Quantum Information Processing (2024) 23:224
https://doi.org/10.1007/s11128-024-04438-2

Variational quantum algorithms: fundamental concepts, applications and challenges

Han Qi[1] · Sihui Xiao[1] · Zhuo Liu[1] · Changqing Gong[1] · Abdullah Gani[2]

Received: 20 February 2024 / Accepted: 16 May 2024 / Published online: 3 June 2024

Abstract

Quantum computing is a new discipline combining quantum mechanics and computer science, which is expected to solve technical problems that are difficult for classical computers to solve efficiently. At present, quantum algorithms and hardware continue to develop at a high speed, but due to the serious constraints of quantum devices, such as the limited numbers of qubits and circuit depth, the fault-tolerant quantum computing will not be available in the near future. Variational quantum algorithms(VQAs) using classical optimizers to train parameterized quantum circuits have emerged as the main strategy to address these constraints. However, VQAs still have many challenges, such as trainability, hardware noise, expressibility and entangling capability. The fundamental concepts and applications of VQAs are reviewed. Then, strategies are introduced to overcome the challenges of VQAs and the importance of further researching VQAs is highlighted.

Keywords Quantum computing · Noisy-intermediate scale quantum · Variational quantum algorithms · Quantum machine learning

✉ Sihui Xiao
xiaosihui@stu.sau.edu.cn

Han Qi
qihan@sau.edu.cn

Zhuo Liu
liuzhuo@stu.sau.edu.cn

Changqing Gong
gongchangqing@sau.edu.cn

Abdullah Gani
abdullah@um.edu.my

[1] School of Computer Science, Shenyang Aerospace University, Shenyang 110000, People's Republic of China

[2] Faculty of Computing and Informatics, University of Malaya, Jalan Universiti, 50603 Kuala Lumpur, Malaysia

1 Introduction

Quantum computing [1] is a kind of computation based on the principles of quantum mechanics, with powerful computing capability far beyond classical computing. Extending from the cutting-edge basic research in mathematics and physics to the cross-fertilization with many engineering disciplines, and then to the development of highly engineered application technologies, quantum computing has become a rapidly emerging technology field. Based on the background of the era of big data and the historical node of the rapid development of quantum computing, it is of great significance to carry out research related to quantum computing.

With the rapid development of quantum algorithms and hardware, mankind has entered a key new era in the development of quantum technology, which has been called the Noisy Intermediate-Scale Quantum (NISQ) era [2]. Quantum devices always produce errors due to the role of the environment and some bias in the control of the quantum devices, a shortcoming that severely limits the size of the quantum system and the depth of circuits. However, quantum computers that can realize fault-tolerant technology [3] have a huge demand for quantum resources, which is difficult to realize in recent quantum devices, which means that fault-tolerant quantum algorithms such as the Shor's algorithm [4], the Grover's algorithm [5], and the HHL algorithm [6] need a very long period of technological accumulation to move toward applications. The variational quantum algorithms (VQAs) [7, 8], by outsourcing the parameter optimization to a classical optimizer, can mitigate the effect of noise on the computational accuracy to a certain extent, and are most promising to show quantum advantages on quantum devices in the near future.

VQAs have been considered for a wide range of applications (see Fig. 1), covering essentially all the applications that researchers have envisioned for quantum computers. In this review, we discuss the exciting prospects for VQAs and highlight the challenges that must be overcome to achieve the ultimate goal of quantum supremacy.

2 Fundamental concepts

One of the main advantages of the VQAs is that they provide a general framework for solving multiple types of problems, featuring the use of a Quantum Processing Unit (QPU) to run parameterized quantum circuits, followed by the use of a Classical Processing Unit (CPU) for parameter optimization. The working principle of the VQAs is shown in Fig. 2. Specifically, the algorithm consists of the following four steps:

Step 1 Encoding the problem as a cost function C(θ) transforms the computational task into an optimization problem such that the computational task can be achieved when C(θ) takes the minimum value.

Step 2 Construct the parameterized quantum circuit ansatz. When θ is taken such that the cost function reaches its minimum value, ansatz becomes a quantum model that can solve the target problem.

Step 3 Measure the output quantum state, obtain the measurement results, and calculate the cost function C(θ).

Step 4 Update the parameter θ using the classical parameter optimizer.

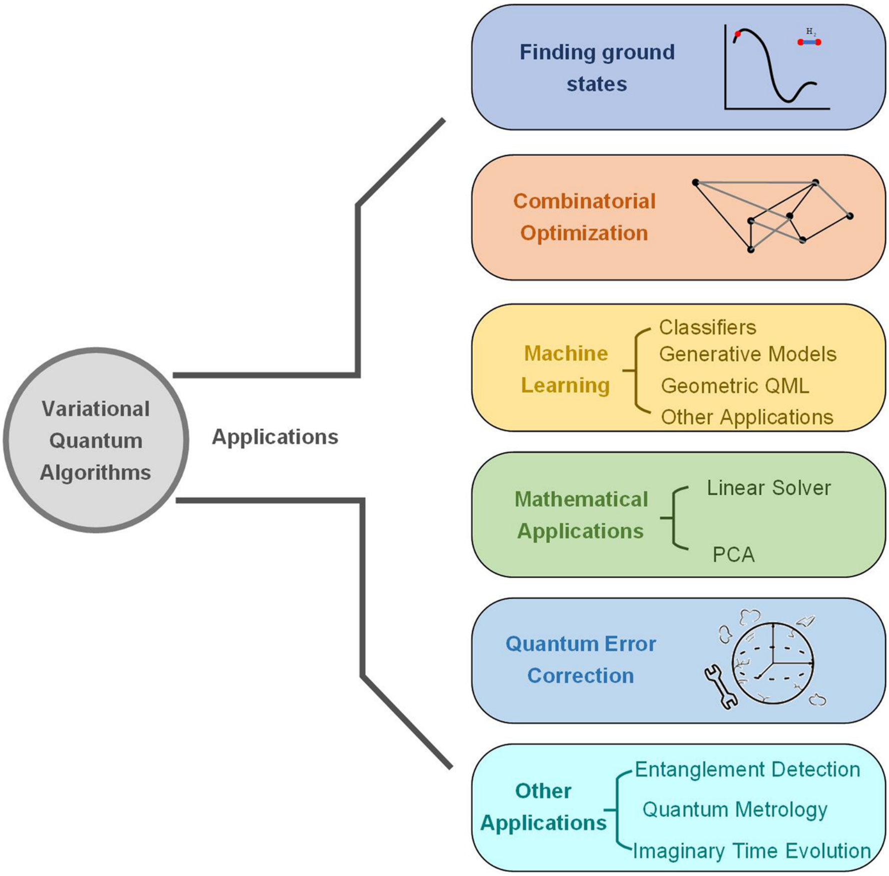

Fig. 1 Applications of variational quantum algorithms (VQAs)

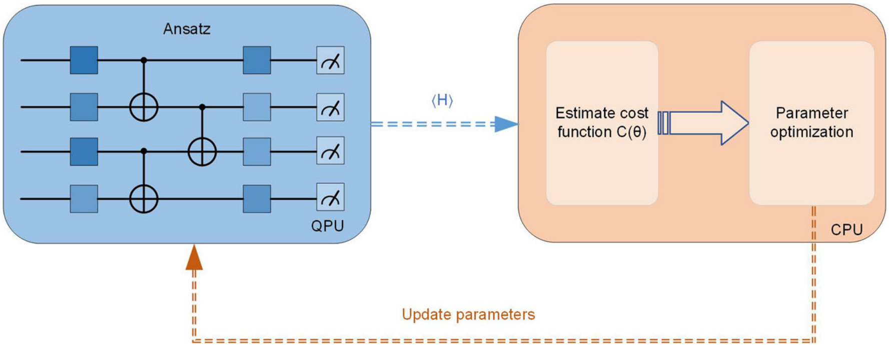

Fig. 2 Schematic diagram of a variational quantum algorithm (VQA)

As shown in Fig. 2, implementing a VQA must consider how to design the cost function to transform the computational task into an optimization problem and how to construct an ansatz. In addition, the chosen classical optimizer may have an impact on the efficiency and accuracy of the algorithm. In what follows, we discuss these basic concepts in VQAs.

2.1 Cost function

The cost function is a crucial aspect of VQA's design. Figuratively speaking, the cost function defines a high-dimensional surface that has high and low undulations in the parameter space. Based on this, the optimizer updates the parameters of the quantum circuit and finds the optimal value of the cost function. In general, the cost function can be expressed as

$$\begin{aligned} C(\theta) &= \sum_j f_j\left(\mathrm{Tr}\left[H_j U(\theta)\rho_j U^\dagger(\theta)\right]\right) \\ &= \sum_j f_j(\langle H_j\rangle), \end{aligned} \tag{1}$$

where ρ_j is a set of input quantum states, H_j is a set of observable measurements, $\langle H_j\rangle$ denotes the result of a H_j measurement at the end of the circuit, and f_j is a set of classical functions that denote the classical post-processing of the measurement. In this equation, j represents different measurement schemes, rather than describing different quantum states. ρ_j and H_j are pairs within the same measurement scheme. Depending on the kind of H_j [9], the methods for computing the cost function can be roughly categorized as follows:

(1) H_j is the Pauli operator. The cost function can be obtained directly from a Pauli measurement of the quantum state. In particular, in some common physical models (e.g., the Ising model), H_j is a linear combination of local Pauli operators:

$$H_j = \sum_k c_{j,k} P_{j,k}, \tag{2}$$

where $P_{j,k}$ is a Pauli operator that acts only on local qubits. If each $P_{j,k}$ acts on at most m qubits, the cost function is said to be m-local, and the local cost function is superior to the global cost function for the gradient vanishing problem [10].

(2) H_j is the density operator σ. The computational task often includes calculating the overlap of states between quantum states ($\mathrm{Tr}[\rho\sigma]$). This type of cost function can be computed on a quantum device by the swap test [11] or its variant destructive swap test [12] (circuit implementations are shown in Fig. 3a and b, respectively).

(3) H_j contains the unitary operator. A transformation of the unitary operator is required, and a common transformation is the Hadamard test [13], as shown in Fig. 3c.

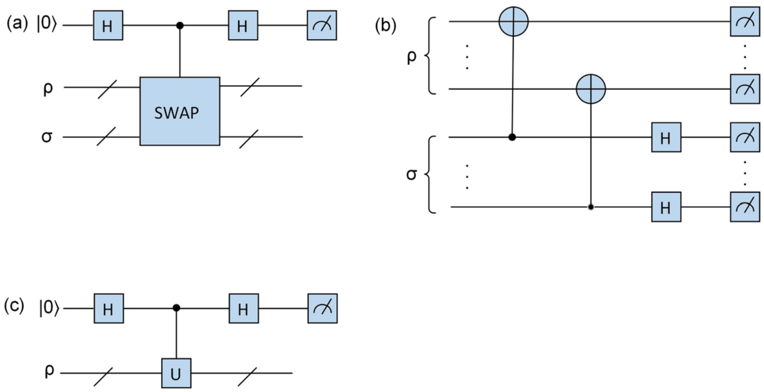

Fig. 3 **a** Swap test circuit; **b** Destructive swap test circuit; **c** Hadamard test circuit

2.2 Ansatz

Ansatz is an important module of variational quantum circuits. The specific form of ansatz determines the form of the parameter θ and how the parameter θ is trained to optimize the cost function. For the cost function of Equation (1), the parameter θ can be encoded as unitary U(θ) applied to the input state of the quantum circuit. U(θ) can usually be expressed as the product of sequentially applied unitaries

$$U(\boldsymbol{\theta}) = U_n(\boldsymbol{\theta}_n) \cdots U_2(\boldsymbol{\theta}_2) U_1(\boldsymbol{\theta}_1), \tag{3}$$

with

$$U_x(\boldsymbol{\theta}_x) = \prod_j e^{-i\theta_j H_j} W_j, \tag{4}$$

where W_j is a parameter-free quantum gate, H_j is a Hamiltonian operator, and θ_x is the x-th component of θ.

