RQAS algorithm as shown in Fig. 2(d) only uses a random sampling algorithm to solve VQE problems. The ARQAS as shown in Figs. 2(a) and 2(c) is similar to the RQAS, but it includes the adaptive algorithm in addition to the random sampling algorithm. Later, we will present the simulation and real results in these algorithms for VQE problems.

In the ADQAS, the basis gates, the qubits connection, and the noise model are used to depend on the type of device being used. For the simulator (shown in Fig. 3, Simulator), the basis gates are the rotation gates R_y, R_z, and the two-qubit gates CNOT, SWAP, and the noise model used is the depolarizing noise model. The depolarizing noise of single-qubit gates is set to 0.001, the depolarizing noise of CNOT is set to 0.05, and the depolarizing noise of SWAP is set to 0.15. The coupling map for the simulator is based on the topological structure of the device (0-1, 1-2, 1-3, 3-4, 4-5, 4-6). For the IBM-Lima (shown in Fig. 3, Lima), the basis gates are the single-qubit gates I, R_z, Sqrt of X (SX), X, and the two-qubit gates CNOT, and the noise model used is a combination of depolarizing noise, thermal relaxation error, and read-out error. The coupling map for the IBM-Lima is based on the topological structure of the device (0-1, 1-2, 1-3, 3-4, 4-5).

The ADQAS uses different initial sample sizes depending on the problem. For the six-qubit Heisenberg model, it initially chooses $N\sim200$ circuits in the whole quantum circuit search space and uses 100 times diversity-guided sampling. For the four-qubit Heisenberg system, the initial sample size is reduced to $N\sim60$ circuits. It selects the best-performing circuits and adds layers to them until the best circuit is found and trains the candidate quantum circuits with the ADAM optimization algorithm with a learning rate of $\eta = 0.05$, the exponential decay rate for the first moment estimation of $\beta_1 = 0.9$, the exponential decay rate for the second-moment estimation of $\beta_2 = 0.99$, and a small value for numerical stability of $\epsilon = 10^{-8}$. According to the performance of candidate quantum circuits, in the ith generation, ADQAS selects the first N/i well-performance quantum circuits. Each selected candidate quantum circuit is added with two kinds of quantum circuits with one layer. Then, the ADQAS repeats the process of narrowing the space, training, selecting, and adding layers until it finds the quantum circuit with the best performance.

Variational quantum eigensolver is a quantum algorithm that uses a quantum computer to find the ground energy of a given Hamiltonian, such as the Heisenberg model. The key idea behind VQE is to use a quantum computer to prepare a quantum state $|\phi(x)\rangle = U(x)|\phi_0\rangle$ that encodes information about the system being studied and then measure the energy of the system $\langle\phi_0|U^\dagger(x)HU(x)|\phi_0\rangle$, and then use classical optimization techniques to adjust the parameters x of the quantum state in order to minimize the energy of the system.

Heisenberg model: The Heisenberg model is a model of a system of spin-1/2 particles, such as electrons or protons, that interact with each other on a two-dimensional lattice. The Hamiltonian describes the model

$$H = \sum_{i,j} J_{i,j}\boldsymbol{\sigma}^i \cdot \boldsymbol{\sigma}^j, \tag{5}$$

TABLE I. The comparison of the classical resources, quantum resources, final performance, and running time (in seconds) of ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under a depolarizing noise simulator for the six-qubit Heisenberg model.

Quantum Circuit	N of R_s	N of CX	Average energy via simulator	Time (s)
ADQAS-C*	43	22	−11.406256	270259
ARQAS-C*	45	18	−11.49023	373647
RQAS-C*	45	25	−10.9238	68466
HEA	60	25	−11.25390	55050

where $J_{i,j} > 0$ is the coupling strength between the ith and jth particles, and σ is the Pauli operator $\boldsymbol{\sigma}^i = (\sigma_x^i, \sigma_y^i, \sigma_z^i)$. In the next two subsections, we demonstrate experimental results for different eigenvalue solving tasks, with one subsection presenting results on a simulator and the other on a real quantum device.

A. Simulator

In the Simulator, the basic operation pool is set as R_y, R_z, CNOT. Depolarizing noise is introduced, with a strength of 0.0001 for single-qubit depolarizing noise and a strength of 0.005 for two-qubit depolarizing noise.

The Hamiltonian of the 2D lattice of six qubits can be rewritten as shown:

$$\begin{aligned} H = {} & X_0X_1 + Y_0Y_1 + Z_0Z_1 + X_0X_2 + Y_0Y_2 + Z_0Z_2 \\ & + X_1X_3 + Y_1Y_3 + Z_1Z_3 + X_2X_3 + Y_2Y_3 + Z_2Z_3 \\ & + X_2X_4 + Y_2Y_4 + Z_2Z_4 + X_3X_5 + Y_3Y_5 + Z_3Z_5 \\ & + X_4X_5 + Y_4Y_5 + Z_4Z_5. \end{aligned} \tag{6}$$

1. The best quantum circuit generated by ADQAS outperforms other algorithms

To evaluate the performance of the best quantum circuit (ADQAS-C*) generated by ADQAS, we compare it to the best quantum circuits (ARQAS-C*, RQAS-C*, and HEA) generated by other algorithms such as ARQAS, RQAS, and the hardware-efficient *Ansatz*-based VQE under a depolarizing noise environment. Table I presents a comparison of these algorithms in terms of their resource usage and performance for solving the 2D Heisenberg problem. It includes information on classical resources (number of parameters in terms of R_s gates), quantum resources (number of quantum gates in terms of R_s gates and CX gates), final performance (average energy), and running time under the depolarizing noise environment.

Table I illustrates that the ADQAS-C* uses similar resources to the ARQAS-C* and RQAS-C*. However, it utilizes fewer classical and quantum resources than the hardware-efficient *Ansatz* under a depolarizing noise environment. In particular, ADQAS-C* reduces resource usage by 28% and 23% when compared to the HEA. Additionally, the performance of ADQAS-C* and ARQAS-C* is similar and better than that of RQAS-C* and HEA. Significantly, ADQAS-C* outperforms the HEA, achieving a final energy of −11.406 256, which is relatively close to the exact ground energy of −12.5175. Although ADQAS-C* and ARQAS-C* have similar resource usage and performance, the running time for ADQAS-C* is also shorter than that of ARQAS-C*. Overall, these results demonstrate the effectiveness of the ADQAS algorithm in achieving high performance with low resource requirements under the depolarizing noise environment.

