FRONTIERS OF PHYSICS

RESEARCH ARTICLE
Volume 19 / Issue 2 / 21202 / 2024

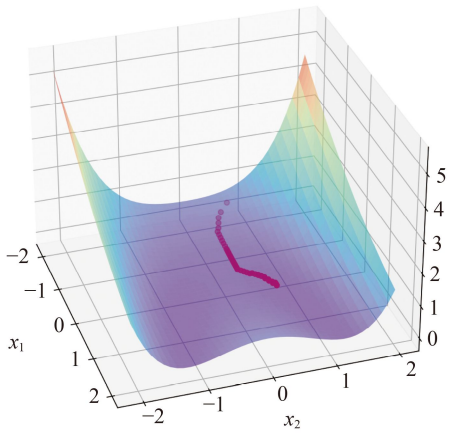
Pure quantum gradient descent algorithm and full quantum variational eigensolver

Ronghang Chen[1,2], Zhou Guang[3], Cong Guo[2], Guanru Feng[2], Shi-Yao Hou[1,2,†]

1 College of Physics and Electronic Engineering, Center for Computational Sciences, Sichuan Normal University, Chengdu 610068, China
2 Shenzhen SpinQ Technology Co., Ltd., Shenzhen 518045, China
3 DEEPROUTE.AI
Corresponding author. E-mail: †hshiyao@sicnu.edu.cn
Received June 28, 2023; accepted September 4, 2023

ABSTRACT

Optimization problems are prevalent in various fields, and the gradient-based gradient descent algorithm is a widely adopted optimization method. However, in classical computing, computing the numerical gradient for a function with d variables necessitates at least $d+1$ function evaluations, resulting in a computational complexity of $O(d)$. As the number of variables increases, the classical gradient estimation methods require substantial resources, ultimately surpassing the capabilities of classical computers. Fortunately, leveraging the principles of superposition and entanglement in quantum mechanics, quantum computers can achieve genuine parallel computing, leading to exponential acceleration over classical algorithms in some cases. In this paper, we propose a novel quantum-based gradient calculation method that requires only a single oracle calculation to obtain the numerical gradient result for a multivariate function. The complexity of this algorithm is just $O(1)$. Building upon this approach, we successfully implemented the quantum gradient descent algorithm and applied it to the variational quantum eigensolver (VQE), creating a pure quantum variational optimization algorithm. Compared with classical gradient-based optimization algorithm, this quantum optimization algorithm has remarkable complexity advantages, providing an efficient solution to optimization problems.The proposed quantum-based method shows promise in enhancing the performance of optimization algorithms, highlighting the potential of quantum computing in this field.



Keywords quantum algorithm, gradient descent, variational quantum algorithm

1 Introduction

Optimization problems are a crucial research area that involves finding parameter values that minimize or maximize an objective function subject to given constraints [1–3]. They have broad-ranging applications in diverse fields, including machine learning [4–6], finance [7–9], quantum chemical calculations [10, 11], and the pharmaceutical industry [12]. Gradient descent is one of the widely used optimization methods in solving optimization [13, 14]. The basic idea of the gradient descent algorithm is to iteratively adjust the model

Higher Education Press

https://doi.org/10.1007/s11467-023-1346-7

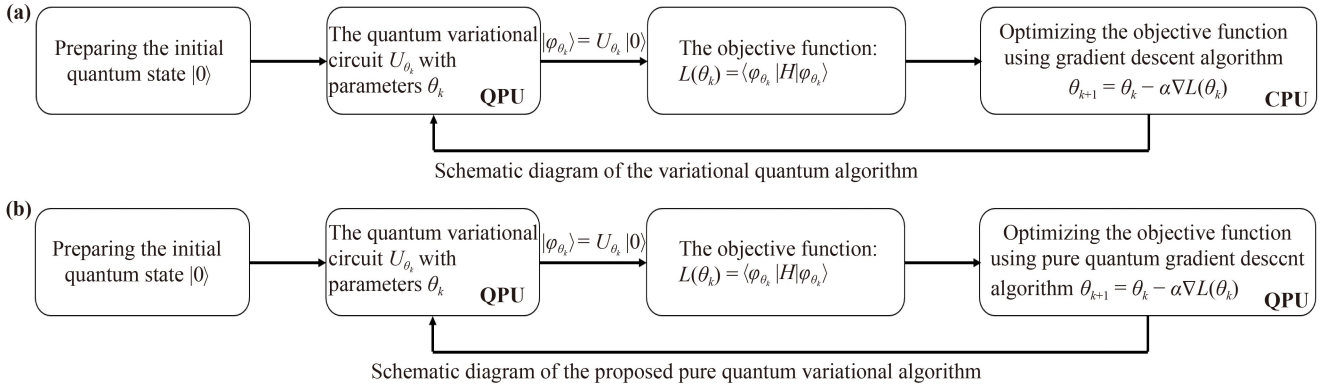


Fig. 1 Diagram of a quantum variational algorithm optimized using gradient descent algorithm. **(a)** The algorithm involves the generation of a parameterized quantum circuit on a quantum computer, followed by optimization of the objective function parameters on a classical computer. This iterative process involves the exchange of data between the quantum and classical computers. **(b)** The algorithm utilizes a pure quantum gradient estimation approach to evolve the parameterized unitary operator on the quantum circuit, effectively replacing classical gradient algorithm. The process is repeated iteratively until the objective function is minimized. All operations of this algorithm can be efficiently performed on a quantum computer.

parameters to minimize the loss function. Specifically, the algorithm computes the gradient of the loss function for each parameter and then updates the parameters in the opposite direction of the gradient to minimize the loss function as rapidly as possible. This process is repeated multiple times until convergence [15].

With the rapid development of quantum computing, variational quantum algorithm have gained increasing attention as a means to solve optimization problems by adjusting some parameters in a quantum circuit to minimize an objective function [16, 17]. In optimization problems, as illustrated in Fig. 1(a), we first create a quantum circuit with parameters to calculate the objective function. Classical optimization algorithms, such as the gradient descent algorithm, are then utilized to optimize the objective function on a classical computer. The optimized parameters are then fed back into the quantum circuit, and the optimization process continues until convergence [13, 18]. Various classical optimization algorithms, such as the conjugate gradient method [19] and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [20–23], are widely used in variational quantum algorithm. It should be noted that during the optimization process, classical optimization algorithms continuously measure the quantum circuit and input the results into the classical computer for computation, which may affect the algorithm's convergence rate. Moreover, when employing the gradient-based gradient descent algorithm on a classical computer, the numerical gradient calculation is the core of the algorithm, with a complexity of $O(d)$ [24]. As the number of variables d increases, the resource consumption grows rapidly, potentially impacting the optimization algorithm's convergence.