The choice of ansatz greatly affects the performance of VQA. On the one hand, ansatz affects the closeness between the final state and the optimal state for solving the problem; on the other hand, the quantum hardware on which the VQA is performed must be taken into account: deeper circuits are susceptible to errors, and some ansatz gates are costly to build from local gates. Thus, ansatz can be categorized as Problem-inspired ansatz and Hardware efficient ansatz.

2.2.1 Problem-inspired ansatz

(1) Unitary coupled cluster ansatz

The Unitary Coupled Cluster (UCC) ansatz [14] is mainly used in quantum chemistry, where it adds quantum correlations to the Hartree-Fock approximation. Peruzzo et al. [15] utilized the UCC ansatz to implement Variational Quantum Eigensolver(VQE).

UCC ansatz consists of a parameterized cluster operator T(θ) and acts on the Hartree-Fock ground state $|\Psi_{\mathrm{HF}}\rangle$:

$$|\Psi(\boldsymbol{\theta})\rangle = e^{T(\boldsymbol{\theta})-T(\boldsymbol{\theta})^{\dagger}}|\Psi_{\mathrm{HF}}\rangle. \tag{5}$$

The cluster operator is given by T(θ)=$T_1(\theta)$+$T_2(\theta)$+...,where

$$T_1(\theta) = \sum_{\substack{i \,\epsilon\, \mathrm{occ} \\ j \,\epsilon\, \mathrm{virt}}} \theta_i^j \; \hat{a}_j^{\dagger} \; \hat{a}_i, \tag{6}$$

$$T_2(\boldsymbol{\theta}) = \sum_{\substack{i_1,i_2 \,\epsilon\, \mathrm{occ} \\ j_1,j_2 \,\epsilon\, \mathrm{virt}}} \theta_{i_1,i_2}^{j_1,j_2} \; \hat{a}_{j_2}^{\dagger} \; \hat{a}_{i_2} \; \hat{a}_{j_1}^{\dagger} \; \hat{a}_{i_1}. \tag{7}$$

The operators $\hat{a}_j^{\dagger}$ ($\hat{a}_i$) are fermionic creation (annihilation) operators. The sets occ (virt) refer to occupied (unoccupied) Hartree-Fock orbits. There are many variants of the UCC ansatz, some of which reduce the circuit depth by considering more efficient ways of compiling fermionic operators [16–19].

(2) Quantum alternating operator ansatz
The Quantum Approximate Optimization Algorithm [20] is used to deal with combinatorial optimization problems, which alternates between applying a cost-function-based Hamiltonian and a mixing Hamiltonian. Hadfield et al. [21] extended this framework to allow alternation between broader sets of operators. At the core of this expansion lies the Quantum Alternating Operator ansatz, which considers general parametrized families of unitaries, rather than restricting to those corresponding to fixed local Hamiltonian time-evolution specified by a parameter. This ansatz enables the representation of a larger, potentially more valuable, set of states compared to the original formulation. Lloyd et al. [22] show that such ansatz is capable of generalized quantum computation for one-dimensional local cost Hamiltonians consisting of nearest-neighbor ZZ terms and provides a generalized set of gates. Morales et al. [23] further generalized the universality of this set of gates with the families of ansatzes defined by sets of graphs and hyper-graphs.The form of Quantum Alternating Operator ansatz is

$$U(\boldsymbol{\beta},\boldsymbol{\gamma}) = \prod_{\ell=1}^{L} e^{-i\beta_\ell H_M} e^{-i\gamma_\ell H_P}, \tag{8}$$

where H_M is mixer Hamiltonian, H_P is problem Hamiltonian whose ground state encodes the solution of the combinatorial problem, L is the number of layers of the circuit, and (β,γ) are two sets of parameters to be optimized. This ansatz essentially implements a Trotterized adiabatic evolution of L Trotter steps, which allows the system to evolve from the ground state of an easily prepared mixer Hamiltonian to the target ground state of the problem Hamiltonian. L determines the precision of the solution.

(3) Variational Hamiltonian ansatz
Inspired by adiabatic state preparation, Wecker et al. [24] developed the Variational Hamiltonian Ansatz(VHA), which prepares a ground state for a given Hamiltonian $H = \sum_i \hat{h}_i$ ($\hat{h}_i$ are Hermitian operators) by a Trotterized adiabatic state preparation process. Here, each Trotter step corresponds to a variational ansatz. Therefore, the unitary is given by:

$$U_{\text{VHA}} = \prod_i e^{\left(i\theta_i \hat{h}_i\right)}. \tag{9}$$

Experiments show that the VHA outperforms a specific form of UCC ansatz in a strongly correlated model system of quantum chemistry.

2.2.2 Hardware efficient ansatz

Unlike the problem-inspired ansatz, hardware efficient ansatz [25] is designed around properties of quantum hardware such as qubit connectivity, restricted gate set, limited gate fidelity and coherence time to ensure efficient implementation on NISQ devices. A common feature of these types of ansatz is the use of a limited set of quantum gates and a specific qubit connection topology. Gate sets typically consist of two-qubit entanglement gates and up to three single-qubit gates, and the circuit consists of blocks containing single-qubit gates and entanglement gates that are applied to multiple qubits in parallel. Each block is called a layer, and ansatz contains several such layers. A hardware efficient ansatz with L layers is of the form:

$$U(\theta) = \prod_{k=1}^{L} U_k\left(\theta_k\right) W_k, \tag{10}$$

where θ_k is the variational parameter, $U_k\left(\boldsymbol{\theta}_k\right) = e^{\left(-i\boldsymbol{\theta}_k V_k\right)}$ is derived from unitary of the Hermitian operator V_k, W_k represents a nonparametric quantum gate.

The selection, connectivity and ordering of the gates in the ansatz affect the ability of the ansatz to cover the entire Hilbert space and the speed of convergence. Some related properties of hardware efficient ansatz have been studied in some literature, including expressibility [26], entangling capability [27], performance on benchmark issues [28] and trainability [29].

2.3 Quantum measurement

Quantum measurement [30] is the process of determining the state of a quantum system. According to the basic principles of quantum mechanics, measurement causes the system to collapse into a specific state of the measurement base, and the measurement results are random. A quantum system described by the state $|\varphi\rangle$ can be described by a set of measurement operators M_m that operate on the state space of the system. The probability of obtaining the outcome m is

$$p(m) = \langle\varphi|M_m^\dagger M_m|\varphi\rangle. \tag{11}$$

The state of the system after measurement is

$$\frac{M_m|\varphi\rangle}{\sqrt{\langle\varphi|M_m^\dagger M_m|\varphi\rangle}}. \tag{12}$$

where the measurement operator needs to satisfy the completeness equation:

$$\sum M_{\mathrm{m}}^\dagger M_{\mathrm{m}} = \mathrm{I}. \tag{13}$$

And the sum of all probabilities is 1:

$$\sum_m p(m) = \sum_m \langle\varphi|M_m^\dagger M_m|\varphi\rangle = 1. \tag{14}$$

Common quantum measurement methods include projective measurement and Positive Operator-Valued Measure, which are respectively introduced below.

2.3.1 Projective measurement

The projective measurement is described by an observable M, which is a Hermitian operator on the state space of a system to be observed. The observable M can be written in spectral decomposition form as follows:

$$M = \sum_m m P_m, \tag{15}$$

where P_m is the projector onto the eigenspace of M with eigenvalue m. The probability of obtaining the outcome m when the state of the measurement is $|\varphi\rangle$ is

$$p(m) = \langle\varphi|P_m|\varphi\rangle\,. \tag{16}$$

Given the measurement m, the state of the quantum system after the measurement is $\frac{P_m|\varphi\rangle}{\sqrt{p(m)}}$. An important feature of projection measurement is the ability to easily calculate the average value of the measurement:

$$E(M) = \sum_m mp(m) = \sum_m m\langle\varphi|P_m|\varphi\rangle = \langle\varphi|\left(\sum_m mP_m\right)|\varphi\rangle = \langle\varphi|M|\varphi\rangle. \tag{17}$$

2.3.2 Positive operator-valued measure

In certain experiments, the system is measured only once, typically at the conclusion of the experiment. In such instances, a mathematical tool called the Positive Operator-Valued Measure(POVM) is particularly well-suited for analyzing measurements. Suppose we define

$$E_m \equiv M_m^\dagger M_m, \tag{18}$$

where E_m is a set of positive operators and satisfies a complete relation, that is, $\sum_m E_m = I$. When the measurement state is $|\varphi\rangle$, the probability of obtaining the outcome m is

$$p(m) = \langle\varphi|E_m|\varphi\rangle. \tag{19}$$

E_m are sufficient to determine the probabilities of the different measurement outcomes. The complete set $\{E_m\}$ is known as a POVM.

2.4 Parameter optimization

After determining the cost function and ansatz, the classical results of the output quantum state of ansatz under the parameter θ are obtained by quantum measurement operations, which in turn calculate the value of the cost function. The optimization principle of VQA is to use a parameter optimizer to update the parameter θ and to find the local optimal value of the given cost function. Thus the success of VQA depends on the efficiency and reliability of the optimization method used. The optimization methods are classified into two categories based on whether they use gradient descent or not.

2.4.1 Gradient-based methods

The most common optimization method iterates according to the gradient indication: the gradient $\nabla_\theta C(\boldsymbol{\theta})$ of the cost function C(θ) with respect to its parameter θ is computed, and the value of the parameter is iteratively varied by using gradient descent to move in the direction of the currently computed gradient as illustrated in the following equation:

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla_\theta C(\theta)|_{\theta^{(t)}}, \tag{20}$$

where η is the step size.

The parameter shift rule [31] is a fundamental tool for computing the analytic gradient in VQA. The rule shows that when the generating element A_k in the parameterized unitary gate $V_l(\boldsymbol{\theta}_l) = \Pi_k \mathrm{e}^{-\mathrm{i}\theta_k A_k/2}$ satisfies $A_k^2 = I$, the partial derivative of the cost function without classical post-processing with respect to the parameter θ_k is as follows:

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_k} = \frac{C\left(\theta_k + \frac{\pi}{2}\right) - C\left(\theta_k - \frac{\pi}{2}\right)}{2}. \tag{21}$$

The original parameter shift rule was designed for single-parameter gates. Wierichs et al. [32] extended this approach to multi-parameter quantum gates.

Based on the unique properties of parameterized quantum circuits, optimization methods specifically for VQA have been proposed. Stokes et al. [33] proposed quantum natural gradient optimization. Compared with the traditional gradient descent method, quantum natural gradient optimization requires additional computation of the factor g^+ in each round of iteration:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \boldsymbol{g}^+ \nabla_\theta C(\theta)|_{\theta^{(t)}}, \tag{22}$$

where g is the Fubini-Study metric tensor and g^+ denotes the pseudo-inverse of g. Quantum natural gradient optimization has faster convergence in some VQE problems compared to traditional gradient descent methods. Koczor et al. [34] generalized the quantum natural gradient method to arbitrary quantum states by using quantum Fisher information as the core metric of the density operator space. This approach explicitly takes into account the shortcomings of variational quantum circuits and can be considered for finding optimal parameters when the used ansatz is not perfect.