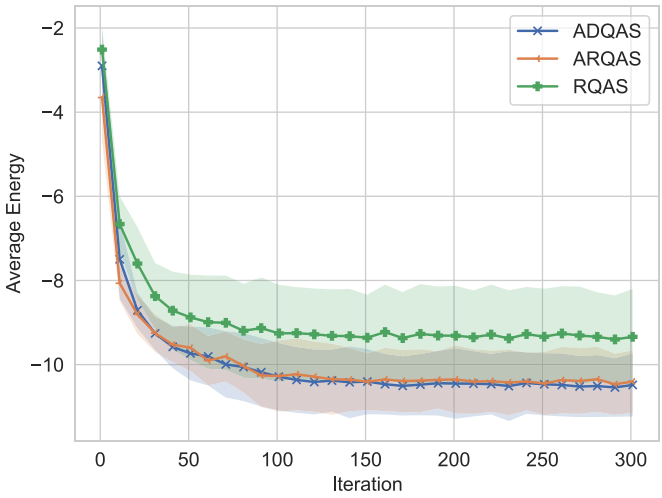

FIG. 5. The average performance of quantum circuits generated by ADQAS, ARQAS, and RQAS for solving the 2D Heisenberg problem.

2. Average performances of quantum circuits generated by ADQAS outperform RQAS algorithms

To evaluate the average performance of ADQAS, ARQAS, and RQAS, all algorithms were run with the same set of hyperparameters. In Fig. 5, we show the mean and variance of the average energy of the average 10 circuits generated by ADQAS, ARQAS, and RQAS. The results indicate that the average performance of quantum circuits generated by ADQAS and ARQAS outperforms RQAS, with a lower average energy and smaller error bar.

B. IBM-Lima

To further analyze the performance of the ADQAS, we solve the 2D Heisenberg problem with four qubits under an IBM real device. In this case, the 2D square lattice of the 4-qubits Hamiltonian model is shown as

$$\begin{aligned} H = {} & X_0X_1 + Y_0Y_1 + Z_0Z_1 + X_0X_2 + Y_0Y_2 + Z_0Z_2 \\ & + X_1X_3 + Y_1Y_3 + Z_1Z_3 + X_2X_3 + Y_2Y_3 + Z_2Z_3. \end{aligned} \tag{7}$$

The search space of candidate quantum circuits with the basis gates (I, R_z, SX, X, and CNOT) is too large. To reduce the search space, we reconstruct the basic gates to be R_y, R_z, and CNOT. This is done by expressing the gates $SX \cdot R_z(\theta) \cdot SX \cdot X$ as $R_y(-\theta)$, as shown in Eq. (8),

$$\begin{aligned} SX \cdot R_z(\theta) \cdot SX \cdot X &= e^{-i\frac{\theta}{2} SX \cdot Z \cdot SX \cdot X} \\ &= e^{iY\frac{\theta}{2}} \\ &= R_y(-\theta). \end{aligned} \tag{8}$$

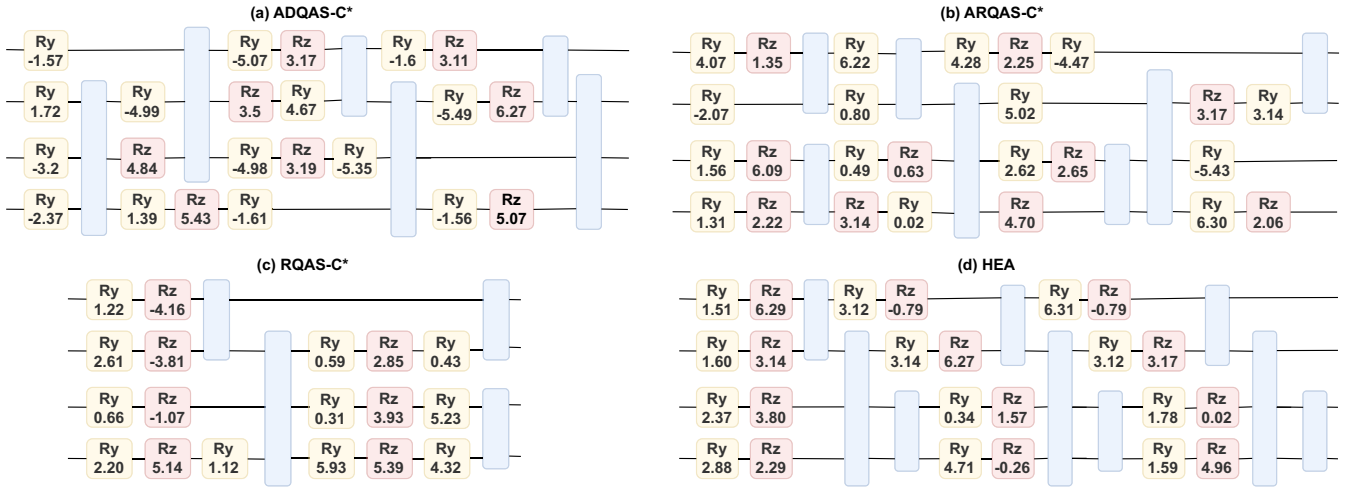

FIG. 6. The quantum circuits generated by ADQAS, ARQAS, RQAS, and HEA-based algorithm. (a) ADQAS-C* under Lima-Simulator noise model for a four-qubit Heisenberg system. (b) ARQAS-C* under a Lima-Simulator noise model for a four-qubit Heisenberg system. (c) RQAS-C* under the Lima-Simulator noise model for a four-qubit Heisenberg system. (d) HEA under the Lima-Simulator noise model for a four-qubit Heisenberg system.

1. Best quantum circuit generated by adaptive diversity-based quantum architecture search algorithm outperforms other algorithms

To evaluate the performance of ADQAS, we generate ADQAS-C* using ADQAS under the Lima-Simulator and compare it to the ARQAS-C*, RQAS-C*, and HEA generated by other algorithms such as ARQAS, RQAS, and the hardware-efficient *Ansatz*-based VQE on both the Lima-Simulator and IBM-Lima devices. The exact ground energy of this four-qubit system is -8. In Fig. 6, we show the number of parameters and the number of quantum gates. In Table II, we present the performance of ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA for solving the 2D Heisenberg problem under the Lima-Simulator and IBM-Lima device, as well as the running time for each algorithm under the Lima-Simulator. The results indicate that ADQAS-C* uses resources similar to ARQAS-C*, but uses fewer classical and quantum resources than HEA under the Lima-Simulator and IBM-Lima device. Additionally, ADQAS-C* and ARQAS-C* have similar performances and outperform RQAS-C* and HEA. Although ADQAS-C* and ARQAS-C* have similar resource usage and performance, ADQAS-C* has a shorter running time than ARQAS-C*. Overall, these results demonstrate the effectiveness of the ADQAS algorithm in achieving high performance with low resource requirements under both the Lima-Simulator and IBM-Lima.

TABLE II. The comparison of final performance and running time (in seconds) between ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the Lima-Simulator and IBM-Lima for a four-qubit Heisenberg system.