Quantum computers leverage the principles of superposition and entanglement to achieve remarkable acceleration in certain problems compared to classical computers [25–33]. In the context of gradient computation, several quantum gradient calculation methods have been proposed, but they suffer from limitations in terms of applicability or complexity [24, 34–37]. Therefore, in this paper, we propose a pure quantum gradient estimation method that does not require classical computation of the gradient. For a multivariate function, this algorithm can achieve numerical gradient computation with just one oracle calculation, and its complexity is only $O(1)$, which does not consume additional computational resources with an increase in the number of variables. Based on this, we implemented a quantum gradient descent algorithm based on pure quantum gradient estimation. We conducted numerical experiments using this algorithm and applied it to calculate the ground state energy of a small-scale Hamiltonian in VQE [18], as shown in Fig. 1(b). The optimization process of this algorithm does not depend on classical computation of the gradient. The numerical experiments demonstrate that our proposed pure quantum gradient estimation algorithm, quantum gradient descent algorithm, and its application in the Heisenberg model have produced satisfactory results. Furthermore, this pure gradient evolution algorithm has theoretical advantages in complexity, which can accelerate the convergence rate of optimization problems in large-scale problems, providing a more efficient optimization algorithm for optimization problems.

The organizational structure of this paper is as follows: In Section 2, we present a pure quantum gradient estimation algorithm. In Section 3, we describe the quantum gradient descent algorithm which is implemented using the proposed pure quantum gradient estimation algorithm. In Section 4, we perform numerical experiments on the proposed pure quantum gradient estimation algorithm and quantum gradient descent algorithm. In

Section 5, we combine the quantum gradient descent algorithm with the VQE algorithm to calculate the ground state energy of the Heisenberg model to implement the full quantum variational eigensolver(FQVE). We analyze the errors of the algorithm in Section 6 and conclude with our findings in Section 7.

2 Pure quantum gradient estimation

The gradient of a function is widely used to locate extrema and offers an effective solution for certain optimization problems [24]. However, in the case of a d-dimensional variable in a function f, calculating the gradient necessitates at least $d+1$ function evaluations using the equation:

$$\frac{\partial f}{\partial x_i} = \frac{f(\boldsymbol{x} - l\boldsymbol{e}_i) - \boldsymbol{x}}{l}. \tag{1}$$

Here, $\boldsymbol{e}_i$ denotes the ith normalized basis vector [34]. As the number of parameters increases, the resources required to compute the gradient grow, often surpassing the capabilities of classical computers. In optimization problems, the computation of the function is typically the most time-consuming task when calculating gradients. Hence, minimizing the number of function evaluations is critical to improving the efficiency of optimization algorithms [34, 38].

2.1 Quantum algorithm for gradient estimation at zero

We observe the availability of a fast quantum algorithm for computing gradients at zero, as proposed in reference [34]. For both classical and quantum cases, given a sufficiently small value $\boldsymbol{x}$, it holds that

$$f(\boldsymbol{x}) \cong f(\boldsymbol{0}) + \boldsymbol{x}\nabla f. \tag{2}$$

For a function of d variables, the classical computation requires evaluating the function $d+1$ times, whereas the utilization of quantum superposition allows obtaining the gradient values of all variables using just one function evaluation. In this quantum algorithm, for d variables, each variable is encoded using n qubits, a superposition state of $n \times d$ qubits is created as

$$|\boldsymbol{0}\rangle \xrightarrow{H} \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta}\rangle. \tag{3}$$

Then, an oracle is constructed as $O_f = \mathrm{e}^{2\pi \mathrm{i}(N/ml)f(\boldsymbol{x})}$, which is applied to the superposition state resulting in a phase that contains the function as

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta}\rangle \xrightarrow{O_f} \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i}\frac{N}{ml} f(\boldsymbol{x})} |\boldsymbol{\delta}\rangle. \tag{4}$$

Let $\boldsymbol{x} = (l/N)\boldsymbol{\delta}$ (where l is a sufficiently small value), perform the transformation in Eq. (2), ignoring the global phase, resulting in a phase that contains the gradient as

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} \mathrm{e}^{\frac{2\pi \mathrm{i}}{m}[\delta_1 \nabla f_1 + \delta_2 \nabla f_2 + \cdots + \delta_d \nabla f_d]} |\boldsymbol{\delta}\rangle. \tag{5}$$

After applying the Inverse Quantum Fourier Transform, a state containing the gradient can be obtained as

$$\left|\frac{N}{m}(\nabla f)_1\right\rangle \left|\frac{N}{m}(\nabla f)_2\right\rangle \cdots \left|\frac{N}{m}(\nabla f)_d\right\rangle. \tag{6}$$

Finally, measurement in the computational basis can obtain the ∇f of the objective function. In the above equation, d represents the number of variables in the function, n represents the number of qubits, $N = 2^n$, m is an order of magnitude estimate of the actual gradient, $\boldsymbol{\delta}$ are n-bit integers (0 to $N-1$), and l is a sufficiently small parameter [34].

Importantly, this algorithm is limited to computing the gradient of the objective function at zero, and its efficiency depends on the value of the parameter m. Therefore, it may not be suitable for practical applications, such as optimization problems.

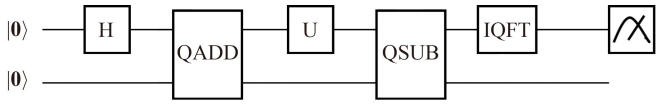


Fig. 2 Schematic diagram of the proposed pure quantum algorithm for gradient estimation. The algorithm is designed to enable direct estimation of the gradient through the use of Hadamard gates (H) to create superposition states, QADD and QSUB gates for entanglement-free quantum addition and subtraction between qubits, an oracle (U) for evolving the gradient, and an Inverse Quantum Fourier Transform (IQFT) for extracting phase information.

2.2 Pure quantum gradient estimation algorithm

We propose a pure quantum gradient estimation algorithm, as shown in Fig. 2, that can efficiently estimate gradients at arbitrary points of the objective function. Our algorithm provides an effective quantum gradient estimation method for optimization problems.