2.4.2 Gradient-free methods

Another type of optimization method is one that does not use gradient descent. Nelder and Mead [35] proposed the Nelder-Mead simplex method for function minimization, which was a derivative-free method. It involves evaluating the objective function at the vertices of a simplex and moving away from the poorest value. This process is adaptive, continuously revising the simplexes to better conform to the nature of the response surface. Vidal et al. [36] proposed a coordinate-descent optimization algorithm to find the ground state of the Hamiltonian M. Given starting point θ and D_j which are eigenvalue distances of Hamiltonian j, evaluate $f(t) := C(\theta + te_j)$. Compute the Fourier transform $\hat{f}(k)$, $k \in D_j$. Rewrite the Fourier expansion as a trigonometric polynomial and find its minimum t_0. Update $\theta = \theta + t_0 e_j$ until no improvement is made. Nakanishi et al. [37] proposed the quantum sequential minimal optimization. The key to this type of optimization method is the use of the following property of the cost function: when the parameters of the parameterized quantum circuit are independent of each other, and when the generating element A_k in the parameterized unitary gate $V_l(\boldsymbol{\theta}_l) = \Pi_k \mathbf{e}^{-\mathrm{i}\theta_k \boldsymbol{A}_k/2}$ satisfies $A_k^2 = I$, fixing the other parameters, the cost function is a sinusoidal function with a period of 2π for any parameter. Based on this, if a single parameter is chosen, the cost function will become a simple sinusoidal curve with a period of 2π, allowing precise minimization of the selected parameter. Then repeat the process for all the parameters until convergence.

3 Applications

VQAs have a wide range of applications, including finding ground states, combinatorial optimization, machine learning, etc. In this section, we will introduce some typical applications and their experimental progress in detail.

3.1 Finding ground states

The best-known application of VQAs is finding the ground state of a given Hamiltonian H. The first proposed VQA, the Variational Quantum Eigensolver (VQE) [15], was devised to offer a near-term resolution to this challenge. The core task of the VQE is to solve the ground state energy and corresponding quantum state of the Hamiltonian H of a closed physical system on the quantum scale. Firstly, prepare an ansatz $|\Psi(\theta)\rangle$ on the quantum device. The whole ansatz construction process can be described as:

$$|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes N} = \prod_{p,q} U_p(\theta_p)V_q|0\rangle^{\otimes N}. \tag{23}$$

The optimization algorithm is then used to continuously optimize the parameter vector θ such that the cost function C(θ)=$\langle\Psi(\boldsymbol{\theta})|H|\Psi(\boldsymbol{\theta})\rangle$ is minimized. According to the Rayleigh-Ritz principle, C(θ) needs to satisfy: $C(\theta) \geq E_0$. where E_0 denotes the ground state energy of the system. From the point of view of numerical analysis, the problem can be understood as solving for the minimum eigenvalue and the eigenvector corresponding to that eigenvalue of a discretized Hamiltonian. The workflow of VQE is next described in detail using the example of solving the ground state energy of a hydrogen molecule.

The present VQE algorithm follows the research idea of modern computational chemistry and adopts the representation of second quantization. Under the second quantization representation, the Hamiltonian is expressed in the form of the following fermionic operator [38]:

$$H_F = \sum_{p,q} h_{pq}a_p^\dagger a_q + \frac{1}{2}\sum_{p,q,r,s} h_{pqrs}a_p^\dagger a_q^\dagger a_r a_s, \tag{24}$$

where the subscripts p,q,r,s stand for different single-electron spin molecular orbitals, the generation operator $a_j^\dagger$ and the annihilation operator a_j denote the creation and annihilation of an electron in the jth molecular orbital, and the h_{pq} and h_{pqrs} are the one-electron and two-electron integrals, respectively. It is important to note that the creation and annihilation operators are not native operations on a quantum computer, and a step of mapping from fermionic to Pauli operators is required. A common mapping method is the Jordan-Wigner transformation [39]:

$$a_j = I \otimes \cdots I \otimes \left(\frac{X_j + iY_j}{2}\right) \otimes Z_{j-1} \cdots \otimes Z_0, \tag{25}$$

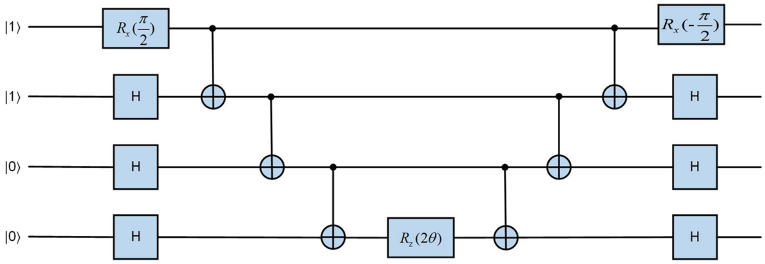

Fig. 4 Simplification of the hydrogen molecule UCC ansatz

$$a_j^\dagger = I \otimes \cdots I \otimes \left(\frac{X_j - iY_j}{2} \right) \otimes Z_{j-1} \cdots \otimes Z_0. \tag{26}$$

Under the JW transformation, VQE operations on hydrogen molecules require 4 qubits. In practical VQE experiments, more efficient but complex mappings are often used to reduce the number of qubits required for the calculation, such as the Bravyi-Kitaev transformation [40]. At present, the optimal mapping method from fermionic operator to Pauli operator is still a problem worth discussing.

After preprocessing the molecular Hamiltonian, the next step is to design an ansatz that uses as few quantum gates as possible while being expressive enough to accomplish the task. The common ansatz structure in VQE is UCC ansatz based on unitary coupled cluster theory. When the cluster operator is truncated to the two-particle excitation operator T_2, the ansatz is called UCCSD ansatz. For molecular hydrogen, the UCCSD operator can be written as

$$U_{\text{UCCSD}}^{\text{H}_2} = e^{\theta_2^0 (a_2^\dagger a_0 - a_0^\dagger a_2) + \theta_3^1 (a_3^\dagger a_1 - a_1^\dagger a_3) + \theta_{23}^{01} (a_3^\dagger a_2^\dagger a_1 a_0 - a_0^\dagger a_1^\dagger a_2 a_3)}. \tag{27}$$

Specifically for the hydrogen molecule under the JW transformation, the reference state is taken as $|\Psi\rangle_{\text{ref}} = |\Psi\rangle_{HF}^{H_2} = |0011\rangle$. The corresponding UCC circuit can be simplified as [41]:

$$U_{\text{UCC}}^{\text{H}_2}(\theta) = e^{-i\theta X_3 X_2 X_1 Y_0}. \tag{28}$$

The ansatz can be specifically written as

$$\begin{aligned} |\Psi(\theta)\rangle &= (\cos(\theta) I - i \sin(\theta) X_3 X_2 X_1 Y_0)\, |0011\rangle \\ &= \cos(\theta)|0011\rangle - \sin(\theta)|1100\rangle, \end{aligned} \tag{29}$$

which is realized by the circuit shown in Fig. 4.

After designing the part that operates on the quantum processor, the optimization method is designed on the classical processor, thus updating the ansatz cyclically. After finding the optimal parameters, the ground state energy of the hydrogen molecule is estimated (the Hamiltonian of the hydrogen molecule contains 15 Pauli strings):

$$\boldsymbol{E}(\boldsymbol{\theta}) = \sum_{k=1}^{15} c_k \langle H_k \rangle, \tag{30}$$

where c_k denotes the weight coefficient.

In addition, VQE can also be used to calculate energy derivatives [42–44], look for excited states [45–48] and solve general nonlinear problems including nonlinear partial differential equations [49]. Currently, VQE has been demonstrated in various quantum architectures, such as superconducting qubits [50–52], photonic systems [15, 53] and trapped ions [41, 54]. Recently, Grimsley et al. [55] proposed an adaptive, problem-tailored VQE, which is relatively unaffected by the large number of local minima and barren plateaus in VQE.

At present, there are many open problems waiting to be solved in VQE algorithm, such as how to reduce the number of qubits involved in VQE tasks, how to efficiently map the Hamiltonian in the form of fermionic operator to the form of Pauli string, how to efficiently construct ansatz of large molecules and how to find the optimal optimization method for VQE experiments.

3.2 Combinatorial optimization

Combinatorial optimization problems refer to finding the optimal solution from a discrete set of feasible solutions, such as the Traveling Salesman Problem, the Maximum Cut(MaxCut) Problem, and other well-known NP-hard problems. Computers may not be able to solve such problems efficiently. In this case, approximate optimal solutions of such optimization problems can be found.

In 2014, Farhi et al. [20] first proposed the Quantum Approximate Optimization Algorithm (QAOA) for solving combinatorial optimization problems in an approximate way. To solve a classical combinatorial optimization problem on a quantum computer, it needs to be transformed into a quantum optimization problem first. The most straightforward approach is to encode the objective function of the original problem as a Hamiltonian such that the ground state of that Hamiltonian corresponds to the solution of the original optimization problem. In this way, the combinatorial optimization problem becomes a problem of solving the ground state of the Hamiltonian, and thus QAOA can be seen as a special case of VQE. Next, we take the MaxCut Problem as an example to introduce how to realize the QAOA algorithm, and the detailed process is shown in Fig. 5. The MaxCut Problem is a graph partitioning problem defined as follows. Given a graph, each vertex is to be assigned one of two colors. An edge can be "cut" when it connects two vertices of different colors. The objective is to assign colors to the vertices in such a way that maximizes the number of edges that can be cut simultaneously.

Step 1 Define a cost Hamiltonian $\hat{H}_C$ that satisfies the eigenequation

$$\hat{H}_C \left|\mathbf{x}\right\rangle = C(\mathbf{x}) \left|\mathbf{x}\right\rangle . \tag{31}$$

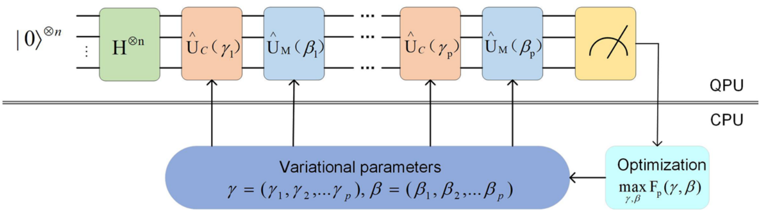

Fig. 5 The working principle of QAOA

In order to find the quantum state representing the optimal solution from the Hilbert space, it is necessary to define another Hamiltonian $\hat{H}_M$ according to the adiabatic theorem [56]. Define a graph G=(V,E), where the elements of set V are the vertices of the graph, and each element of set E is a pair of vertices denoting an edge connecting these two vertices. Then $\hat{H}_C$ and $\hat{H}_M$ are given by the following form:

$$\hat{H}_C = \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(I - Z_i Z_j), \tag{32}$$

$$\hat{H}_M = \sum_{j\in V} X_j, \tag{33}$$

where I is the identity operator and $Z_j(X_j)$ is the Pauli-Z(Pauli-X) operator acting on the jth qubit.

Step 2 The eigenstate with the largest eigenvalue of $\hat{H}_M$ is:

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle. \tag{34}$$

Take this quantum state $|s\rangle$ as the initial state of the algorithm.

Step 3 Define the unitary transformation according to the Hamiltonian $\hat{H}_C$ and $\hat{H}_M$, respectively:

$$\hat{U}_C(\gamma) = e^{-i\gamma \hat{H}_C} = \prod_{i=1, j<i}^{n} R_{Z_i Z_j}(-2w_{ij}\gamma), \tag{35}$$

$$\hat{U}_M(\beta) = e^{-i\beta \hat{H}_M} = \prod_{i=1}^{n} R_{X_i}(2\beta), \tag{36}$$

where γ and β are the variational parameters of the circuit.

Step 4 Given any integer p≥1 and the parameter $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_p)$, $\boldsymbol{\beta} = (\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \ldots, \boldsymbol{\beta_p})$, define

$$|\psi_p(\gamma, \beta)\rangle = e^{-i\beta_p \hat{H}_M} e^{-i\gamma_p \hat{H}_C} \ldots e^{-i\beta_1 \hat{H}_M} e^{-i\gamma_1 \hat{H}_C} |s\rangle, \tag{37}$$

where p denotes the number of layers of $\hat{U}_C$ and $\hat{U}_M$ used, $\hat{U}_C$ and $\hat{U}_M$ are alternately acted on the initial state $|s\rangle$ p times.