Quantum circuit	Average energy via Lima-Simulator	Average energy via IBM-Lima	Time (s)
ADQAS-C*	-6.53	-5.67	64295
ARQAS-C*	-7.08	-5.55	99154
RQAS-C*	-5.89	-5.01	49706
HEA	-6.07	-5.12	46529

2. Average performances of quantum circuits generated by ADQAS algorithm outperform RQAS algorithms

To evaluate the average performance of ADQAS, ARQAS, and RQAS, all algorithms were run with the same set of hyperparameters. The results, which are presented in Table III, show the average energy of the average 10 circuits generated by ADQAS ($\overline{\text{ADQAS-C}}$), ARQAS ($\overline{\text{ARQAS-C}}$), and RQAS($\overline{\text{RQAS-C}}$) on both IBM-Lima quantum simulator and IBM-Lima. It can be observed that the circuits generated by ADQAS and ARQAS have a lower average energy, indicating better performance.

IV. RESULTS FOR VQC PROBLEMS

In this section, we consider the variational quantum classifier, as shown in Fig. 4, VQC, under both the Simulator and the IBM-Lima devices. In this application, the goal is to use the variational quantum algorithm to predict the species of iris flowers based on four features and to predict credit card fraud detection based on 28 features. These are examples of supervised learning tasks, where the goal is to learn a model that can predict the output (the species of the iris flower or whether a credit card transaction is fraudulent) based on a

TABLE III. The mean average performance comparison of quantum circuits generated by ADQAS, ARQAS, and RQAS under the Lima-Simulator and IBM-Lima for the 2D Heisenberg problem.

Quantum circuits	Mean average energy via Lima-Simulator	Mean average energy via IBM-Lima
$\overline{\text{ADQAS-C}}$	-5.65	-5.26
$\overline{\text{ARQAS-C}}$	-5.86	-5.21
$\overline{\text{RQAS-C}}$	-5.35	-4.47

set of input features (the characteristics of the iris flower or that of the credit card transaction). To demonstrate the efficiency of the ADQAS algorithm for designing optimal quantum circuits with high performance and low resource requirements, we conducted a comparison with three other algorithms: the HEA, the RQAS, and the ARQAS. Later, we will present the simulation and real results in these algorithms for VQC problems.

The variational quantum classifier shown in Fig. 4 is a machine-learning algorithm that uses a quantum computer to perform classification tasks. In a classification task, the goal is to assign a label y to an input data point $\boldsymbol{x}$ based on certain features or characteristics of that data point. The core principle behind VQC is the use of a quantum circuit, which is a sequence of quantum gates that manipulate the state of a quantum system. This circuit is designed to take input data and encode them into the state of the quantum system $|\boldsymbol{x}\rangle$ in the encoder block, and then use the parametrized classifier $U(\theta)$ to perform classification tasks. To train the quantum classifier, a classical optimization algorithm is used to adjust the parameters θ of the quantum circuit in order to minimize the classification error.

Classification problem: There are two applications of a variational quantum classifier: Iris classification and credit card fraud detection. The Iris classification application involves using a VQC to classify different species of iris flowers based on their sepal length, sepal width, petal length, and petal width. For this task, we have selected 100 data sets with two species of iris flowers, and the ratio of training to testing data sets is 1:1. The ADQAS algorithm is used to find a high-performing VQC that can accurately classify the different species of iris flowers.

Credit card fraud detection in finance uses trained VQC to detect genuine and fraudulent transactions. In this task, a data set of genuine and fraudulent transactions is used, with a ratio of 1:1 (100:100). The data set is preprocessed using principal component analysis to reduce the dimensionality from 28 to 4, as shown in Fig. 7. In the figure, purple points represent genuine transactions, and light blue points represent fraudulent transactions. Several indicators are defined in order to evaluate the performance of the VQC. True Negative (TN) is the number of times a fraudulent transaction is correctly identified as genuine. True Positive (TP) is the number of times that a fraudulent transaction is correctly identified as fraudulent. False Positive (FP) is the number of times that a genuine transaction is incorrectly identified as fraudulent. False Negative (FN) is the number of times that a genuine transaction is incorrectly identified as genuine. The accuracy is defined as accu = (TP+TN)/(TP+TN+FP+FN), the precision is precision = TP/(TP+FP), the f-score is f-score = $2 *$ precision $*$ recall/(pre+recall), and the recall is defined as recall = TP/(TP+FN).

The loss function for this classification task is defined as the cross-entropy, which measures the difference between the predicted and actual values. To introduce the nonlinear property, we use the sigmoid activation function that maps real-valued input to the range [0, 1], which has the form

$$\sigma(\boldsymbol{x}_i) = \frac{1}{1+e^{-\boldsymbol{x}_i}}.$$

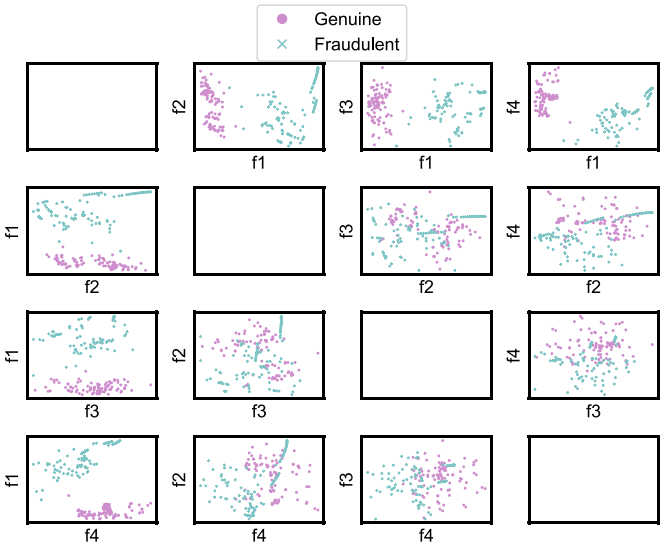

FIG. 7. Scatterplots of credit card fraud detection feature compositions explore relationships between different features in a data set. Each transaction is represented by a point on the scatterplot, with one feature on the x-axis and another on the y-axis.

Then, we use the softmax function to normalize the output of a model into a probability distribution over a set of classes, and the form is

$$\text{softmax}(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i}}{\sum_{j=1}^{K} e^{\boldsymbol{x}_j}},$$

where $\boldsymbol{x}$ is a vector of real-valued inputs, i is the index of a particular element in the vector, and K is the number of classes. So, we write the loss function as

$$L(\theta) = \sum -w_i y_i \log(\text{softmax}(\sigma(\langle\phi(\mathbf{x}_i,\theta)|O|\phi(\mathbf{x}_i,\theta)\rangle))), \tag{9}$$

where y_i is the actual value and w_i is the corresponding weight. If $y_i = 0$, then $w_i = 1$, and if $y_i = 1$, then $w_i = 10$.