When approximating the objective function, the Taylor series can be expressed for an infinitesimal quantity $(x - x_0)$ as follows: $f(x) \cong f(x_0) + \nabla f(x_0)(x - x_0)$. In our algorithm, we set $\boldsymbol{x_0} = \Delta\boldsymbol{x}$ (compute the gradient at this point), $\boldsymbol{x}_1 = (l/N)\boldsymbol{\delta}'$ (where l is a sufficiently small parameter, and $\boldsymbol{x}_1$ is a sufficiently small quantity, $\boldsymbol{\delta}' = 2^{n+1} - 1, N = 2^n$), and $\boldsymbol{x} = \boldsymbol{x}_1 + \Delta\boldsymbol{x}$. Therefore, we obtain

$$f(\boldsymbol{x}) \cong f(\Delta\boldsymbol{x}) + \boldsymbol{x}_1 \nabla f(\Delta\boldsymbol{x}). \tag{7}$$

Consequently, we only need to construct an oracle

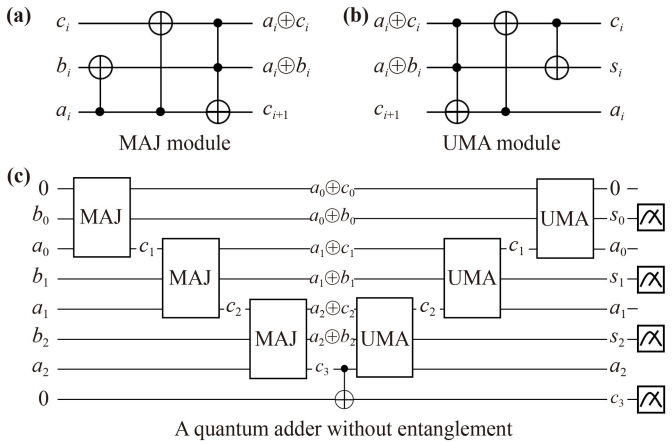


Fig. 3 Schematic diagram of a quantum adder without entanglement. **(a)** The MAJ module performs addition on the input states to obtain the sum of the two states and a carry qubit. **(b)** The UMA module adds the sum and the carry qubit from the MAJ module to achieve the effect of a full adder. **(c)** This quantum adder (QADD) uses MAJ and UMA modules to perform addition operations on the input state, and the output states are not entangled with each other, which can be used for subsequent operations on specific target qubits.

containing $f(\boldsymbol{x})$.

For an objective function with d variables, we first create a superposition state (using quantum superposition, the gradient can be computed for d variables simultaneously):

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}|\boldsymbol{\delta}\rangle. \tag{8}$$

We then use a separable variable quantum adder [39] to add $\Delta\boldsymbol{x}$ to the superposition state, resulting in

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle. \tag{9}$$

The purpose of this operation is to shift the quantum state by $\Delta\boldsymbol{x}$ to compute the gradient at this point. The separable variable quantum adder is used to add $\Delta\boldsymbol{x}$ to the superposition state and ensure that the qubits do not become entangled with each other. For example, the CNOT gate can be considered as an entangled adder. When the input of qubit q_0 is $\sqrt{2}/2\,(|0\rangle+|1\rangle)$, and the input of qubit q_1 is $|0\rangle$, after passing through the CNOT gate, q_0 and q_1 become entangled, and the output of q_1 is influenced by q_0, making it unsuitable for subsequent calculations (shown in Fig. 4). Therefore, we use a quantum adder without entanglement(QADD) shown in Fig. 3 in this case.

In order to compute gradients at arbitrary points ($\Delta\boldsymbol{x}$), we employ a quantum adder without entanglexment (QADD) to shift the quantum state by $\Delta\boldsymbol{x}$ and calculate

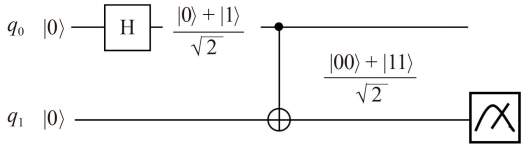


Fig. 4 Schematic diagram of a quantum adder that generates entanglement. The output qubits of this type of quantum adder will become entangled with each other, making it impossible to extract information about the target qubits for subsequent calculations.

the gradient at the desired point. To estimate gradients at arbitrary locations in our algorithm, taking into account the carry after the quantum adder, we construct an oracle $O_f=\mathrm{e}^{2\pi\mathrm{i}(N/ml)f(\boldsymbol{x})}$, where $\boldsymbol{x}=\boldsymbol{x_1}+\Delta\boldsymbol{x},\boldsymbol{x_1}=(l/N)\boldsymbol{\delta'},\boldsymbol{\delta'}=2^{n+1}-1,N=2^n$, acting on the state $|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle$. Taking the example of a 2-qubit system and calculating the derivative at $\Delta\boldsymbol{x}=1$, the diagonal elements of this oracle can be expressed as

$$\mathrm{e}^{2\pi\mathrm{i}(\frac{N}{ml})}\begin{bmatrix} f(0\frac{l}{N}+\Delta x)\\ f(1\frac{l}{N}+\Delta x)\\ f(2\frac{l}{N}+\Delta x)\\ f(3\frac{l}{N}+\Delta x)\\ f(4\frac{l}{N}+\Delta x)\\ f(5\frac{l}{N}+\Delta x)\\ f(6\frac{l}{N}+\Delta x)\\ f(7\frac{l}{N}+\Delta x)\end{bmatrix}.$$

To achieve the desired state for the constructed oracle, we utilize a quantum adder to transform the previous equal-weight superposition state from $|0\rangle+|1\rangle+|2\rangle+|3\rangle$ to $|1\rangle+|2\rangle+|3\rangle+|4\rangle$.

$$\begin{aligned} &O_f\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}|\boldsymbol{\delta}+\Delta x\rangle\\ &=\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0,\boldsymbol{\delta'}=0}^{N-1,2^{n+1}-1}\mathrm{e}^{2\pi\mathrm{i}(\frac{N}{ml})\nabla f(\Delta\boldsymbol{x})\frac{l}{N}\boldsymbol{\delta'}}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle, \end{aligned}$$

can be written as

$$\begin{aligned} &\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})0}|1\rangle+\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})1}|2\rangle\\ &+\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})2}|3\rangle+\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})3}|4\rangle. \end{aligned}$$

Through the utilization of the QADD, we successfully attain the target quantum state required for applying the oracle, facilitating gradient estimation at any desired

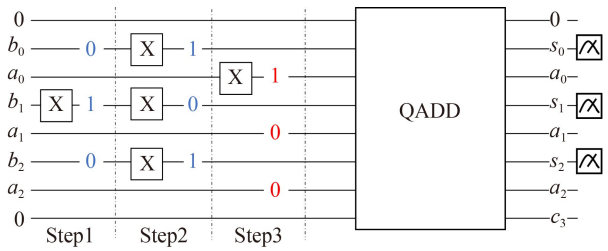


Fig. 5 Schematic diagram of a quantum complement circuit. A quantum subtractor can be regarded as adding a negative number using an adder, so it is only necessary to apply the adder to the complement of the subtrahend. For example, to get the complement of −010, in Step1, construct the quantum state of the value qubits of −010. In Step2, take the one's complement of the value qubits. In Step3, construct the quantum state of $|1\rangle$. Then, use a quantum adder to add $|1\rangle$ to the complemented value qubits obtained in Step2 to obtain the two's complement of −010, which is represented as $|s_2 s_1 s_0\rangle$ in the value qubits.

point in the objective function. By implementing subsequent operations, we are able to effectively realize gradient estimation at arbitrary points within the objective function.