Step 5 Notate $F_p(\gamma, \beta)$ as the expected value of $\hat{H}_C$ with respect to $|\psi_p(\gamma, \beta)$:

$$F_p(\gamma, \beta) = \langle \psi_p(\gamma, \beta)|\hat{H}_C|\psi_p(\gamma, \beta)\rangle. \tag{38}$$

Step 6 A classical optimization algorithm is used to iteratively update the parameters γ and β, to find the optimal set of parameters (γ^*, β^*) to maximize the expected value $F_p(\gamma, \beta)$:

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} F_p(\gamma, \beta). \tag{39}$$

Since QAOA has been proposed, a large number of scholars have initiated research on it. A promising variant is the warm-start QAOA (WS-QAOA) proposed by Egger et al. [57] and Tate et al. [58]. The warm-start approach modifies the distribution of bit strings in the initial state of the ansatz, deviating from the equal superposition typically used in standard QAOA. This adjustment aims to enhance the amplitude of an approximate solution obtained classically. Compared to standard QAOA, the WS-QAOA demonstrated improvement in solution quality compared to the original version of QAOA, especially for shallow depths. Zhou et al. [59] conducted an in-depth study on the performance of QAOA on the MaxCut Problem by developing a parameter optimization procedure based on a heuristic strategy to find high-quality parameters that are difficult to be handled by traditional schemes. Bravyi et al. [60] experimentally found that higher-level recursive QAOA outperforms the most well-known general-purpose classical ones based on rounding an SDP relaxation. Herrman et al. [61] explored a multi-angle ansatz for QAOA, aiming to reduce circuit depth and enhance the approximation ratio by increasing the number of classical parameters. Empirical evidence was provided for eight-vertex graphs, indicating that a single layer of the multi-angle ansatz rivaled three layers of the conventional approach in solving MaxCut problems. Additionally, the multi-angle QAOA demonstrated superior approximation ratios compared to QAOA at equivalent depths across a set of MaxCut instances on fifty and one-hundred vertex graphs. Amaro et al. [62] introduced the filtering variational quantum eigensolver which utilized filtering operators to attain accelerated and more dependable convergence toward the optimal solution. Through numerical evaluations using randomly weighted MaxCut instances, they demonstrated the superior performance of filtering variational quantum eigensolver compared to both the original VQE and QAOA. Crooks et al. [63] show that QAOA with fixed circuit depth is not sensitive to problem size; In addition, MaxCut QAOA can be efficiently implemented on gate-based quantum computers with limited qubit connections using qubit switching networks. Guerreschi et al. [64] claim that QAOA requires at least hundreds of qubits to achieve quantum speed-up. Streif et al. [65] compared QAOA with quantum annealing and simulated annealing, showing that QAOA can find solutions to special structural optimization problems when both quantum annealing and simulated annealing fail. Recently, Lykov et al. [66] proposed a QAOA simulator that reduces the optimization time of classical QAOA parameters by a factor of 11 for 26 qubits compared to a state-of-the-art GPU quantum circuit simulator based on cuQuantum.

In addition to theoretical studies, related experimental demonstrations have proliferated, with experimental work on superconducting quantum [67], ion trap [68], optical quantum [69], and other mainstream types of hardware devices. Harrigan et al. [67] implemented QAOA with up to 23 qubits on Google's Sycamore superconducting quantum processor. Ebadi et al. [70] solved the Maximum Independent Set Problem in two spatial dimensions using an array of Rydberg atoms with up to 289 qubits.

3.3 Quantum machine learning

Quantum machine learning is a cross-disciplinary field that applies the principles and techniques of quantum computing to the field of machine learning. While traditional machine learning algorithms may become very complex when dealing with certain problems, quantum machine learning aims to take advantage of the parallelism and interference of quantum computing to improve computational efficiency. Some of the more common applications of quantum machine learning are described below, focusing mainly on classification models and generative models. Furthermore, a significant advancement in the field that has occurred recently is discussed: geometric quantum machine learning.

3.3.1 Classification models

The classification task is a very basic problem in the field of machine learning, but also has important practical applications. The classification process is essentially a process of labeling data. When the input data satisfies a certain condition, the data is labeled accordingly, thus completing the classification. Classification problems are usually given a dataset containing N samples: $\{(x^{(k)}, y^{(k)})\}_{k=1}^{N}$, where $x^{(k)}$ is the input data, $y^{(k)}$ is the label of the data. The aim is to use this dataset to train a Variational Quantum Classifier (VQC). The VQC can make the correct classification when it encounters the unprocessed data.

Typically, the given dataset is classical data, so the first step is to encode the classical data into quantum states that can be executed on a quantum device, i.e., $x \to U_{\varphi(x)} |0\rangle^{\otimes n}$, which can also be referred to as a feature map. $\varphi(x)$ is a preprocessing function that converts data vectors into circuit parameters. The next step is to apply a parameterized quantum circuit U(θ) on the quantum state, and then define the cost function as the distance between the true label and the expected value of some observable measurement O, i.e.,

$$C(\boldsymbol{\theta}) = \sum_k \left(y^{(k)} - \langle\varphi|^{(k)} \boldsymbol{U}^{\dagger}(\boldsymbol{\theta}) \boldsymbol{O} \boldsymbol{U}(\boldsymbol{\theta}) |\varphi\rangle^{(k)} \right)^2 . \tag{40}$$

The above process is shown in Fig. 6 and consists of three modules, namely preprocessing module, ansatz module and post-processing module. The pre-processing module samples the data vector x from the dataset and maps it to the encoding circuit

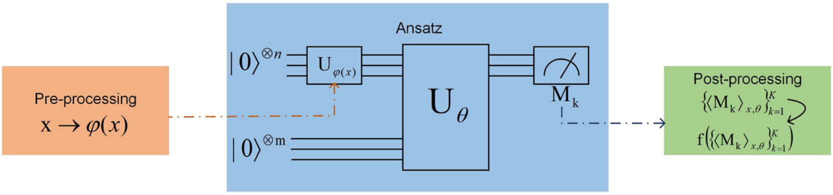
Pre-processing
$x \rightarrow \varphi(x)$
Ansatz
$|0\rangle^{\otimes n}$
$U_{\varphi(x)}$
$|0\rangle^{\otimes m}$
U_θ
M_k
Post-processing
$\{\langle M_k \rangle_{x,\theta}\}_{k=1}^{K}$
$f(\{\langle M_k \rangle_{x,\theta}\}_{k=1}^{K})$

Fig. 6 The working principle of VQC

$U_{\varphi(x)}$ of the ansatz module. The variational circuit U(θ) acts on the prepared state of $U_{\varphi(x)}$ and the additional register of the ancilla qubit, produces the state $U_{\boldsymbol{\theta}} U_{\varphi(\boldsymbol{x})}|0\rangle$. A set of observables $\{\langle M_k \rangle_{x,\theta}\}_{k=1}^{K}$ is estimated from the measurements, and these estimates are mapped to the output space by the classical post-processing function f. The cost function is then computed and optimized using a classical optimizer. The cost function converges to the minimum value by continuously adjusting the parameters.

An early representative of this type of work is the hybrid classical-quantum algorithm for machine learning on near-term quantum processors proposed by Mitarai et al. [71] in 2018, which combines low-depth quantum circuits and classical computers to approximate complex nonlinear functions in a way that takes advantage of the exponentially large space of quantum systems. In the same year, inspired by tensor networks, Grant et al. [72] used TTN and MERA to construct a highly entangled ansatz for binary classification of two classical machine learning datasets, Iris and MNIST. In their simulations, MERA consistently outperformed TTN.The trained VQC was resistant to depolarizing noise and was successfully deployed on the ibmqx4 quantum computer. In 2020, Schuld et al. [73] proposed a VQC to encode the input features as amplitudes of the quantum system. Since this encoding circuit can be very expensive, the authors considered the depth of the circuit while ensuring that it is highly expressive. The authors perform classification experiments on the inputs using quantum circuits with parameterized single-qubit and two-qubit gates as well as single-qubit measurements. The results show that this VQC performs well on classical machine learning datasets and uses fewer parameters than traditional methods. Havlíček et al. [74] implemented variational quantum classifiers at five different depths on a superconducting processor. They found that as the depth of the circuits increases, the classification accuracy increases accordingly. Wang et al. [75] combined quantum computing with the extreme learning machine. They proposed a quantum extreme learning machine that achieved exponential acceleration on perfect quantum devices and a variational quantum extreme learning machine that ran on near-term quantum devices. Recently, Park et al. [76] proposed a variational quantum approximation support vector machine (VQASVM) algorithm.The authors also tested VQASVM on a cloud-based NISQ processor using a small dataset, demonstrating its potential for practical applications. To date, the scale of experiments related to variational quantum classification has been relatively small due to the limitations of available quantum hardware. In future, as quantum devices continue to evolve, the scale of experiments will expand.

3.3.2 Generative models

The idea of training a variational quantum ciruit for a quantum machine learning implementation can also be appiled for a generative model, which is an unsupervised learning task that has been successfully applied in many domains, such as image generation, speech synthesis, and missing text inference. Generative models' goal is to model an unknown probability distribution and generate new samples similar to the training data [77]. Specifically, the task is to learn a model distribution q_θ that is close to the target distribution p, i.e., to find the parameters shown in the following equation:

$$\theta^* = \arg\min_{\boldsymbol{\theta}} D(p, q_\theta), \tag{41}$$

where D defines the similarity between p and q_θ. Since the target probability distribution is unknown, it is approximated using a dataset $\{\boldsymbol{v}^{(i)}\}_{i=1}^N$ which is distributed according to the target distribution. The framework of Figure 6 can also be used to explain the generative models. Assume that the target distribution satisfies $v^{(i)} \in \{0, 1\}^n$. First, since there is no input data for this type of problem, the encoding circuit is set to $U_{\varphi(\boldsymbol{x})} = I$. Then, U_θ is applied to the initial state $|0\rangle^{\otimes n}$. Finally, the set of operators $\{\langle M_{\boldsymbol{v}}\rangle_{\boldsymbol{\theta}}\}_{\boldsymbol{v}}$ is measured, where $M_{\boldsymbol{v}} = |v\rangle\langle v|$ are projectors for the bitstrings. The generated model, called a Quantum Circuit Born Machine (QCBM), realizes the probability distribution

$$q_{\boldsymbol{\theta}}(\boldsymbol{v}) = \mathrm{tr}\Big(M_{\boldsymbol{v}} U_{\boldsymbol{\theta}} |0\rangle\langle 0| U_{\boldsymbol{\theta}}^\dagger\Big). \tag{42}$$

Since the target distribution data is binary, no post-processing is required. If the target distribution data is real-valued, the bit string can be interpreted as a discrete output and a post-processing module can be used to recover the real values.