VQC problems under different quantum devices can use different encodings for representing quantum states. In general, quantum states can be represented in a number of different ways, including amplitudes (amplitude encoding) and angles (angle encoding). Amplitude encoding represents a quantum state as a complex amplitude, which is a mathematical quantity that describes the probability of finding a particle in a particular state. Angle encoding represents a quantum state as a set of angles, which define the state in terms of rotations around the x, y, and z axes of the Bloch sphere.

In the next two subsections, we present experimental results for different classification tasks using two distinct quantum encoding methods. One subsection reports results on the quantum simulator, while the other presents results on the IBM-Lima device.

A. Simulator

In this case, the Iris flower has four features, which means that a four-qubit system is required to encode the feature information using angle encoding. Since the goal is to predict

TABLE IV. The comparison of the classical resources, quantum resources, final performance, and running time (in seconds) of ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the noise simulator for Iris classification.

Quantum circuit	N of R_s	N of CX	Average accuracy via Simulator	Time (s)
ADQAS-C*	4	2	1	128676
ARQAS-C*	5	2	1	522099
RQAS-C*	5	4	1	134344
HEA	16	6	1	133749

only two species, the VQC can use a single-qubit observer as a measurement operator.

1. Best quantum circuit generated by ADQAS outperforms other algorithms

To analyze the performance of the ADQAS-C* generated by ADQAS, we compare it to the ARQAS-C*, RQAS-C*, and HEA generated by other algorithms such as ARQAS, RQAS, and the hardware-efficient *Ansatz*-based VQE under a depolarizing noise environment. The results of these comparisons are presented in Table IV, which show the classical and quantum resources used, as well as the final performance for predicting the species of Iris flowers in these environments for ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA.

Table IV shows that the ADQAS-C* algorithm uses similar resources to the ARQAS-C* and RQAS-C* algorithms. However, it utilizes fewer classical and quantum resources than HEA-based VQE under a depolarizing noise environment. In particular, ADQAS-C* reduces classical and quantum resource usage by 75% and 73% when compared to the HEA-based VQE. The performances of ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA are similar and equal to 1. The running time for ADQAS-C*, RQAS-C*, and HEA is similar and shorter than that of ARQAS-C*. Overall, the comparison in Table IV shows that the ADQAS-C* and RQAS-C* algorithms both perform better while using fewer classical and quantum resources and less time.

2. Average performance of quantum circuits generated by ADQAS outperforms RQAS algorithms

To illustrate the average performance of ADQAS, ARQAS, and RQAS, all algorithms were run with the same set of hyperparameters. In Fig. 8, we show the mean and variance of the testing accuracy for predicting the species of Iris flowers of the average 10 circuits generated by ADQAS, ARQAS, and RQAS. The results indicate that the average performance of quantum circuits generated by ADQAS and ARQAS outperform RQAS, with a lower average energy and smaller error bar.

B. IBM-Lima

In this subsection, we consider two different classification problems: Iris classification and credit card fraud detection. Both of these problems have four features, and we use amplitude encoding to represent these features using two qubits. As these are both two-class classification problems, we can use a single-qubit observer as the measurement operator in our quantum classifier under IBM-Lima. The basic gates we use are X, SX, R_z, and CNOT, with coupling maps of 0-1 and 1-0. To train our quantum classifier, we use the ADQAS algorithm, which employs the Adam optimizer, 20 000 shots, and Pauli-Z measurement. We trained the quantum classifier on the IBM-Lima quantum simulator and then tested its performance on the real IBM-Lima quantum device.

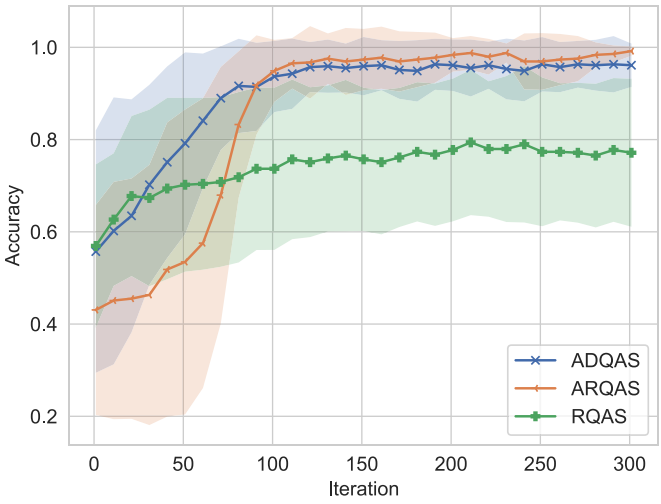

FIG. 8. The average testing accuracy of quantum circuits generated by ADQAS, ARQAS, and RQAS for Iris classification.

1. Best quantum circuit generated by ADQAS outperforms other algorithms

To analyze the performance of ADQAS, we generate the ADQAS-C* using ADQAS under the Lima-Simulator and compare it to the ARQAS-C*, RQAS-C*, and HEA generated by other algorithms such as ARQAS, RQAS, and the hardware-efficient *Ansatz*-based VQE on both the Lima-Simulator and IBM-Lima devices. In Fig. 9, we show the number of parameters and the number of quantum gates. In Tables V and VI, we present the performance and running time of ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under Lima-Simulator and IBM-Lima for classifying the Iris follower and detecting credit card fraud, respectively.

TABLE V. The comparison of final performance and running time (in seconds) between ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the Lima-Simulator and IBM-Lima for Iris classification.

Quantum circuit	Average accuracy via Lima-Simulator	Average accuracy via IBM-Lima	Time (s)
ADQAS-C*	1	1	100582
ARQAS-C*	1	1	108716
RQAS-C*	1	1	65984
HEA	1	0.67	6100