Next, we construct an oracle containing $f(\boldsymbol{x})$ in phase and apply it to the shifted superposition state above, obtaining

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle \xrightarrow{O_f} \frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}(\frac{N}{ml})f(\boldsymbol{x})}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle. \tag{10}$$

Using the above formula and ignoring the global phase, we obtain the quantum state containing the gradient information in phase:

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}/m[\delta_1(\nabla f)_1+\delta_2(\nabla f)_2+\cdots+\delta_d(\nabla f)_d]}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle. \tag{11}$$

To extract the phase information, we need to shift the superposition state back to the initial equal-weight position, so we need a separable quantum subtractor (shown in Fig. 5), which gives

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}[\delta_1(\nabla f)_1+\delta_2(\nabla f)_2+\cdots+\delta_d(\nabla f)_d]}|\boldsymbol{\delta}\rangle. \tag{12}$$

Taking the example of a 2-qubit system and calculating the derivative at $\Delta x = 1$, the quantum state

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0,\boldsymbol{\delta}'=0}^{N-1,2^{n+1}-1}\mathrm{e}^{2\pi\mathrm{i}(\frac{N}{ml})\nabla f(\Delta\boldsymbol{x})\frac{l}{N}\boldsymbol{\delta}'}|\boldsymbol{\delta}+\Delta\boldsymbol{x}\rangle,$$

after passing through the quantum subtractor can be expressed as

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0,\boldsymbol{\delta}'=0}^{N-1,2^{n+1}-1}\mathrm{e}^{2\pi\mathrm{i}(\frac{N}{ml})\nabla f(\Delta\boldsymbol{x})\frac{l}{N}\boldsymbol{\delta}'}|\boldsymbol{\delta}\rangle,$$

in other words,

$$\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})0}|0\rangle+\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})1}|1\rangle + \mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})2}|2\rangle+\mathrm{e}^{\frac{2\pi\mathrm{i}}{m}\nabla f(\Delta\boldsymbol{x})3}|3\rangle.$$

By utilizing the quantum subtractor, we successfully transform the quantum state into a form suitable for performing Quantum Fourier Transform.

Finally, applying the Inverse Quantum Fourier Transform yields the quantum state containing the gradient information:

$$\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_1\right\rangle\cdots\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_i\right\rangle\cdots\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_d\right\rangle. \tag{13}$$

where the subscription i refers to the i-th component of the gradient, which is a vector. In the above formula, d represents the number of variables in the function, n represents the number of qubits, $N=2^n$, m is an estimate of the order of magnitude of the actual gradient and can be taken as 2^n, and l is an extremely small parameter.

In the above process, the gradient estimation is for the case where $\Delta\boldsymbol{x}$ is positive. For the case where $\Delta\boldsymbol{x}$ is negative, we can simply subtract a positive value $\Delta\boldsymbol{x}^+$ (by adding the two's complement of $\Delta\boldsymbol{x}$ using a quantum adder) from the initial superposition state, resulting in

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}|\boldsymbol{\delta}-\Delta\boldsymbol{x}^+\rangle, \tag{14}$$

apply the oracle to add the gradient information to the phase:

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}/m[\delta_1(\nabla f)_1+\delta_2(\nabla f)_2+\cdots+\delta_d(\nabla f)_d]}\left|\boldsymbol{\delta}-\Delta\boldsymbol{x}^+\right\rangle, \tag{15}$$

and then add this positive value $\Delta\boldsymbol{x}^+$:

$$\frac{1}{\sqrt{N^d}}\sum_{\boldsymbol{\delta}=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}/m[\delta_1(\nabla f)_1+\delta_2(\nabla f)_2+\cdots+\delta_d(\nabla f)_d]}|\boldsymbol{\delta}\rangle. \tag{16}$$

Finally, the Inverse Quantum Fourier Transform can be applied to obtain a quantum state that contains the gradient information at the negative value of $\Delta\boldsymbol{x}$:

$$\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_1\right\rangle\cdots\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_i\right\rangle\cdots\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_d\right\rangle.$$

During the derivation presented above, we note that

Algorithm 1: Pure quantum gradient estimation algorithm

Input:
$f(\boldsymbol{x})$: A objective function for which the gradient is to be computed;
m: One parameter estimates the order of magnitude of the gradient;
n: The number of qubits required to encode each variable;
N: $N = 2^n$;
d: The quantity of variables;
l: A parameter that is small enough;
$\Delta\boldsymbol{x}$: The point at which the gradient wants to be estimated;
$\boldsymbol{x_1}$: $\boldsymbol{x_1} = \frac{l}{N}\boldsymbol{\delta'}, \boldsymbol{\delta'} = 0, 1, \cdots, 2^{n+1} - 1$;
$\boldsymbol{x}$: $\boldsymbol{x} = \boldsymbol{x_1} + \Delta\boldsymbol{x}$;

1. **Register:** Utilize n-qubit input registers $|x_1\rangle$, $|x_2\rangle$, ..., $|x_d\rangle$, with each qubit initialized to $|\boldsymbol{0}\rangle$;
2. **Init:** Apply a Hadamard gate to each qubit in the input registers, resulting in

$$|\boldsymbol{0}\rangle \xrightarrow{H} \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta}\rangle.$$

3. **Add:** Apply a quantum adder with separable variables to add $\Delta\boldsymbol{x}$ to the superposition state:

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta}\rangle \xrightarrow{QADD} \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta} + \Delta\boldsymbol{x}\rangle.$$

4. **Oracle:** Construct an oracle $O_f = e^{2\pi i(N/ml)f(\boldsymbol{x})}$ that contains the function $f(\boldsymbol{x})$ in its phase. Apply the oracle to the input registers, resulting in

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} |\boldsymbol{\delta} + \Delta\boldsymbol{x}\rangle \xrightarrow{O_f} \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i}(\frac{N}{ml})f(\boldsymbol{x})} |\boldsymbol{\delta} + \Delta\boldsymbol{x}\rangle.$$

Ignoring the global phase, the input registers are now approximately in the state:

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} \mathrm{e}^{\frac{2\pi \mathrm{i}}{m}[\delta_1(\nabla f)_1 + \delta_2(\nabla f)_2 + \cdots + \delta_d(\nabla f)_d]} |\boldsymbol{\delta} + \Delta\boldsymbol{x}\rangle.$$

5. **Sub:** After applying a quantum subtractor to subtract $\Delta\boldsymbol{x}$ from the superposition state, the input registers are left in the state:

$$\frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{\delta}=0}^{N-1} \mathrm{e}^{\frac{2\pi \mathrm{i}}{m}[\delta_1(\nabla f)_1 + \delta_2(\nabla f)_2 + \cdots + \delta_d(\nabla f)_d]} |\boldsymbol{\delta}\rangle.$$

6. **IQFT:** Apply the Inverse Quantum Fourier Transform (IQFT) to the input registers, the quantum state can be expressed as

$$\left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_1\right\rangle \cdots \left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_i\right\rangle \cdots \left|\frac{N}{m}(\nabla f(\Delta\boldsymbol{x}))_d\right\rangle.$$