Benedetti et al. [78] introduced a variational framework for training QCBM. They used particle swarm to minimize the Kullback–Leibler divergence (i.e.,$D(p, q_{\boldsymbol{\theta}}) = \sum_{\boldsymbol{v}} p(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v})}{q_{\boldsymbol{\theta}}(\boldsymbol{v})}$)'s approximation. In experimental simulations, they successfully trained QCBM for the canonical Bars-and-Stripes dataset and the Boltzmann distribution, and provided a new evaluation metric: qBAS score. Zhu et al. [79] implemented the scheme on a four-qubit trapped ion computer and experimentally demonstrated the convergence of the model to the target distribution. Liu et al. [80] designed an efficient gradient-based learning algorithm for QCBM. The proposed learning algorithm can be run on recent quantum devices and can demonstrate the quantum advantages of probabilistic generative modeling. The authors used gradient descent method to minimize $D(p, q_{\boldsymbol{\theta}}) = \| \sum_{\boldsymbol{v}} p(\boldsymbol{v})\varphi(\boldsymbol{v}) - \sum_{\boldsymbol{v}} q_{\boldsymbol{\theta}}(\boldsymbol{v})\varphi(\boldsymbol{v})\|^2$. Their approach allows gradient estimation using discrete target data, which is usually not possible in classical implicit models. Hamilton et al. [81] implemented the scheme on IBM Q20 Tokyo computer. Coyle et al. [82] proposed a new cost function and introduced quantum kernels into the generative model, and finally demonstrated the superiority of the algorithm experimentally. Benedetti et al. [83] utilized QCBM as variational distributions over discrete variables. They explored two distinct realizations: one with an adversarial objective and

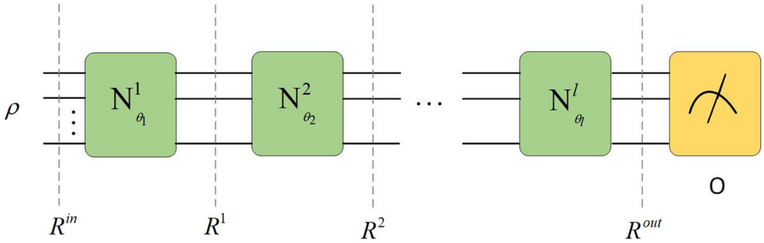

Fig. 7 The architecture of EQNNs

the other based on the kernelized Stein discrepancy. Their methods facilitate efficient variational inference with distributions that exceed the representational capabilities of classical computers.

3.3.3 Geometric quantum machine learning

Geometric deep learning [84] is a rapidly developing subject as a result of recent research on the use of symmetries in machine learning, particularly in problems with more general symmetry groups than translations. Geometric Quantum Machine Learning (GQML) [85] is the term for this application in quantum computing. Using concepts from geometric deep learning, GQML builds quantum models with precise inductive biases according to the symmetries of the given issue. For example, it makes sense to use models whose outputs are invariant to the action of any local unitary [86] when dividing states into those with a high or low level of multipartite entanglement [87].

Nguyen et al. [85] present a theoretical framework for constructing GQML models by extending the concept of classical equivariant neural networks(ENNs) to equivariant quantum neural networks (EQNNs).The architecture of EQNNs is shown in Fig. 7. Dashed lines denote the representations of the symmetry group G at particular stages within the EQNN, which may vary across layers. The input state ρ is acted upon by the representation R^{in}. The lth layer of the EQNNs, $N^l_{\boldsymbol{\theta}_l}$, must be (G, R^l, R^{l+1})-equivariant. Therefore, the full architecture $\phi = N^L_{\theta_L} \circ \cdots \circ N^1_{\theta_1}$, is (G, R^{in}, R^{out})-equivariant. Here, the definition of (G, R^{in}, R^{out})-equivariant is given as follows. A representation (R, H) of a group G on a vector space H is a homomorphism $R : G \rightarrow GL(H)$ from the group G to the space of invertible linear operators on H, which maintains the group structure of G. Given a group G and its representations $(R^{\text{in}}, H^{\text{in}})$ and $(R^{\text{out}}, H^{\text{out}})$. A linear map ϕ is (G, R^{in}, R^{out})-equivariant if and only if

$$\phi \circ \text{Ad}_{R^{\text{in}}(g)} = \text{Ad}_{R^{\text{out}}(g)} \circ \phi, \forall g \in G, \tag{43}$$

where $\text{Ad}_{R(g)}$ denotes the adjoint representation. The measurement operator O acts on the output representation R^{out}. EQNNs have the potential to mitigate several critical challenges in quantum machine learning, including barren plateaus, an abundance of local minima, and inadequate data requirements.

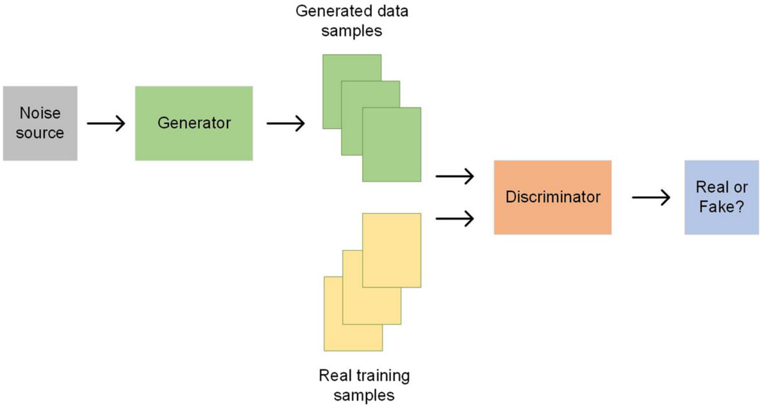

Fig. 8 The working principle of GAN

While recent proposals have initiated the exploration of the potential of GQML [88–90], the field is still nascent, necessitating a more methodical approach to developing symmetry-encoded models.

3.3.4 Other applications

(1) Quantum Generative Adversarial Networks
Generative Adversarial Networks(GAN) [91] consists of two parts: the generator and the discriminator. The generator accepts random noise as input and generates new samples by learning the data distribution. The discriminator determines whether the input samples are real data or not, as shown in Fig. 8.

Quantum Generative Adversarial Networks (QGAN) [91] aims to accelerate the training of classical generative adversarial networks by utilizing quantum computational properties. In the QGAN framework, the generator G and the discriminator D correspond to two parameterized quantum circuits U_G and U_D, respectively. The generator's goal is to minimize the cost function, aiming to make the generated samples more realistic; the discriminator's goal is to maximize the cost function, trying to correctly discriminate between real and generated data. These two form an optimization problem in a min-max game. The generator and the discriminator fight against each other and keep learning, which ultimately makes the discriminator unable to judge whether the output results of the generator are real or not. Aiming at problems such as unstable training of QGAN, Chakrabarti et al. [92] proposed Quantum Wasserstein Generative Adversarial Networks (QWGAN) to improve the robustness and scalability of the model.

(2) Quantum autoencoder
The quantum autoencoder [93] consists of an encoder and a decoder, which is an algorithm used for feature dimension reduction. Inputting the quantum state ρ_{AB} of the composite quantum system AB, the encoder circuit E=U(θ) is applied to the quantum state to obtain $U(\boldsymbol{\theta})\boldsymbol{\rho}_{AB}\boldsymbol{U}^{\dagger}(\boldsymbol{\theta})$. By encoding the information of quantum system AB

onto quantum system A, quantum system B only needs to be measured, i.e., $\tilde{\boldsymbol{\rho}}_A = \mathrm{Tr}_B(U(\boldsymbol{\theta})\boldsymbol{\rho}_{AB}\boldsymbol{U}^{\dagger}(\boldsymbol{\theta}))$. At this point, the compression of the message is completed. When decoding, it is necessary to introduce a system C of the same dimension as the system B and fix its quantum state ρ_C. The decoder circuit $D = U^{\dagger}(\boldsymbol{\theta})$ is applied to the quantum system AC to obtain the recovered quantum state $\tilde{\boldsymbol{\rho}}_{AC} = \boldsymbol{U}^{\dagger}(\boldsymbol{\theta})[\boldsymbol{\rho}_c \otimes \mathrm{Tr}_B(\boldsymbol{U}(\boldsymbol{\theta})\boldsymbol{\rho}_{AB}\boldsymbol{U}^{\dagger}(\boldsymbol{\theta}))]\boldsymbol{U}(\boldsymbol{\theta})$. Cao et al. [94] generalized the quantum autoencoder to mixed states and provided a noise-assisted algorithm to improve the recovering fidelity.

3.4 Mathematical applications

At present, a variety of VQAs have been proposed to solve mathematical problems. Two of them, the variational quantum linear solver (VQLS) and Principal Component Analysis (PCA) involving diagonalized covariance matrices, are presented next.

3.4.1 Variational quantum linear solver

Solving systems of linear equations is a fundamental computational problem with a wide range of applications in science and engineering. Quantum computing offers the possibility of exponential acceleration of this task. For a system of linear equations $Ax = b$, the VQLS [95] first decomposes the matrix A into a linear combination of unitary matrices $A = \sum_j c_j A_j$, and then prepares the corresponding quantum state $|b\rangle$ based on the normalized vector b. Input $|0\rangle$ into ansatz to obtain $|x\rangle = \boldsymbol{U}(\boldsymbol{\theta})\,|0\rangle$, and then minimize the cost function

$$
\begin{aligned}
C(\boldsymbol{\theta}) &= 1 - \frac{\mathrm{Tr}\left\lfloor |b\rangle\,\langle b|\,\boldsymbol{A}\,|x\rangle\,\langle x|\,\boldsymbol{A}^{\dagger}\right\rfloor}{\mathrm{Tr}\left[\boldsymbol{A}\,|x\rangle\,\langle x|\,\boldsymbol{A}^{\dagger}\right]} \\
&= 1 - \frac{\langle b|\,\boldsymbol{A}\,|x\rangle\,\langle x|\,\boldsymbol{A}^{\dagger}\,|b\rangle}{\langle x|\,\boldsymbol{A}^{\dagger}\boldsymbol{A}\,|x\rangle} \\
&= 1 - \frac{\sum_{j,k} c_j c_k^{*}\,\langle b|\,\boldsymbol{A}_j\,|x\rangle\,\langle x|\,\boldsymbol{A}_k^{\dagger}\,|b\rangle}{\sum_{j,k} c_j c_k^{*}\,\langle x|\,\boldsymbol{A}_j^{\dagger}\boldsymbol{A}_k\,|x\rangle}.
\end{aligned}
\tag{44}
$$

Now $A\left|x^{\mathrm{opt}}\right\rangle \propto |b\rangle$. The linear equations can be solved by calculating $\left|x^{\mathrm{opt}}\right\rangle / \left|A\left|x^{\mathrm{opt}}\right\rangle\right|$. Lubasch et al. [49] extended VQLS to a large class of nonlinear problems. Their method was demonstrated with the time-independent nonlinear Schrödinger equation, employing the total energy as the cost function and discretizing space into a finite grid. Utilizing multiple copies of variational quantum states within the cost-evaluation circuit enables this VQA to handle nonlinear functions.

3.4.2 Principal component analysis

PCA serves to reduce the dimensionality of the data, it involves diagonalizing the covariance matrix of the dataset. In 2014, Lloyd et al. [96] proposed a quantum algorithm for PCA, but its subroutines, quantum phase estimation and density

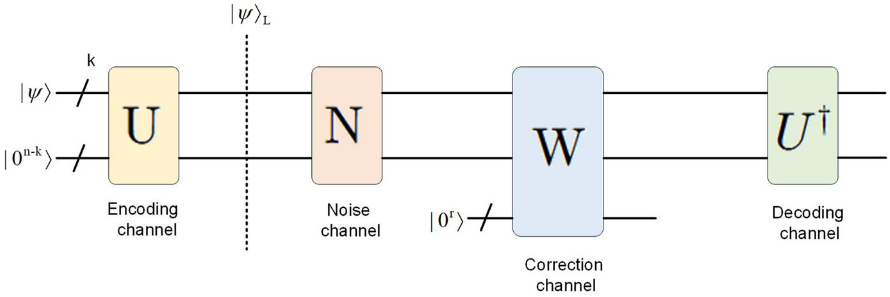

Fig. 9 The working principle of quantum error correction

matrix exponentiation, were not implementable in the NISQ era. In 2019, LaRose et al. [97] proposed a variational quantum state diagonalization algorithm. For a quantum state ρ, the cost function quantifies the Hilbert-Schmidt distance between $\tilde{\rho}(\boldsymbol{\theta}) = U(\boldsymbol{\theta})\rho U(\boldsymbol{\theta})^\dagger$ and $\boldsymbol{Z}(\tilde{\rho}(\boldsymbol{\theta}))$, where $\boldsymbol{Z}$ is the dephasing channel. The authors successfully diagonalized a quantum state on Rigetti's quantum computer and found the entanglement spectrum of the base state of the Heisenberg model on the simulator.