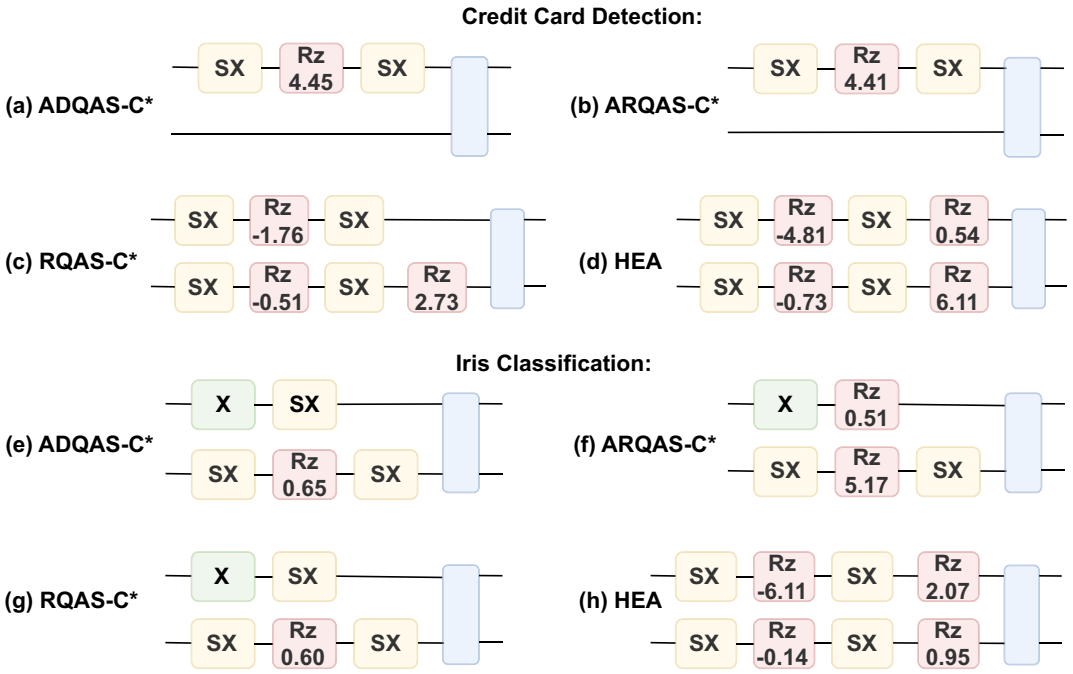

FIG. 9. The quantum circuits generated by ADQAS, ARQAS, RQAS, and HEA-based algorithms for Iris classification and credit card fraud detection, respectively. (a)–(d) ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the Lima-Simulator noise model for credit card fraud detection. (e)–(h) ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the Lima-Simulator noise model for Iris classification.

It appears that ADQAS-C*, ARQAS-C*, RQAS-C* have a shallow depth and fewer parameters compared to the HEA in Fig. 9 under the IBM-Lima environment. Additionally, in Tables V and VI, the ADQAS-C* and ARQAS-C* have been shown to have better performance than the HEA. Furthermore, ADQAS-C* takes less time than ARQAS-C*.

2. *Average performances of quantum circuits generated by ADQAS outperform RQAS*

To evaluate the average performance of ADQAS, ARQAS, and RQAS, all algorithms were run with the same set of hyperparameters. For the task of Iris classification, the average performance results of the 10 circuits generated by ADQAS, ARQAS, and RQAS for the testing data sets on both the quantum simulator and the quantum device are presented in Table VII. Apparently, the $\overline{\text{ADQAS-C}}$ and $\overline{\text{ARQAS-C}}$ have better performances.

TABLE VI. The comparison of final performance and running time (in seconds) between ADQAS-C*, ARQAS-C*, RQAS-C*, and HEA under the Lima-Simulator and IBM-Lima for credit card fraud detection.

Quantum circuit	Average recall via Lima-Simulator	Average recall via IBM-Lima	Time (s)
ADQAS-C*	0.92	0.92	37816
ARQAS-C*	0.92	0.92	55058
RQAS-C*	0.88	0.88	18332
HEA	0.6	0.6	11571

For the task of credit card fraud detection, the average performance results of the average 10 circuits generated by ADQAS, ARQAS, and RQAS for the testing data sets on both the quantum simulator and the quantum device are presented in Table VIII. The $\overline{\text{ADQAS-C}}$ and $\overline{\text{ARQAS-C}}$ also have better performances.

V. PERFORMANCE EVALUATION OF VARIOUS ANSÄTZES FOR DIFFERENT PROBLEMS

To evaluate the effectiveness of the ADQAS method, we conduct several numerical experiments involving seven different problems. For each problem, we evaluated the performance of eight quantum algorithms. Each quantum algorithm is independently tested through five trials, and the results are recorded for analysis. The metrics include testing accuracy, average energy, or approximate optimization results, depending on the specific problem.

The results of the evaluation performed on the IBM fake Kolkata system are presented in Tables IX and X. These tables provide the mean $\bar{\boldsymbol{x}}$ and the maximum deviation, $\max|\boldsymbol{x}-\bar{\boldsymbol{x}}|$,

TABLE VII. The mean final average performance comparison of quantum circuits generated by ADQAS, ARQAS, and RQAS under the Lima-Simulator and IBM-Lima for Iris classification.

Quantum circuits	Mean average accuracy via Lima-Simulator	Mean average Accu via IBM-Lima
$\overline{\text{ADQAS-C}}$	0.74	0.81
$\overline{\text{ARQAS-C}}$	0.76	0.82
$\overline{\text{RQAS-C}}$	0.62	0.6

TABLE VIII. The mean final average performance comparison of quantum circuits generated by ADQAS, ARQAS, and RQAS under the Lima-Simulator and IBM-Lima for credit card fraud detection.

Quantum circuits	Mean average recall via Lima-Simulator	Mean average recall via IBM-Lima
ADQAS-C	0.86	0.86
ARQAS-C	0.86	0.87
RQAS-C	0.72	0.71

of the testing accuracy, average energy, or approximate optimization results for each model problem and optimization method.

The evaluation considered seven distinct model problems: a two-qubit entanglement witness (c1), an eight-qubit eigenvalue problem for a hydrogen chain (H4) (c2), a 12-qubit eigenvalue problem for a hydroxyl cation (OH−) (c3), a 14-qubit eigenvalue problem for beryllium hydride (BeH2) (c4), a 20-qubit eigenvalue problem for the Ising model (c5), a 10-qubit max cut problem (c6), and a 10-qubit portfolio optimization problem (c7).

The evaluation encompassed eight optimization methods, each offering a different approach to solving the model problems. These methods included ADQAS-C*, LBL-ADQAS-C* BS1, BS2, BS3, BS4, BS5, and BS6. ADQAS-C* represents the best quantum circuit generated by ADQAS. LBLADQAS-C* represents the best quantum circuit generated by the LBL-ADQAS method, as discussed in Sec. VII. BS1 is a circuit-centric quantum *Ansatz* [35] with the layer of a single arbitrary unitary gate and controlled unitary gate layer. BS2 is the HEA *Ansatz* [13]. BS3 is the EfficientSU2 *Ansatz* [38], which consists of layers of single-qubit operations spanned by SU(2) and CX entanglements. BS4 is the QAOA *Ansatz* [9] with the mixer layer and the cost layer. BS5 is the two local quantum *Ansatz* with the layer of R_Y and the layer of controlled R_Y [39]. BS6 is a two-design quantum *Ansatz* [40], which consists of alternating rotation and entanglement layers with an initial layer of $\sqrt{\mathrm{H}} = \mathrm{R_y}(\pi/4)$ gates.