7. **Measure:** Measure the quantum state obtained after applying IQFT in the computational basis to obtain the components of ∇f;

Result: A quantum state obtains the components of ∇f.

the evolved gradient should have the same sign as the parameter m. When m is fixed to be 2^n, the gradient obtained from the algorithm is positive. However, when the actual gradient is negative, due to the periodicity of the exponential function in the Inverse Quantum Fourier Transform, for a negative integer k [34],

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \mathrm{e}^{-2\pi \mathrm{i} j/N|k|} |j\rangle \longrightarrow |N - |k|\rangle,$$

therefore,

$$\frac{1}{\sqrt{N}} \sum_{\delta=0}^{N-1} \mathrm{e}^{-2\pi \mathrm{i}|\frac{\nabla f}{m}|\delta} |\delta\rangle \longrightarrow \left|N - |\frac{N}{m}\nabla f|\right\rangle. \tag{17}$$

Thus, the algorithm can also obtain a quantum state related to the negative gradient, which, at $m = 2^n$, can be understood as the two's complement of the negative gradient without the sign qubit.

3 Quantum gradient descent algorithm

We have implemented quantum gradient descent using the pure quantum gradient estimation algorithm. In the classical gradient descent process [19, 24, 40], $x_i^{k+1} = x_i^k - \alpha\nabla f(x_i^k)$, where α is the learning rate, x_i^k is the kth iteration of the ith variable. In our quantum implementation, for a positive component $\nabla f(x_i^k)$ of the gradient, we can obtain $|x_i^{k+1}\rangle = |x_i^k - \alpha((N/m)\nabla f)\rangle$, which can be understood as normal gradient descent, where we subtract a positive number from x_i^k. For a negative component $\nabla f(x_i^k)$ of the gradient, we can obtain $\left|x_i^{k+1}\right\rangle = |x_i^k - \alpha(N - |(N/m)\nabla f|)\rangle$. This can be interpreted as subtracting the two's complement notation of the negative gradient from x_i^k. In other words, adding the absolute value of the negative number. Thus, we have $\left|x_i^{k+1}\right\rangle = |x_i^k + \alpha(|\nabla f(x_i^k)|)\rangle$, which is no different from normal gradient descent. Therefore, whether our algorithm obtains the true gradient or the two's complement notation of the true gradient, it does not affect the calculation process of gradient descent.

In the gradient descent formula $x_i^{k+1} = x_i^k - \alpha\nabla f(x_i^k)$, quantum multiplication is required, which can be challenging due to the large number of qubits needed for constructing a quantum multiplier. Therefore, in the case of limited qubits, we propose a hybrid approach that combines classical multiplication with pure quantum gradient estimation in the gradient descent process. Specifically, we use the pure quantum gradient estimation algorithm to estimate the gradient value of the objective function, and then update the parameters using classical methods. However, due to the introduction of classical computation, we need to perform measurements after each gradient estimation and update the parameters using classical computers, which may affect the calculation

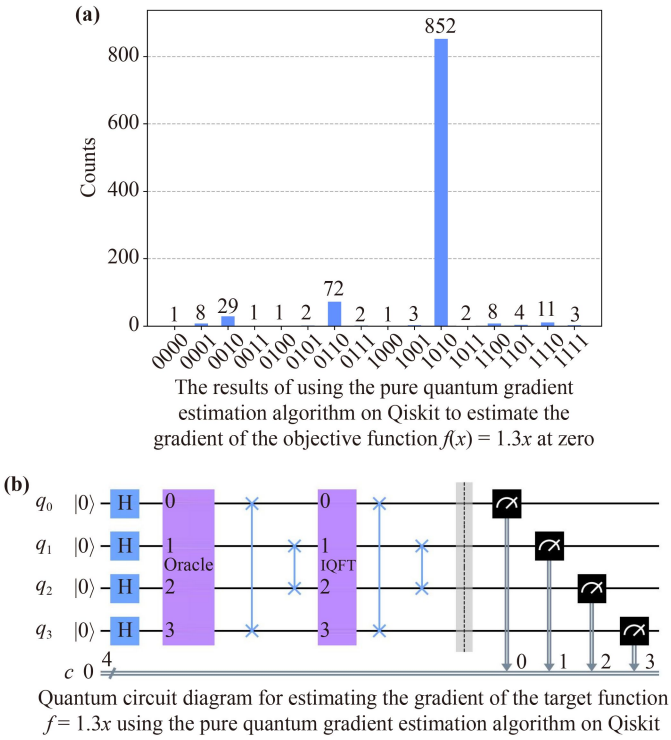


Fig. 6 Quantum circuit diagram and results for estimating the gradient at zero.

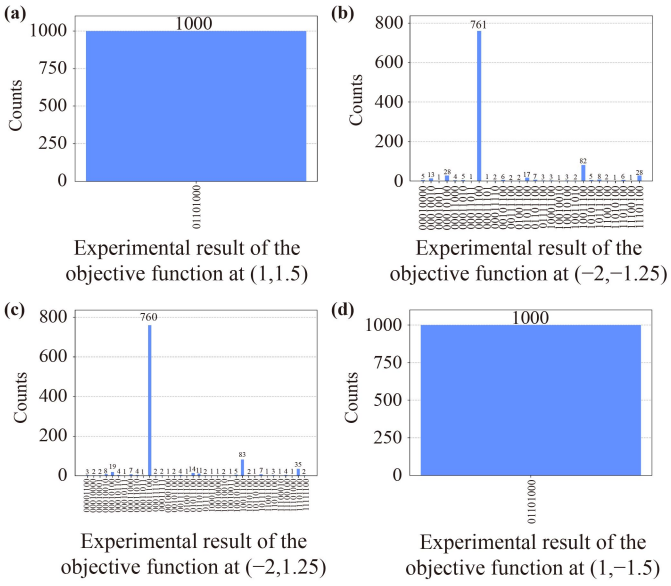


Fig. 7 Using the pure quantum gradient estimation algorithm on Qiskit to estimate the gradient of the target function $f(x_1, x_2) = 0.1(x_1 + x_2^2)^2 + 0.1(1 + x_2^2)^2$ at various initial points.

speed to some extent. In the long run, we expect the emergence of large-scale fault tolerant quantum computers, which will enable our quantum gradient descent algorithm to complete calculations without repeated measurements and improve its performance.

Analyzing the results of numerical simulations in Section 4.3, we found that although the algorithm partially uses classical computers, it can still effectively find the optimal value of the objective function, which can be used in practical applications.

4 Results

We conducted numerical simulations of the pure quantum gradient estimation algorithm and quantum gradient descent algorithm using Qiskit [41]. Our results showed that these algorithms were effective in estimating the gradients of the objective function, finding the optimal values of the objective function.