3.5 Quantum error correction

Quantum error correction [4] mitigates the effects of hardware noise. The basic framework, shown in Fig. 9, can be summarized as the following process: The encoding channel U encodes the quantum state $|\Psi\rangle$ of k logical qubits into n physical qubits to obtain the logical quantum state $|\Psi\rangle_{\mathrm{L}}$. After the logical quantum state $|\Psi\rangle_{\mathrm{L}}$ passes through the noise channel N (determined by the hardware device), the error correction channel W tries to correct the effect of the noise on it. Finally, the decoding channel $U^\dagger$ (the inverse of the encoding channel) decodes the quantum state from the physical bits. The advantage of using VQAs for quantum error correction is that the characteristics of hardware devices can be comprehensively considered when constructing parameterized circuits, e.g., the types of quantum gates supported as well as topological structures, so the more efficient hardware-related error correction schemes can be designed. Two representative exploratory works are presented next.

Johnson et al. [98] proposed a Quantum Variational Error Corrector(QVECTOR). The basic idea is to represent the encoding channel U and the error correction channel W shown in Fig. 9 as parameterized quantum circuits $\boldsymbol{U}(\cdot) = \boldsymbol{U}(\boldsymbol{\theta}_1)(\cdot)\boldsymbol{U}^\dagger(\boldsymbol{\theta}_1)$ and $\boldsymbol{W}(\cdot) = \boldsymbol{W}(\boldsymbol{\theta}_2)(\cdot)\boldsymbol{W}^\dagger(\boldsymbol{\theta}_2)$. Define the average fidelity of the following form as a cost function:

$$F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \int_\psi \boldsymbol{d}\mu(\psi)\langle\psi|\boldsymbol{U}^\dagger(\boldsymbol{\theta}_1) \cdot \boldsymbol{W}(\boldsymbol{\theta}_2) \cdot \boldsymbol{N} \cdot \boldsymbol{U}(\boldsymbol{\theta}_1)(|\psi\rangle\langle\psi|)|\psi\rangle, \qquad (45)$$

where $\mu(\psi)$ denotes the Haar measure. This cost function describes the average behavior of the above error correction scheme on the input quantum states. When $F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = 1$, $\boldsymbol{U}(\boldsymbol{\theta}_1)$ and $\boldsymbol{W}(\boldsymbol{\theta}_2)$ realize complete error correction for the noisy channel N. The experimental results show that compared to the five-bit quantum error-correcting code [99], QVECTOR has better performance in dealing with specific noise.

Xu et al. [100] proposed a variational circuit compiler for quantum error correction, the core of which is to encode $|\Psi\rangle_{\mathrm{L}}$ as the ground state eigenvectors of the Hamiltonian H. VQE is called to solve the ground state energy and eigenvector of H. Assume that S is a stabilizer code [101] of a certain single qubit error correction code and its stabilizer is $S = \langle g_1, \cdots, g_k \rangle$.$\left|\Psi^{\perp}\right\rangle_{\mathrm{L}}$ is the orthogonal state of $|\Psi\rangle_{\mathrm{L}}$. Define the operator $O_{\mathrm{L}} = |\Psi\rangle_{\mathrm{L}} \langle\Psi|_{\mathrm{L}} - \left|\Psi^{\perp}\right\rangle_{\mathrm{L}} \langle\Psi^{\perp}|_{\mathrm{L}}$, and O_{L} encodes the coefficient information of the logical quantum state $|\Psi\rangle_{\mathrm{L}}$. Then define the Hamiltonian:

$$H = -\sum_{k=1}^{K} c_k \boldsymbol{g}_k - c_o \boldsymbol{O}_{\mathrm{L}}, \tag{46}$$

where $c_k, c_o > 0$. It can be verified that $|\Psi\rangle_{\mathrm{L}}$ is the ground state eigenvector of H, corresponding to the ground state energy $E_0 = -\left(c_o + \sum_{k=1}^{K} c_k\right)$. Taking the five-bit quantum error-correcting code as an example, for different hardware devices, the above method can give a logical quantum state preparation circuit with fewer quantum gates.

3.6 Other applications

This section describes several VQAs that do not fall into the above categories but are very representative.

3.6.1 Entanglement detection

Quantum entanglement is a key resource in quantum technology. The capability to manipulate quantum entanglement forms the foundation for realizing practical applications of quantum technologies. While numerous techniques have been proposed to detect quantum entanglement, they are not applicable for NISQ devices. Wang et al. [102] proposed Variational Entanglement Detection (VED), which provided a new method for quantum entanglement detection on NISQ devices.The VED algorithm leverages the positive map criterion and operates as follows: Firstly, it decomposes a positive map into a series of quantum operations that can be executed on NISQ devices. Subsequently, it variationaly estimates the minimum eigenvalue of the final state, derived from performing these operations on the target state and averaging the resulting states. Two methods are proposed for computing the average: the initial method computes the average by weighting the output states based on the quasi-probability distribution, while the second method estimates the average using a sampling technique, which is probabilistic in nature. The first method computes the average by weighting the output states based on the quasi-probability distribution, while the second method estimates the average using a sampling technique which is probabilistic. Finally, if the optimized minimal eigenvalue is negative, it asserts that the target state is entangled.

3.6.2 Quantum metrology

Quantum metrology [103] aims to increase the precision of a measured quantity that is estimated in the presence of statistical errors using entangled quantum states. Quantum Fisher information (QFI) quantifies the ultimate accuracy of estimating parameters from quantum states and can be viewed as a reliability metric. However, estimating the QFI of a mixed state is often a computationally demanding task. Beckey et al. [104] proposed Variational Quantum Fisher Information Estimation (VQFIE), which treats the QFI as a cost function and updates the parameters to maximize it.

3.6.3 Imaginary time evolution

Imaginary time evolution [105] is a tool for the study of quantum systems and is widely used in many areas of physics. In real-time evolution, the propagation function of a quantum system with Hamiltonian H is $\mathbf{e}^{-\mathrm{i}Ht}$. In imaginary time evolution, since the time $\tau = \mathrm{i}t$ is imaginary, the propagation function of the system is $\mathrm{e}^{-H\tau}$. The longer the time of evolution, the lower the energy of the system will be, and eventually it will stabilize at the system's ground state energy E_0. Since the imaginary time evolution process is mathematical rather than physical, how to simulate the evolution process is the key to the imaginary time evolution algorithm. Motta et al. [106] introduced the quantum imaginary time evolution, which was an analog of classical algorithm for finding ground and excited states. Mcardle et al. [107] proposed to use a variational quantum circuit for the simulation of imaginary time evolution and calculate the ground state energy of the chemical system. Benedetti et al. [108] concentrated on making the variational simulation of time evolution as efficient as possible in terms of quantum hardware resources. With respect to imaginary time evolution, their approach significantly diminishes hardware requirements.

4 Challenges

Although VQAs have been proven to be advantageous in solving specific problems in both theory and practice, there are still many challenges in the field. This section focuses on three challenges for VQAs and potential solutions.

4.1 Trainability

The current challenges in trainability of VQAs are mainly caused by barren plateau. The barren plateau was first proposed in 2018 by McClean et al. [109],which means that when the ansatz performs stochastic parameter initialization, the gradient variance of the cost function which is in the average sense of sampling $\mathrm{Var}\left[\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_k}\right]$ will decrease exponentially with the problem size. This phenomenon causes the optimization surface of the problem to become very flat. In order to determine the optimization direction, the number of measurements required to calculate the accuracy will be very large. Eventually it will be difficult to find the global minimum for either gradient-based or non-gradient-based optimization methods. The mathematical reason behind this phenomenon is that unstructured ansatz satisfies the property of 2-designs [110] when randomly initializing the parameters. Theoretical studies on mitigating barren plateaus are also centered around this core property.

Grant et al. [111] demonstrated an initialization strategy theoretically and experimentally. The key idea was to randomly select some initial parameter values and then specify the remaining parameters so that the entire circuit consists of a series of unit arrays. This could reduce the randomness in the circuit thereby destroying the 2-designs property to improve trainability. Skolik et al. [112] proposed a layered strategy using a number of initial layers to train and then sequentially add circuit structure and layers. Experimental results show that, compared with the full circuit training, this strategy is helpful to alleviate the barren plateau phenomenon due to the low depth of the circuit and the small number of parameters in one step training. In addition to the above strategy of designing initialization training scheme, redesigning the cost function to express it in the form of a local cost function (measuring only some of the qubits at the same time) [10] has also been shown to be effective in dealing with the barren plateau phenomenon. However, whether the cost function of many algorithms exists in such a form is still undetermined. Besides, the tensor network structure has proven to be a powerful variational ansatz, which can alleviate the barren plateau phenomenon [113, 114].

In addition, it has been shown that GQML can alleviate barren plateau in QML. The symmetrization procedure in GQML diminishes the expressiveness of the ansatz by decreasing the count of free parameters. Furthermore, it modifies the dynamical Lie algebra linked with the ansatz generators, a factor closely intertwined with barren plateau phenomena.

4.2 Hardware noise

Hardware noise has many effects on the implementation of the algorithm, such as it may slow down the training process, it may affect the final value of the optimal cost, and so on. In general, quantum noise can be categorized into coherent and incoherent noise. Coherent noise may be caused by poor hardware calibration, resulting in $U(\theta + \delta)$ being actually executed when $U(\theta)$ is executed. In general, coherent noise produces little effect on classical optimization methods. Incoherent noise tends to have an effect on the overall landscape of the cost function, flattening it or changing the location of the extreme values. Stilck et al. [115] compared the complexity of VQAs and classical algorithms under noisy conditions, and demonstrated that VQAs no longer had advantages over classical algorithms under conditions of large noise. Xue et al. [116] investigated the effects of typical quantum noise channels on QAOA. The experimental results found that the optimization parameters under quantum noise were very close to the ideal optimization parameters, indicating that QAOA is resistant to low-intensity Pauli noise. Endo et al. [117] proposed to combine error mitigation methods with VQA thereby mitigating the effects of hardware noise. Quantum error mitigation (QEM) suppresses physical errors through classical post-processing of measurements.

The core is not to restore the ideal output state, but to restore the ideal measurement statistics.

4.3 Expressibility and entangling capability

The expressibility of ansatzes refers to the size of the unitary variable range that can be expressed when taking all the parameters for the circuit. The stronger the expressibility, the larger the solution space covered by the optimization process. In general, the deeper the circuit, the more expressive it is. But at the same time it brings more serious trainability problems and noise. Entangling capability refers to the ability of the circuit to generate entangled states, depending on the connections between qubits in the circuit and the choice of quantum gates. A higher entangling capability indicates a more complex circuit, which can lead to an increase in computational resource requirements. Therefore, how to balance circuit depth with expressibility, circuit complexity with entangling capability, and how to design ansatzes with stronger expressibility and entangling capability are important challenges for VQAs.

Sim et al. [26] defined and calculated the expressibility and entangling capability of several commonly used anstazes. In addition to this, the authors compared three connection configurations between two quantum gates: nearest-neighbor (NN), circuit-block (CB), and all-to-all (AA), as shown in Fig. 10. The NN configuration is a natural arrangement of two-qubit operations on a linear array of qubits. The CB configuration is a natural arrangement of a series of qubits forming a closed loop. The AA configuration is a complete connection between qubits. The experimental results show that the NN configuration has the worst expressibility, and the adoption of the AA configuration leads to good expressibility and entangling capability, but there are trade-offs in the number of parameters, circuit depth and qubit connectivity. Although the CB configuration is slightly inferior compared to the AA configuration, it provides a less expensive solution.