For the considered problems, we explore different baseline methods ranging from one to three layers, except for the QAOA *Ansatz*. In the maxcut and portfolio optimization task, we also explore the performance of the QAOA *Ansatz* with additional layers. The test accuracy of VQC, ground energy of VQE, and the optimization result of QAOA obtained by each *Ansatz* are presented in Tables IX and X. Our algorithm consistently outperforms the other methods for all model problems in terms of test accuracy or average energy, and it outperforms the other methods for 5/7 model problems in terms of converged iterations. For all model problems except for the Ising model (c5), we set the iteration number at 500. However, for the Ising model with 20 qubits, we limit the iteration to 100 steps. This decision is due to the high computational time required for this model, where each experiment of 100 steps exceeds five hours.

We aim to achieve the highest test accuracy or the lowest average energy with our proposed method. Trainability includes the number of converged iterations and the achieved accuracy. The number of converged iterations alone does not guarantee optimal results, but in the majority of cases it does outperform them, as highlighted in Tables IX and X. It is important to note that certain circuits with fewer parameters or shortcuts may converge quickly but not with lower accuracy. On the other hand, the tables demonstrate that our algorithm consistently achieves better accuracy compared to other methods, and it performs competitively in terms of converged iterations. Thus, our proposed method not only demonstrates superior accuracy but also exhibits strong competitiveness in terms of trainability.

VI. EXPRESSIBILITY: KL DIVERGENCE

Expressibility can help to understand the reason that autodesigned circuits outperform human-designed circuits in terms of accuracy. Higher expressibility enables autodesigned circuits to encompass a wider range of problem solutions, facilitating more effective optimization towards desired outcomes.

For an arbitrary parametrized quantum circuit $U(\theta)$, the fidelity between two specific quantum circuits (parametrized by ψ^* and θ^*) is given by $F = |\langle 0|U^\dagger(\psi^*)U(\theta^*)|0\rangle|^2$, which follows a probability distribution $F \sim P(f)$. Especially, the probability density function of fidelities for the ensemble of Haar random states is well-established and can be expressed analytically as follows:

$$P_{\mathrm{Haar}}(F) = (2^n - 1)(1 - F)^{2^n - 2}, \tag{10}$$

TABLE IX. Performance evolution of various *Ansätze* for different problems: averaged test accuracy (c1: entanglement witness with 2 qubits), averaged average energy (c2: hydrogen chain with 8 qubits, c3: hydroxyl cation with 12 qubits, c4: beryllium hydride with 14 qubits, and c5: Ising model with 20 qubits), and averaged approximate optimization results (c6: max cut problem with 10 qubits, c7: portfolio optimization problem with 10 qubits).

Method	Problem	ADQAS-C*	LBL-ADQAS-C*	BS1	BS2	BS3	BS4	BS5	BS6
VQC	c1	$\mathbf{0.72 \pm 0.01}$	/	0.71 ± 0.02	0.71 ± 0.02	0.72 ± 0.03	/	0.64 ± 0.03	0.70 ± 0.03
	c2	$\mathbf{-3.22 \pm 0}$	/	-3.04 ± 0.02	-3.12 ± 0.11	-3.12 ± 0.10	/	-3.14 ± 0.06	-2.90 ± 0.06
	c3	$\mathbf{-78.54 \pm 0}$	-78.43 ± 0.24	-77.71 ± 0.49	-78.17 ± 0.43	-78.24 ± 0.12	/	-78.11 ± 0.30	-73.50 ± 14.88
VQE	c4	/	$\mathbf{-18.46 \pm 0.01}$	-17.66 ± 0.01	-18.36 ± 0	-18.23 ± 0.01	/	-18.34 ± 0	-15.94 ± 2.49
	c5	/	$\mathbf{-19.81 \pm 0}$	-16.01 ± 0.26	-16.71 ± 2.18	-14.65 ± 0.46	/	-15.78 ± 1.17	-7.57 ± 3.46
QAOA	c6	$\mathbf{-13.88 \pm 0.10}$	/	-12.90 ± 0.06	-12.93 ± 0.02	-13.62 ± 0.016	-11.67 ± 0.47	-13.78 ± 0.25	-12.92 ± 1.42
	c7	$\mathbf{-1.94 \pm 0.01}$	/	-1.87 ± 0.01	-1.93 ± 0.01	-1.92 ± 0	-1.79 ± 0.03	-1.91 ± 0.01	-1.64 ± 0.18

TABLE X. The averaged number of iterations to reach convergence of various *Ansätze* for different problems (c1–c7).

Method	Problem	ADQAS-C*	LBL-ADQAS-C*	BS1	BS2	BS3	BS4	BS5	BS6
VQC	c1	$\mathbf{18.6 \pm 6.6}$	/	31.4 ± 12.6	27.8 ± 5.8	27.8 ± 22.2	/	$\mathbf{18.6 \pm 3.4}$	22.8 ± 8.2
	c2	$\mathbf{1 \pm 0}$	/	182 ± 138	158 ± 52	144 ± 126	/	220 ± 110	166 ± 104
	c3	$\mathbf{1 \pm 0}$	246 ± 104	310 ± 160	406 ± 106	356 ± 144	/	362 ± 138	424 ± 174
VQE	c4	/	$\mathbf{304 \pm 340}$	500 ± 0	500 ± 0	500 ± 0	/	500 ± 0	498 ± 10
	c5	/	$\mathbf{1 \pm 0}$	100 ± 0	100 ± 0	100 ± 0	/	100 ± 0	100 ± 0
QAOA	c6	288 ± 228	/	306 ± 36	330 ± 90	314 ± 114	$\mathbf{192 \pm 188}$	350 ± 60	326 ± 86
	c7	140 ± 30	/	188 ± 132	216 ± 64	$\mathbf{128 \pm 12}$	162 ± 32	144 ± 26	248 ± 132

where F corresponds to the fidelity of Haar random states, and n is the dimension of the qubit system.

To assess the expressibility of parametrized quantum circuits, one approach [36] is to compare the distribution of state fidelities generated by a parametrized quantum circuit to the state fidelities produced by the Haar distribution. In statistical mathematics, the difference between two probability distributions can be measured using the Kullback-Leibler (KL) divergence [37]. For discrete probability distributions, P and Q, the KL divergence can be defined as

$$D_{\mathrm{KL}}(P||Q) = \sum_j P(j) \ln \frac{P(j)}{Q(j)}. \tag{11}$$

The quantum expressibility of a parametrized quantum circuit ($\mathrm{QExpr}_{\mathrm{PQC}}$) can be described as the KL divergence between the estimated state fidelity distribution of the PQC and that of the Haar distribution:

$$\mathrm{QExpr}_{\mathrm{PQC}} = D_{\mathrm{KL}}[P_{\mathrm{PQC}}(f)||P_{\mathrm{Haar}}(f)], \tag{12}$$

where P_{PQC} represents the estimated probability distribution of fidelities resulting from generating states from a parametrized quantum circuit. A smaller $\mathrm{QExpr}_{\mathrm{PQC}}$ value (i.e., $\mathrm{QExpr}_{\mathrm{PQC}}$ closer to 0) indicates a stronger expressibility of the parametrized quantum circuit.