4.1 Gradient estimation at point zero

In this study, we used Qiskit [41] to numerically simulate the pure gradient ascent. We encoded the objective function $f(x) = 1.3x$ in the ground state using four qubits, with two qubits encoding the decimal values. We set the parameters $N = 4$ and $m = 4$ and conducted 1000 experiments using the *qasm_simulator* simulator. From Fig. 6, we observed that the state $|(N/m)\nabla f\rangle$ had a probability of 85.2% of being in the state $|0101\rangle$, which corresponds to an estimated gradient of 1.25 for the objective function. The small difference between the estimated and actual gradients suggests that the algorithm was successful in estimating the objective function's gradient.

4.2 Gradient estimation at any point

In this section, we encoded the objective function $f(x_1, x_2) = 0.1(x_1 + x_2^2)^2 + 0.1(1 + x_2^2)^2$ in the ground state using four qubits, with two qubits encoding the decimal values for each variable. It transforms a binary string x of length n into a quantum state $|x\rangle = |i_x\rangle$ with n qubits, where $|i_x\rangle$ is the computational basis state. The parameters $N = 4$ and $m = 4$ were used, and we conducted 1000 experiments using the *qasm_simulator* [41]. The gradients of the objective function at $(1, 1.5)$, $(-2, -1.25)$, $(-2, 1.25)$, and $(1, -1.5)$ are shown in Fig. 7. The results indicate that the final quantum states for each point had a probability of 100% of being in the state $|00010110\rangle$, 76.1% of being in the state $|00111000\rangle$, 76% of being in the state $|00111000\rangle$, and 100% of being in the state $|00010110\rangle$, respectively. Combining with symbolic analysis, the estimated gradient values for each point were $(-0.25, 1.5)$, $(-0.75, -2)$, $(-0.75, 2)$, and $(-0.25, -1.5)$. The small differences between the estimated and actual gradient values suggest that the quantum gradient descent algorithm successfully estimated the gradients of the objective function.

4.3 Numerical results of gradient descent algorithm

In this study, we utilized the *qasm_simulator* [41] to

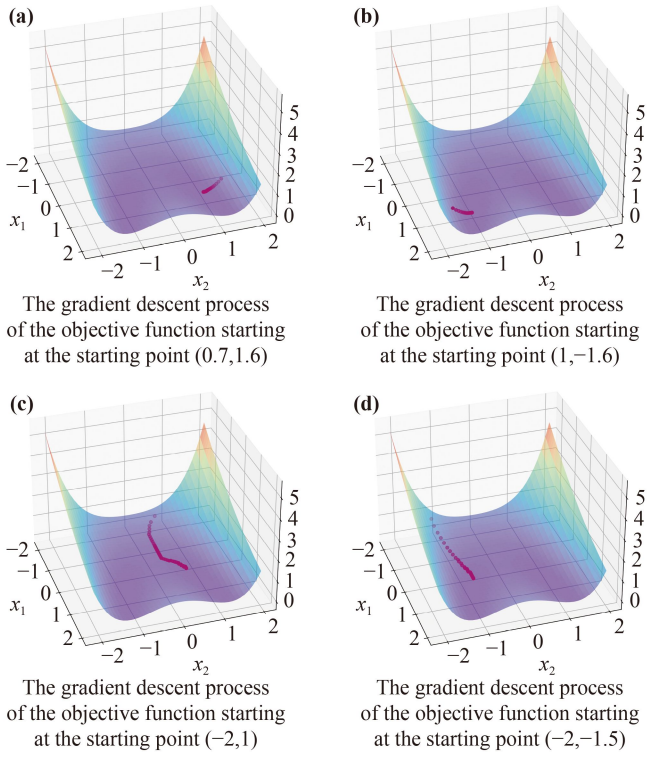


Fig. 8 The results of quantum gradient descent of the objective function $f(x_1, x_2) = 0.1(x_1 + x_2^2)^2 + 0.1(1 + x_2^2)^2$ at various starting points using this pure quantum gradient descent algorithm on Qiskit.

implement gradient descent. Our numerical experiments demonstrate that the pure quantum gradient estimation algorithm-based quantum gradient descent algorithm can effectively locate the extremum of the objective function. This provides a firm foundation for applying quantum gradient descent to optimization problems.

For the objective function $f(x_1, x_2) = 0.1(x_1 + x_2^2)^2 + 0.1(1 + x_2^2)^2$, Fig. 8 shows the iteration process of quantum gradient descent from different starting points. All four gradient descent processes used four qubits (including two qubits encoding decimals) to encode data. The step size $\alpha = 0.05$ was used in Figs. 8(a) and (b), where the starting point of Fig. 8(a) was $(x_1, x_2) = (0.7, 1.6)$ and the extremum found by the extended algorithm was $(x_1^{'}, x_2^{'}) = (0.825, 0.994)$, with an actual gradient of $(-0.033, 0.06)$. The starting point of Fig. 8(b) was $(x_1, x_2) = (1, -1.6)$, and the extremum found by the extended algorithm was $(x_1^{'}, x_2^{'}) = (1.09, -1.05)$, with an actual gradient of $(-0.0025, -0.048)$.

The step size $\alpha = 0.3$ was used in Figs. 8(c) and (d), where the starting point of Fig. 8(c) was $(x_1, x_2) = (-2, 1)$, and the extremum found by the extended algorithm was $(x_1^{'}, x_2^{'}) = (0.14, 0.625)$, with an actual gradient of $(-0.05, -0.09)$. The starting point of Fig. 8(d) was $(x_1, x_2) = (-2, -1.5)$, and the extremum found by the extended algorithm was $(x_1^{'}, x_2^{'}) = (0.2875, -0.75)$, with an actual gradient of $(-0.055, 0.04875)$.

The actual gradients of the extremum points found by the quantum gradient descent algorithm based on pure quantum gradient estimation in all four processes were very small, indicating that the gradient at those points could be approximated as zero. Therefore, it can be concluded that this algorithm successfully found the extremum of the objective function from different starting points.

5 Full quantum variational eigensolver

Variational quantum eigensolver (VQE) is a quantum-classical hybrid algorithm with enormous potential for application on NISQ quantum devices [13, 18, 42–44]. The objective of VQE is to find the ground state and ground state energy of a given Hamiltonian H. In classical computational physics, the variational method is commonly used to estimate the ground state energy of a given Hamiltonian H: parameterize the wave function as $|\psi\rangle = |\psi(\boldsymbol{\theta})\rangle$, update $\boldsymbol{\theta}$ to minimize the expectation value $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ until convergence. On quantum computers, VQE parameterizes the wavefunction with a quantum circuit $U(\boldsymbol{\theta})$ applied to the initial state $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$, and then optimizes $\boldsymbol{\theta}$ to minimize the expectation value $E(\boldsymbol{\theta}) = \langle\mathbf{0}|U^{\dagger}(\boldsymbol{\theta})HU(\boldsymbol{\theta})|\mathbf{0}\rangle$, until convergence [18].