5 Conclusions and outlook

This review introduces the basic concepts of VQAs, the research progress of such algorithms in several areas and the main challenges that the algorithms are currently facing. It can be found that VQAs have provided effective and potentially advantageous solutions in many fields. However, as a new field, VQAs still have many technical difficulties. Explore how to overcome these bottlenecks as well as the applications of VQAs in more fields will be an important research direction in the future.

In the foreseeable future, there is a probable transition anticipated for VQAs from the initial stages of proposal and development to the phase of implementation. Researchers will strive to incorporate more extensive and realistic problem sets into VQA systems, moving away from simplistic examples. These implementations will integrate various cutting-edge strategies aimed at enhancing the performance of VQAs.

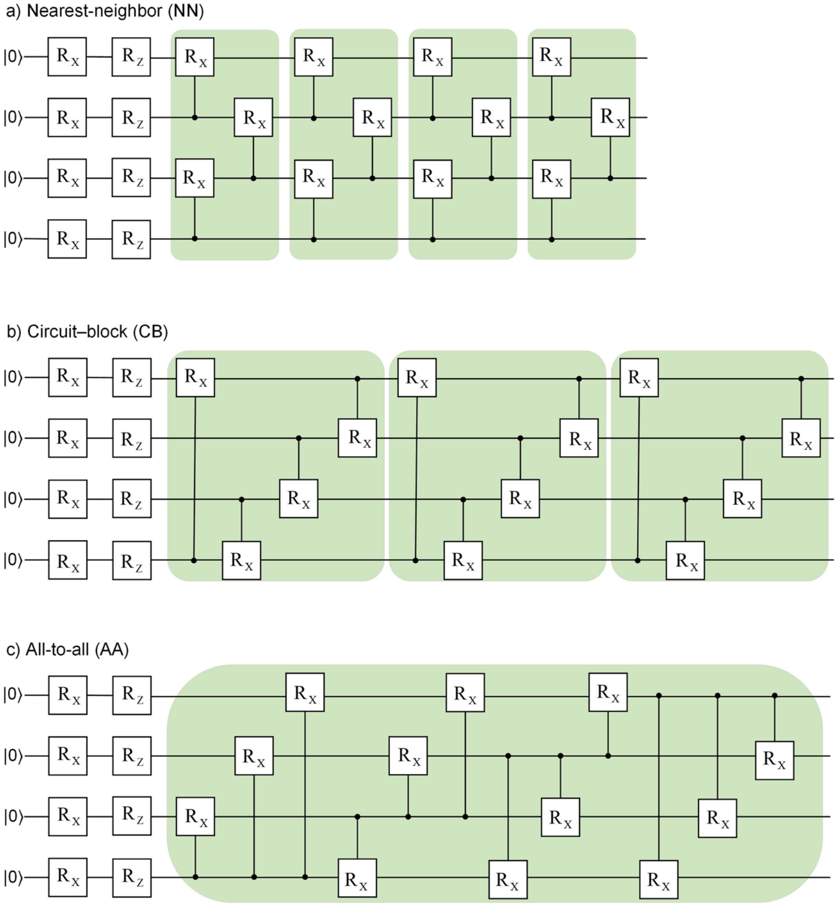

Fig. 10 Three configurations of connections between quantum gates

In the pursuit of achieving quantum advantage, the analytical and heuristic scaling analysis of VQAs will assume growing significance. Anticipated advancements in the future are expected to yield improved methodologies for assessing the scalability of VQAs. These advancements are likely to encompass gradient scaling as well as other scaling factors, including the density of local minima and the configuration of the cost landscape. These foundational findings will serve as valuable guidance in the exploration for quantum advantage. Future developments will also bring about an enhanced VQA toolkit. Quantum-aware optimizers will take advantage of newfound insights into the cost structure. In order to speed up the process of training the parameters in VQAs, these enhanced optimizers will lessen the effects of tiny gradients and prevent local minima. Additionally, the parameter optimization will be further expedited and the testing of VQAs will be streamlined using commercial software packages.

In the more distant future, VQAs will even find use even when the fault-tolerant era arrives. Strategies related to solving the challenges that existed in the NISQ era, such as maintaining lower depth circuits and choosing circuit configurations wisely, may still be useful in the fault-tolerant era.

Acknowledgements This work was funded in part by the Liaoning Provincial Department of Education Research under Grant LJKZ0208, in part by the Scientific Research Foundation for Advanced Talents from Shenyang Aerospace University under Grant 18YB06.

Author Contributions H.Q. and S.X. wrote the main manuscript text. Z.L. , C.G. and A.G. revised the manuscript critically for important intellectual content. All authors reviewed the manuscript.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Steane, A.: Quantum computing. Rep. Prog. Phys. **61**(2), 117 (1998)
2. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
3. Shor, P.W.: Fault-tolerant quantum computation. In: Proceedings of 37th Conference on Foundations of Computer Science, pp. 56–65. IEEE (1996)
4. Devitt, S.J., Munro, W.J., Nemoto, K.: Quantum error correction for beginners. Rep. Prog. Phys. **76**(7), 076001 (2013)
5. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
6. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Phys. Rev. Lett. **103**(15), 150502 (2009)
7. Cerezo, M., Arrasmith, A., Babbush, R., et al.: Variational quantum algorithms. Nat. Rev. Phys. **3**(9), 625–644 (2021)
8. Bharti, K., Cervera-Lierta, A., Kyaw, T.H., et al.: Noisy intermediate-scale quantum algorithms. Rev. Mod. Phys. **94**(1), 015004 (2022)
9. Ran-Yi-Liu, C., Ben-Chi, Z., Zhi-Xin, S., et al.: Hybrid quantum-classical algorithms: foundation, design and applications. Acta Phys. Sin. (2021). https://doi.org/10.7498/aps.70.20210985
10. Cerezo, M., Sone, A., Volkoff, T., et al.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat. Commun. **12**(1), 1791 (2021)
11. Buhrman, H., Cleve, R., Watrous, J., et al.: Quantum fingerprinting. Phys. Rev. Lett. **87**(16), 167902 (2001)
12. Garcia-Escartin, J.C., Chamorro-Posada, P.: Swap test and Hong-Ou-Mandel effect are equivalent. Phys. Rev. A **87**(5), 052330 (2013)
13. Aharonov, D., Jones, V., Landau, Z.: A polynomial quantum algorithm for approximating the jones polynomial. In: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, pp. 427–436 (2006)
14. Taube, A.G., Bartlett, R.J.: New perspectives on unitary coupled-cluster theory. Int. J. Quantum Chem. **106**(15), 3393–3401 (2006)
15. Peruzzo, A., McClean, J., Shadbolt, P., et al.: A variational eigenvalue solver on a photonic quantum processor. Nat. Commun. **5**(1), 4213 (2014)
16. Motta, M., Ye, E., McClean, J.R., et al.: Low rank representations for quantum simulation of electronic structure. npj Quantum Inf. **7**(1), 83 (2021)
17. Matsuzawa, Y., Kurashige, Y.: Jastrow-type decomposition in quantum chemistry for low-depth quantum circuits. J. Chem. Theory Comput. **16**(2), 944–952 (2020)
18. Kivlichan, I.D., McClean, J., Wiebe, N., et al.: Quantum simulation of electronic structure with linear depth and connectivity. Phys. Rev. Lett. **120**(11), 110501 (2018)

19. Setia, K., Bravyi, S., Mezzacapo, A., et al.: Superfast encodings for fermionic quantum simulation. Phys. Rev. Res. **1**(3), 033033 (2019)
20. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. (2014) arXiv preprint arXiv:1411.4028
21. Hadfield, S., Wang, Z., O'gorman, B., et al.: From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. Algorithms **12**(2), 34 (2019)
22. Lloyd, S.: Quantum approximate optimization is computationally universal (2018) arXiv preprint arXiv:1812.11075
23. Morales, M.E., Biamonte, J.D., Zimborás, Z.: On the universality of the quantum approximate optimization algorithm. Quantum Inf. Process. **19**, 1–26 (2020)
24. Wecker, D., Hastings, M.B., Troyer, M.: Progress towards practical quantum variational algorithms. Phys. Rev. A **92**(4), 042303 (2015)
25. Kandala, A., Mezzacapo, A., Temme, K., et al.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature **549**(7671), 242–246 (2017)
26. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv. Quantum Technol. **2**(12), 1900070 (2019)
27. Ballarin, M., Mangini, S., Montangero, S., et al.: Entanglement entropy production in quantum neural networks. Quantum **7**, 1023 (2023)
28. Hubregtsen, T., Pichlmeier, J., Stecher, P., et al.: Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. Quantum Mach. Intell. **3**, 1–19 (2021)
29. McClean, J.R., Boixo, S., Smelyanskiy, V.N., et al.: Barren plateaus in quantum neural network training landscapes. Nat. Commun. **9**(1), 4812 (2018)
30. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th, Anniversary Cambridge University Press, Cambridge (2010)
31. Schuld, M., Bergholm, V., Gogolin, C., et al.: Evaluating analytic gradients on quantum hardware. Phys. Rev. A **99**(3), 032331 (2019)
32. Wierichs, D., Izaac, J., Wang, C., et al.: General parameter-shift rules for quantum gradients. Quantum **6**, 677 (2022)
33. Stokes, J., Izaac, J., Killoran, N., et al.: Quantum natural gradient. Quantum **4**, 269 (2020)
34. Koczor, B., Benjamin, S.C.: Quantum natural gradient generalized to noisy and nonunitary circuits. Phys. Rev. A **106**(6), 062416 (2022)
35. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
36. Vidal, J.G., Theis, D.O.: Calculus on parameterized quantum circuits (2018) arXiv preprint arXiv:1812.06323
37. Nakanishi, K.M., Fujii, K., Todo, S.: Sequential minimal optimization for quantum-classical hybrid algorithms. Phys. Rev. Res. **2**(4), 043158 (2020)
38. McArdle, S., Endo, S., Aspuru-Guzik, A., et al.: Quantum computational chemistry. Rev. Mod. Phys. **92**(1), 015003 (2020)
39. Nielsen, M.A., et al.: The fermionic canonical commutation relations and the Jordan-Wigner transform. School of Physical Sciences The University of Queensland 59 (2005)
40. Bravyi, S.B., Kitaev, A.Y.: Fermionic quantum computation. Ann. Phys. **298**(1), 210–226 (2002)
41. Hempel, C., Maier, C., Romero, J., et al.: Quantum chemistry calculations on a trapped-ion quantum simulator. Phys. Rev. X **8**(3), 031022 (2018)
42. Mitarai, K., Nakagawa, Y.O., Mizukami, W.: Theory of analytical energy derivatives for the variational quantum eigensolver. Phys. Rev. Res. **2**(1), 013129 (2020)
43. Parrish, R.M., Hohenstein, E.G., McMahon, P.L., et al.: Hybrid quantum/classical derivative theory: Analytical gradients and excited-state dynamics for the multistate contracted variational quantum eigensolver (2019) arXiv preprint arXiv:1906.08728
44. O'Brien, T.E., Senjean, B., Sagastizabal, R., et al.: Calculating energy derivatives for quantum chemistry on a quantum computer. npj Quantum Inf. **5**(1), 113 (2019)
45. McClean, J.R., Kimchi-Schwartz, M.E., Carter, J., et al.: Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. Phys. Rev. A **95**(4), 042308 (2017)
46. Nakanishi, K.M., Mitarai, K., Fujii, K.: Subspace-search variational quantum eigensolver for excited states. Phys. Rev. Res. **1**(3), 033062 (2019)