Meanwhile, ADQAS generates a set of quantum circuits in each generation. The expressibility of ADQAS ($\mathrm{QExpr}_{\mathrm{ADQAS}}$) can be defined as the minimum value of $\mathrm{QExpr}_{\mathrm{PQC}}$ among all the parametrized quantum circuits in ADQAS:

$$\mathrm{QExpr}_{\mathrm{ADQAS}} = \min_{\mathrm{PQC} \in \mathrm{ADQAS}} \mathrm{QExpr}_{\mathrm{PQC}}, \tag{13}$$

which establishes that $\mathrm{QExpr}_{\mathrm{ADQAS}} \leqslant \mathrm{QExpr}_{\mathrm{PQC}}$ when PQC belongs to the quantum circuit generation of ADQAS. If the HEA or a more expressive quantum circuit is present in the generations of ADQAS, the ADQAS method has at least the same power as the HEA. In the IBM-lima quantum system, the basic single-qubit gates include SX, X, I, and $\mathrm{R_z}$, and the two-qubit gate is CX. By defining $\mathrm{R_y} = \mathrm{SX} \cdot \mathrm{R_z} \cdot \mathrm{SX} \cdot \mathrm{X}$ and considering that CX can be applied to the ith and jth qubits following the coupling map of the device with a fake Kolkata system, it becomes apparent that ADQAS can generate the HEA *Ansatz*. Therefore, it is evident that our method can effectively generate a more uniform state distribution compared to the HEA.

To illustrate this, we compare our method with the HEA. The simulation focuses on ADQAS with 200 initially generated circuits with one layer, ranging from 2 to 12 qubits. The results are shown in Fig. 10, in which it can be observed that for every point, the $\mathrm{QExpr}_{\mathrm{ADQAS}}$ is lower than $\mathrm{QExpr}_{\mathrm{HEA}}$. This indicates that the expressibility of our method is higher than that of the HEA. Consequently, our method demonstrates a greater ability to achieve superior results compared to human-designed circuits like the HEA.

VII. SCALABILITY

A. LBL framework

To address the scalability issue in quantum circuit design, we propose a layered ADQAS approach. This approach involves the use of a hybrid algorithm that combines the layer-by-layer (LBL) method with the ADQAS method. The LBL-ADQAS algorithm is designed to handle large qubit problems that have more than 10 qubits. For such problems, we use a single-qubit layer search space of SX-layer, X-layer, $\mathrm{R_z}$-layer, SX-layer, and a two-qubit layer search space of CX-layer.

To evaluate the performance of our algorithm, we apply the circuit generated by traditional ADQAS, the circuit generated by LBL-ADQAS, and BS *Ansätzes* under the IBM fake Kolkata system to solve the eigenvalue of a hydroxyl cation with 12 qubits. We compare the average energy results and the number of iterations required for convergence between LBL-ADQAS-C* and ADQAS-C*, as shown in Tables IX and X. The comparison reveals that although LBL-ADQAS-C* performs slightly worse than ADQAS-C*, it outperforms the other methods.

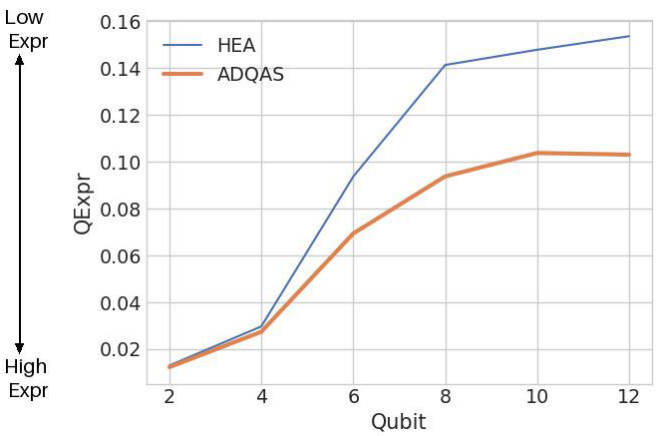

FIG. 10. Comparison of the expressibility value QExpr between the HEA and ADQAS in the quantum system, ranging from 2 to 12 qubits.

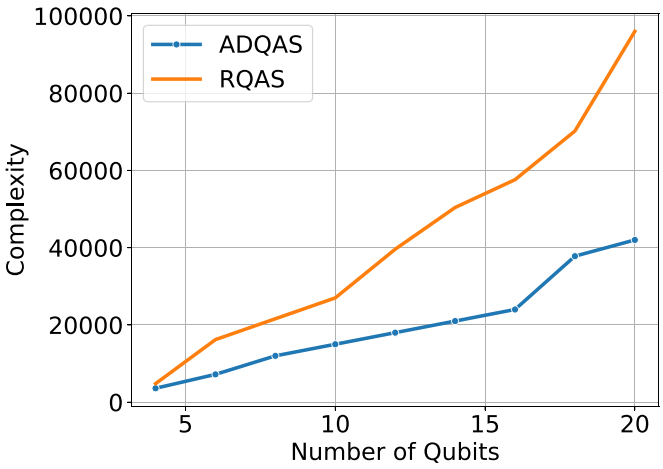

FIG. 11. Comparison of computational complexity between ADQAS and RQAS using the Ising model with qubit sizes ranging from 4 to 20.

There exists a tradeoff between the precision and scalability of circuit search algorithms. The traditional ADQAS method can generate highly optimized circuits but may have limited scalability due to the large number of candidate circuits it generates, which increases the search space size. On the other hand, the LBL-ADQAS method reduces the search space size, enhancing scalability, but it may not discover the most optimized circuits and may miss some optimization opportunities.

Therefore, when selecting circuit search algorithms, both aspects should be considered, and the tradeoff between them should be balanced. If the problem requires high precision, the traditional ADQAS method can be chosen. However, if the problem is relatively simple but with a large scale, a search algorithm with better scalability may be a better choice, as it can enhance the scalability of the search algorithm while ensuring an acceptable circuit for the problem.