Gradient-based gradient descent is a commonly used optimization method in the VQE. In VQE, the gradient can be directly calculated using the parameter-shift rule [45, 46]:

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \theta_i} = (\langle H\rangle_{\boldsymbol{\theta}_i^+} - \langle H\rangle_{\boldsymbol{\theta}_i^-})/2,$$

where $\boldsymbol{\theta}_i^{\pm} = \boldsymbol{\theta} \pm \boldsymbol{e}_i$, $\boldsymbol{e}_i$ is the ith unit vector in the parameter space [18]. Hardware-efficient ansatz [43], unitary coupled clustered ansatz [43, 47], and Hamiltonian variational ansatz [48, 49] are common choices for $U(\boldsymbol{\theta})$.

This article applies the quantum gradient descent algorithm to VQE to find the ground state energy of the 2-qubit Heisenberg model and compares it with the gradient-based traditional gradient descent algorithm. The Hamiltonian can be written as

$$H_h = X_1 \otimes X_2 + Y_1 \otimes Y_2 + Z_1 \otimes Z_2,$$

where X_i, Y_i, Z_i are the Pauli operators on the ith qubit. The ansatz is shown in Fig. 9, the initial parameter θ is a random number between 0 and π, and the learning rate $\alpha = 0.25$.

Specifically, we replace the traditional gradient descent algorithm based on gradients with the pure quantum gradient estimation algorithm and quantum gradient descent algorithm to obtain the full quantum variational eigensolver(FQVE).

If we want to implement the FQVE algorithm on a fully quantum circuit and introduce our pure quantum gradient estimation algorithm, as shown in Eq. (10), we

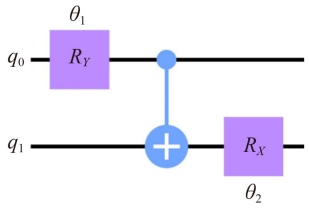


Fig. 9 The ansatz for 2-qubit VQE. θ_1, θ_2 are the parameters to be optimized.

need to construct an oracle on the phase that contains the objective function, that is, we need to construct such an oracle:

$$O_f : \mathrm{e}^{2\pi \mathrm{i}(\frac{N}{ml})\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle}.$$

We notice that the problem can be addressed using the techniques, such as block encoding technique [36, 50], Hamiltonian simulation [36, 51], described in Appendix A. However, even for small-scale model problems mentioned in this paper, the method in Appendix A requires a large number of qubits (more than 50 qubits), which makes it challenging to perform real experiments on current quantum computers or simulators. Therefore, similar to the general VQE, in this paper, we use the conventional VQE method to obtain the expected value of the Hamiltonian and then optimize it using our pure quantum gradient descent algorithm. Nevertheless, our FQVE algorithm performs similarly to the general VQE algorithm on this problem. Furthermore, we look forward to the emergence of large-scale quantum computers, where our FQVE algorithm has an advantage over the general VQE algorithm in theory, running on fully quantum computers.

We encode the parameters using four qubits, with two qubits for decimal representation. We compare the performance of the method to the classical gradient descent algorithm, and present the raw data results in Fig. 10. Our experiments demonstrate that, after sufficient iterations, both the quantum and classical gradient descent algorithms can find the ground state energy of this model. Interestingly, we observe that the quantum gradient descent algorithm performs almost as well as the classical gradient descent algorithm for small-scale parameters in VQE. However, for large-scale parameters (N), the quantum gradient descent algorithm (complexity $O(1)$) has a significant advantage over the classical gradient descent algorithm (complexity $O(N)$) in terms of complexity.

6 The impact of numerical errors

In Section 4, we conducted numerical experiments and observed that our algorithm produced a gradient of 1.25 for the objective function $f = 1.3x$, encoded with two

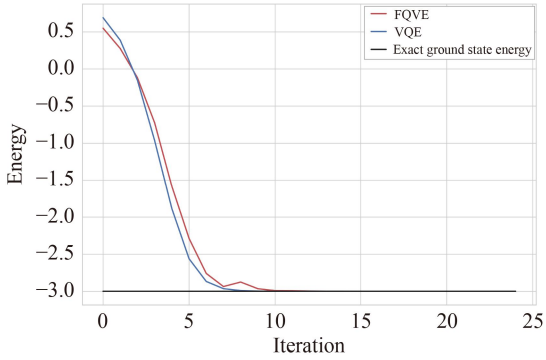


Fig. 10 Results of the full quantum variational eigensolver. The black line represents the exact ground state energy of the model, which is –3. The red line represents the optimization process using the full quantum variational eigensolver, while the blue line represents the optimization process using variational quantum eigensolver. Both optimization methods converge to –2.99 after sufficient iterations.

Algorithm 2: Full Quantum Variational Eigensolver

Input:
$\boldsymbol{\theta}$: randomly initialized parameters;
H: hamiltonian of the model;
R: number of iterations;

1. **Register:** each qubit is initialized to $|\mathbf{0}\rangle$;
2. **if** $k < R$ *or the objective function does not converge* **then**
3. **Anstaz:** generate a quantum circuit using the parameters $\boldsymbol{\theta}$: $U^k_{\boldsymbol{\theta}}$;
4. **Circuit:** generate a circuit $\left|\psi^k_{\boldsymbol{\theta}}\right\rangle$ using the ansatz $U^k_{\boldsymbol{\theta}}$:
$$\left|\psi^k_{\boldsymbol{\theta}}\right\rangle = U^k_{\boldsymbol{\theta}}|\mathbf{0}\rangle;$$
5. **Objective function:** the objective function can be expressed as $L^k_{\boldsymbol{\theta}} = \left\langle\psi^k_{\boldsymbol{\theta}}\right| H \left|\psi^k_{\boldsymbol{\theta}}\right\rangle$;
6. **Optimization:** compute the gradient of the parameters $\boldsymbol{\theta}$ using the pure quantum gradient estimation algorithm, and update $\boldsymbol{\theta}$ using the quantum gradient descent algorithm,
$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha\nabla f(\boldsymbol{\theta}^k);$$
7. **else**
8. Break;
9. **end**

Result: ground state energy of the model.

qubits to represent the decimal number. As illustrated in Fig. 11, increasing the number of qubits used for encoding the decimal numbers improved the precision of the algorithm's output for the same objective function. This is because the precision of numerical representation in a quantum state increases with the number of qubits used for encoding the decimal numbers. Therefore, increasing the number of qubits used for encoding decimal numbers could improve the precision of the algorithm, given suffi-

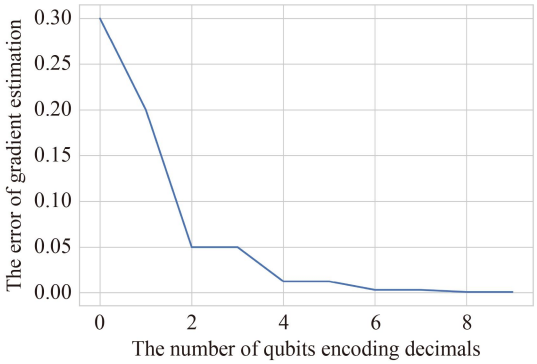


Fig. 11 The relationship between the number of qubits encoding decimals and the error of gradient estimation in the algorithm. When m is fixed to the same value ($m = 2$), the more qubits used to encode decimals, the smaller the error and higher the accuracy.