47. Parrish, R.M., Hohenstein, E.G., McMahon, P.L., et al.: Quantum computation of electronic transitions using a variational quantum eigensolver. Phys. Rev. Lett. **122**(23), 230401 (2019)
48. Zhang, F., Gomes, N., Yao, Y., et al.: Adaptive variational quantum eigensolvers for highly excited states. Phys. Rev. B **104**(7), 075159 (2021)
49. Lubasch, M., Joo, J., Moinier, P., et al.: Variational quantum algorithms for nonlinear problems. Phys. Rev. A **101**(1), 010301 (2020)
50. Zhang, J., Ferguson, R., Kühn, S., et al.: Simulating gauge theories with variational quantum eigensolvers in superconducting microwave cavities. Quantum **7**, 1148 (2023)
51. Kandala, A., Temme, K., Corcoles, A.D., et al.: Extending the computational reach of a noisy superconducting quantum processor (2018). arXiv preprint arXiv:1805.04492
52. Chen, M.C., Gong, M., Xu, X., et al.: Demonstration of adiabatic variational quantum computing with a superconducting quantum coprocessor. Phys. Rev. Lett. **125**(18), 180501 (2020)
53. Lee, D., Lee, J., Hong, S., et al.: Error-mitigated photonic variational quantum eigensolver using a single-photon ququart. Optica **9**(1), 88–95 (2022)
54. Zhang, J.N., Arrazola, I., Casanova, J., et al.: Probabilistic eigensolver with a trapped-ion quantum processor. Phys. Rev. A **101**(5), 052333 (2020)
55. Grimsley, H.R., Barron, G.S., Barnes, E., et al.: Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus. npj Quantum Inf. **9**(1), 19 (2023)
56. Farhi, E., Goldstone, J., Gutmann, S., et.al.: Quantum computation by adiabatic evolution. Physics (2000)
57. Egger, D.J., Mareček, J., Woerner, S.: Warm-starting quantum optimization. Quantum **5**, 479 (2021)
58. Tate, R., Farhadi, M., Herold, C., et al.: Bridging classical and quantum with SDP initialized warm-starts for QAOA. ACM Trans. Quantum Comput. **4**(2), 1–39 (2023)
59. Zhou, L., Wang, S.T., Choi, S., et al.: Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. Phys. Rev. X **10**(2), 021067 (2020)
60. Bravyi, S., Kliesch, A., Koenig, R., et al.: Hybrid quantum-classical algorithms for approximate graph coloring. Quantum **6**, 678 (2022)
61. Herrman, R., Lotshaw, P.C., Ostrowski, J., et al.: Multi-angle quantum approximate optimization algorithm. Sci. Rep. **12**(1), 6781 (2022)
62. Amaro, D., Modica, C., Rosenkranz, M., et al.: Filtering variational quantum algorithms for combinatorial optimization. Quantum Sci. Technol. **7**(1), 015021 (2022)
63. Crooks, G.E.: Performance of the quantum approximate optimization algorithm on the maximum cut problem (2018). arXiv preprint arXiv:1811.08419
64. Guerreschi, G.G., Matsuura, A.Y.: Qaoa for max-cut requires hundreds of qubits for quantum speed-up. Sci. Rep. **9**(1), 6903 (2019)
65. Streif, M., Leib, M.: Comparison of QAOA with quantum and simulated annealing (2019). arXiv preprint arXiv:1901.01903
66. Lykov, D., Shaydulin, R., Sun, Y., et al.: Fast simulation of high-depth QAOA circuits. In: Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, pp. 1443–1451 (2023)
67. Harrigan, M.P., Sung, K.J., Neeley, M., et al.: Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nat. Phys. **17**(3), 332–336 (2021)
68. Pagano, G., Bapat, A., Becker, P., et al.: Quantum approximate optimization of the long-range ising model with a trapped-ion quantum simulator. Proc. Natl. Acad. Sci. **117**(41), 25396–25401 (2020)
69. Xiaogang, Q., Xiaoqi, Z., Jianwei, W., et al.: Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. Nat. Photonics **12**, 534–539 (2018)
70. Ebadi, S., Keesling, A., Cain, M., et al.: Quantum optimization of maximum independent set using Rydberg atom arrays. Science **376**(6598), 1209–1215 (2022)
71. Mitarai, K., Negoro, M., Kitagawa, M., et al.: Quantum circuit learning. Phys. Rev. A **98**(3), 032309 (2018)
72. Grant, E., Benedetti, M., Cao, S., et al.: Hierarchical quantum classifiers. npj Quantum Inf. **4**(1), 65 (2018)
73. Schuld, M., Bocharov, A., Svore, K.M., et al.: Circuit-centric quantum classifiers. Phys. Rev. A **101**(3), 032308 (2020)
74. Havlíček, V., Córcoles, A.D., Temme, K., et al.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)

75. Wang, Y., Lin, K.Y., Cheng, S., et al.: Variational quantum extreme learning machine. Neurocomputing **512**, 83–99 (2022)
76. Park, S., Park, D.K., Rhee, J.K.K.: Variational quantum approximate support vector machine with inference transfer. Sci. Rep. **13**(1), 3288 (2023)
77. Benedetti, M., Lloyd, E., Sack, S., et al.: Parameterized quantum circuits as machine learning models. Quantum Sci. Technol. **4**(4), 043001 (2019)
78. Benedetti, M., Garcia-Pintos, D., Perdomo, O., et al.: A generative modeling approach for benchmarking and training shallow quantum circuits. npj Quantum Inf. **5**(1), 45 (2019)
79. Zhu, D., Linke, N.M., Benedetti, M., et al.: Training of quantum circuits on a hybrid quantum computer. Sci. Adv. **5**(10), eaaw9918 (2019)
80. Liu, J.G., Wang, L.: Differentiable learning of quantum circuit born machines. Phys. Rev. A **98**(6), 062324 (2018)
81. Hamilton, K.E., Dumitrescu, E.F., Pooser, R.C.: Generative model benchmarks for superconducting qubits. Phys. Rev. A **99**(6), 062323 (2019)
82. Coyle, B., Mills, D., Danos, V., et al.: The born supremacy: quantum advantage and training of an ising born machine. npj Quantum Inf. **6**(1), 60 (2020)
83. Benedetti, M., Coyle, B., Fiorentini, M., et al.: Variational inference with a quantum computer. Phys. Rev. Appl. **16**(4), 044057 (2021)
84. Bronstein, M., Bruna, J., Cohen, T., etal.: Geometric deep learning: grids, groups, graphs, geodesics, and gauges (2021). arxiv:2104.13478
85. Nguyen, Q.T., Schatzki L., Braccia, P., et.al. Theory for equivariant quantum neural networks (2022). arXiv preprint arXiv:2210.08566
86. Larocca, M., Sauvage, F., Sbahi, F.M., et al.: Group-invariant quantum machine learning. PRX Quantum **3**(3), 030341 (2022)
87. Schatzki, L., Arrasmith, A., Coles, P. J., et al.: Entangled datasets for quantum machine learning (2021). arXiv preprint arXiv:2109.03400
88. Meyer, J.J., Mularski, M., Gil-Fuster, E., et. al.: Exploiting symmetry in variational quantum machine learning (2022). arXiv preprint arXiv:2205.06217
89. Skolik, A., Cattelan, M., Yarkoni, S., et al.: Equivariant quantum circuits for learning on weighted graphs. npj Quantum Inf. **9**(1), 47 (2023)
90. Zheng, H., Li, Z., Liu, J., et al.: Speeding up learning quantum states through group equivariant convolutional quantum ansätze. PRX Quantum **4**(2), 020327 (2023)
91. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial networks. Commun. ACM (2020). https://doi.org/10.1145/3422622
92. Chakrabarti, S., Yiming, H., Li, T., et. al.: Quantum wasserstein generative adversarial networks. Advances in Neural Information Processing Systems 32 (2019)
93. Romero, J., Olson, J.P., Aspuru-Guzik, A.: Quantum autoencoders for efficient compression of quantum data. Quantum Sci. Technol. **2**(4), 045001 (2017)
94. Cao, C., Wang, X.: Noise-assisted quantum autoencoder. Phys. Rev. Appl. **15**(5), 054012 (2021)
95. Bravo-Prieto, C., LaRose, R., Cerezo, M., et. al.: Variational quantum linear solver (2019). arXiv preprint arXiv:1909.05820
96. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nat. Phys. (2014). https://doi.org/10.1038/nphys3029
97. LaRose, R., Tikku, A., O'Neel-Judy, É., et al.: Variational quantum state diagonalization. npj Quantum Inf. **5**(1), 57 (2019)
98. Johnson, P.D., Romero, J., Olson, J., et al.: Qvector: an algorithm for device-tailored quantum error correction (2017). arXiv preprint arXiv:1711.02249
99. Laflamme, R., Miquel, C., Paz, J.P., et al.: Perfect quantum error correcting code. Phys. Rev. Lett. **77**(1), 198 (1996)
100. Xu, X., Benjamin, S.C., Yuan, X.: Variational circuit compiler for quantum error correction. Phys. Rev. Appl. **15**(3), 034068 (2021)
101. Gottesman, D.: Stabilizer codes and quantum error correction. California Institute of Technology (1997)
102. Wang, K., Song, Z., Zhao, X., et al.: Detecting and quantifying entanglement on near-term quantum devices. npj Quantum Inf. **8**(1), 52 (2022)
103. Koczor, B., Endo, S., Jones, T., et al.: Variational-state quantum metrology. New J. Phys. **22**(8), 083038 (2020). https://doi.org/10.1088/1367-2630/ab965e

104. Beckey, J.L., Cerezo, M., Sone, A., et al.: Variational quantum algorithm for estimating the quantum fisher information. Phys. Rev. Res. (2022). https://doi.org/10.1103/physrevresearch.4.013083
105. Lehtovaara, L., Toivanen, J., Eloranta, J.: Solution of time-independent Schrödinger equation by the imaginary time propagation method. J. Comput. Phys. **221**(1), 148–157 (2007)
106. Motta, M., Sun, C., Tan, A.T., et al.: Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. Nat. Phys. **16**(2), 205–210 (2020)
107. McArdle, S., Jones, T., Endo, S., et al.: Variational ansatz-based quantum simulation of imaginary time evolution. npj Quantum Inf. **5**(1), 75 (2019)
108. Benedetti, M., Fiorentini, M., Lubasch, M.: Hardware-efficient variational quantum algorithms for time evolution. Phys. Rev. Res. **3**(3), 033083 (2021)
109. McClean, J.R., Boixo, S., Smelyanskiy, V.N., et al.: Barren plateaus in quantum neural network training landscapes. Nat. Commun. (2018). https://doi.org/10.1038/s41467-018-07090-4
110. Harrow, A.W., Low, R.A.: Random quantum circuits are approximate 2-designs. Commun. Math. Phys. **291**, 257–302 (2009)
111. Grant, E., Wossnig, L., Ostaszewski, M., et al.: An initialization strategy for addressing barren plateaus in parametrized quantum circuits. Quantum **3**, 214 (2019)
112. Skolik, A., McClean, J.R., Mohseni, M., et al.: Layerwise learning for quantum neural networks. Quantum Mach. Intell. **3**, 1–11 (2021)
113. Dborin, J., Barratt, F., Wimalaweera, V., et al.: Matrix product state pre-training for quantum machine learning. Quantum Sci. Technol. **7**(3), 035014 (2022)
114. Martín, E.C., Plekhanov, K., Lubasch, M.: Barren plateaus in quantum tensor network optimization. Quantum **7**, 974 (2023)
115. Stilck França, D., Garcia-Patron, R.: Limitations of optimization algorithms on noisy quantum devices. Nat. Phys. **17**(11), 1221–1227 (2021)
116. Xue, C., Chen, Z.Y., Wu, Y.C., et al.: Effects of quantum noise on quantum approximate optimization algorithm. Chin. Phys. Lett. **38**(3), 030302 (2021)
117. Endo, S., Cai, Z., Benjamin, S.C., et al.: Hybrid quantum-classical algorithms and quantum error mitigation. J. Phys. Soc. Jpn. (2021). https://doi.org/10.7566/jpsj.90.032001

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.