B. Scalability analysis

Within the LBL framework, we conducted a complexity analysis considering the number of qubits, layers, iterations, and samplers. We use the Ising model, with sizes ranging from 4 to 20 qubits, as an illustrative example. The complexity is calculated as $n \times L \times T \times N$, where n is the number of qubits, L is the number of layers, T is the number of iterations for each sampler, and N is the number of samplers. Figure 11 demonstrates that ADQAS significantly reduces complexity compared to RQAS methods, thus enhancing feasibility for large-scale systems.

VIII. CONCLUSION

Hybrid quantum-classical algorithms hold promise in providing a quantum advantage. Nevertheless, designing parametrized quantum circuits that perform well in the presence of noise and coupling topologies is a significant challenge. In this study, we introduce an adaptive diversity-based quantum adaptive search algorithm that finds efficient quantum circuits while optimizing the use of classical and quantum resources. Our algorithm utilizes an adaptive approach to identify circuits and employs a diversity-based method to reduce the pool of candidate circuits.

Our ADQAS algorithm significantly reduces classical and quantum resources by 28% and 23%, respectively, compared to a hardware-efficient *Ansatz*, and outperforms RQAS in terms of average energy for the six-qubit Heisenberg model when evaluated on a simulator. Furthermore, our algorithm produces optimized circuits with a more shallow depth and fewer parameters than a hardware-efficient *Ansatz* and has lower average energy than RQAS for the four-qubit Heisenberg model when tested on the IBM-Lima.

For the Iris classification task, our algorithm generates optimized circuits that reduce classical and quantum resources by 75% and 73%, respectively, compared to a hardware-efficient *Ansatz* and achieve higher accuracy than RQAS on a simulator. Additionally, our algorithm exhibits superior recall and accuracy compared to the random-based algorithm for both Iris classification and credit card fraud detection when tested on the IBM-Lima.

ACKNOWLEDGMENTS

The authors of this work would like to extend their sincere gratitude to HKUST and HSBC for their invaluable support and contribution to this research project. Additionally, we are deeply grateful to all the individuals from both parties who generously contributed their time and expertise to this work. Their contributions were essential to the success of the project, and we are truly thankful for their dedication and effort.

[1] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 1994), pp. 124–134.

[2] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 1996), pp. 212–219.

[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[4] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[5] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill *et al.*, Quantum advantage in learning from experiments, Science **376**, 1182 (2022).

[6] S. F. Ahmad, R. Rawat, and M. Moharir, Quantum machine learning with hqc architectures using non-classically simulable

feature maps, in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (IEEE, Piscataway, NJ, 2021), pp. 345–349.

[7] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, Hierarchical quantum classifiers, npj Quantum Inf. **4**, 65 (2018).

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

[9] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[10] O. Kyriienko and E. B. Magnusson, Unsupervised quantum machine learning for fraud detection, arXiv:2208.01203.

[11] M. Grossi, N. Ibrahim, V. Radescu, R. Loredo, K. Voigt, C. Von Altrock, and A. Rudnik, Mixed quantum–classical method for fraud detection with quantum feature selection, IEEE Trans. Quantum Eng. **3**, 1 (2022).

[12] Y. Li, J. Hu, X.-M. Zhang, Z. Song, and M.-H. Yung, Variational quantum simulation for quantum chemistry, Adv. Theor. Simul. **2**, 1800182 (2019).

[13] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[14] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.

[15] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, Learning transferable architectures for scalable image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2018), pp. 8697–8710.

[16] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, Mnasnet: Platform-aware neural architecture search for mobile, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2019), pp. 2820–2828.

[17] H. Liu, K. Simonyan, and Y. Yang, Darts: Differentiable architecture search, arXiv:1806.09055.

[18] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary Algorithms for Solving Multi-objective Problems* (Springer, New York, NY, 2007), Vol. 5.

[19] E. Torabian and R. V. Krems, Compositional optimization of quantum circuits for quantum kernels of support vector machines, Phys. Rev. Res. **5**, 013211 (2023).

[20] D. Stamoulis, R. Ding, D. Wang, D. Lymberopoulos, B. Priyantha, J. Liu, and D. Marculescu, Single-path nas: Designing hardware-efficient convnets in less than 4 hours, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, Switzerland, 2019), pp. 481–497.

[21] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles, Machine learning of noise-resilient quantum circuits, PRX Quantum **2**, 010324 (2021).

[22] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver, arXiv:1910.09694.

[23] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, Quantum circuit architecture search: Error mitigation and trainability enhancement for variational quantum solvers, npj Quantum Inf. **8**, 62 (2022).

[24] D. Chivilikhin, A. Samarin, V. Ulyantsev, I. Iorsh, A. Oganov, and O. Kyriienko, Mog-vqe: Multiobjective genetic variational quantum eigensolver, arXiv:2007.04424.

[25] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, Quantumnas: Noise-adaptive Search for Robust Quantum Circuits, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, Piscataway, NJ, 2022), pp. 692–708.

[26] Y. Huang, Q. Li, X. Hou, R. Wu, M.-H. Yung, A. Bayat, and X. Wang, Robust resource-efficient quantum variational ansatz through an evolutionary algorithm, Phys. Rev. A **105**, 052414 (2022).

[27] E.-J. Kuo, Y.-L. L. Fang, and S. Y.-C. Chen, Quantum architecture search via deep reinforcement learning, arXiv:2104.07715.

[28] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko, Reinforcement learning for optimization of variational quantum circuit architectures, Adv. Neur. Inf. Proc. Syst. **34**, 18182 (2021).

[29] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Differentiable quantum architecture search, Quantum Sci. Technol. **7**, 045023 (2022).

[30] F.-X. Meng, Z.-T. Li, X.-T. Yu, and Z.-C. Zhang, Quantum circuit architecture optimization for variational quantum eigensolver via Monte Carlo tree search, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[31] J. Cheng, H. Wang, Z. Liang, Y. Shi, S. Han, and X. Qian, Topgen: Topology-aware bottom-up generator for variational quantum circuits, arXiv:2210.08190.

[32] J. Chu, X. Yu, S. Yang, J. Qiu, and Q. Wang, Architecture entropy sampling-based evolutionary neural architecture search and its application in osteoporosis diagnosis, Complex & Intell. Syst. **9**, 213 (2023).

[33] H. L. Tang, V. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, PRX Quantum **2**, 020310 (2021).

[34] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).

[35] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[36] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, Adv. Quantum Techn. **2**, 1900070 (2019).

[37] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Statis. **22**, 79 (1951).

[38] Y. Y. Atas, J. Zhang, R. Lewis, A. Jahanpour, J. F. Haase, and C. A. Muschik, SU(2) hadrons on a quantum computer via a variational approach, Nat. Commun. **12**, 6499 (2021).

[39] C. Roman-Vicharra and J. J. Cai, Quantum gene regulatory networks, npj Quantum Inf. **9**, 67 (2023).

[40] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).