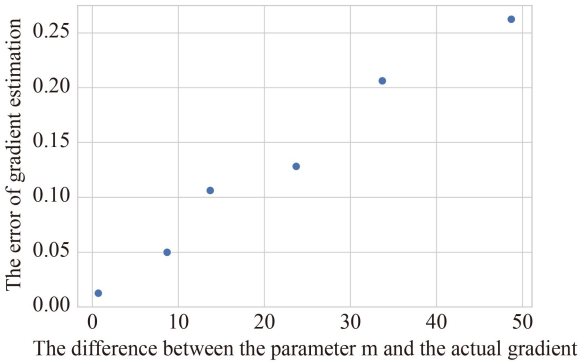


Fig. 12 The relationship between the size of the parameter m and the error of gradient estimation in the algorithm. When the number of qubits used to encode decimals is fixed, the closer the parameter m is to the actual gradient, the smaller the error.

cient computing resources.

Furthermore, we noted that the parameter m is another factor that influences the estimation of the gradient. The parameter m provides an approximate estimate of the magnitude of the actual gradient, which is typically larger than the actual gradient. If m is smaller than the actual gradient, the quantum state $|(N/m)\nabla f(\Delta \boldsymbol{x})\rangle$ obtained by the algorithm will be larger than the value N that can be represented by n qubits. And as shown in Fig. 12, for the objective function $f = 1.3x$, the estimation provided by the algorithm becomes more accurate as the value of m approaches the actual gradient.

7 Conclusion

In this work, we propose an effective pure quantum gradient estimation algorithm for estimating numerical gradients. The algorithm has a theoretical query complexity of $O(1)$ [34], which provides significant advantages over classical algorithm with a computational complexity of $O(d)$. Numerical simulation results using Qiskit [41] demonstrate that the algorithm can successfully estimate the gradient of a complex objective function. Our error analysis shows that increasing the number of encoding quantum bits can improve the accuracy of the algorithm to some extent. Building upon this algorithm, we implement a quantum gradient descent algorithm, which, despite being limited by hardware qubit counts, performs well in finding the extremum of a relatively complex objective function, as demonstrated by our numerical simulations. When solving for the ground state energy of a small-scale Hamiltonian [14, 18], the performance of our full quantum variational eigensolver is comparable to that of variational quantum eigensolver. However, for large-scale Hamiltonian, our algorithm has a significant theoretical advantage. These results suggest that our algorithm provides a more efficient quantum optimization method than classical optimization algorithm, which theoretically can improve convergence speed and increase the precision of optimization problems.

Declarations The authors declare that they have no competing interests and there are no conflicts.

Acknowledgements R. Chen and S. Y. Hou are supported by the National Natural Science Foundation of China under Grant No. 12105195.

Appendix A: Converting a probability oracle to a phase oracle

To combine our quantum gradient descent algorithm with VQE to form a new full quantum variational eigensolver, it is necessary to put the objective function $L_{\boldsymbol{\theta}} = \langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle$ into the phase as shown in Eq. (10) [36], that is, we need to construct such an oracle:

$$O_f : \mathrm{e}^{2\pi \mathrm{i}(\frac{N}{ml})\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle}.$$

We note that the objective function can be transformed into a probability oracle, and then the block encoding technique can be used to transform this probability oracle into a block encoding of a diagonal matrix containing probabilities [36, 50]. Next, the Hamiltonian simulation can be used to implement putting this probability oracle into the phase [51]. Then, our proposed pure quantum gradient estimation algorithm can be used to estimate the gradient of the objective function.

Regarding the Hamiltonian of the model $H_h = \sum_{j=1}^{M} a_j U_j$, we construct $\mathrm{prepareW}{:}|0\rangle \longrightarrow \sum_j \sqrt{a_j}|j\rangle$ and $\mathrm{selectH}{:} = \sum_j |j\rangle\langle j| \otimes U_j$. Under the assumption that $\sum_j a_j = 1$ note that

$$\begin{aligned}&\langle 0|\langle\psi_{\boldsymbol{\theta}}|\text{prepareW}^{\dagger}(\text{selectH})\text{perpareW}|0\rangle|\psi_{\boldsymbol{\theta}}\rangle\\&=\sum_j\sum_k\sqrt{a_j a_k}\langle k|j\rangle\otimes\langle\psi_{\boldsymbol{\theta}}|U_j|\psi_{\boldsymbol{\theta}}\rangle\\&=\langle\psi_{\boldsymbol{\theta}}|\sum_j a_j U_j\,|\psi_{\boldsymbol{\theta}}\rangle\\&=\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle.\end{aligned}\tag{A1}$$

We can use a technique similar to the Hadamard test as shown in Fig. A1 to transform the objective function into a probability [36]. We can obtain: $HC(U)H|0\rangle|\psi_{\boldsymbol{\theta}}\rangle$, where $C(U)$ is a controlled unitary operation,then we can obtain

$$\begin{aligned}&\frac{H(|0\rangle|\psi_{\boldsymbol{\theta}}\rangle+|1\rangle U|\psi_{\boldsymbol{\theta}}\rangle)}{\sqrt{2}}\\&=|0\rangle\left(\frac{(1+U)|\psi_{\boldsymbol{\theta}}\rangle}{2}\right)+|1\rangle\left(\frac{(1-U)|\psi_{\boldsymbol{\theta}}\rangle}{2}\right).\end{aligned}\tag{A2}$$

Therefore, by measuring the first qubit, we obtain a probability of $(1-\langle\psi_{\boldsymbol{\theta}}|U|\psi_{\boldsymbol{\theta}}\rangle)/2$ for the outcome to be 1. The combination with Eq. (A1) leads to a probability of $(1-\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle)/2$ for the outcome to be 1. Thus we obtain a probability oracle O_p for the expected ground state energy of the model, resulting in

$$O_p|\mathbf{0}\rangle|\theta\rangle\longrightarrow(\sqrt{p}|1\rangle|\psi_0\rangle+\sqrt{1-p}|0\rangle|\psi_1\rangle)|\theta\rangle,$$

where $|\psi_0\rangle$ and $|\psi_1\rangle$ are arbitrary quantum states, and $p=(1-\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle)/2$ [36].

We can observe that

$$(\langle\mathbf{0}|\otimes I)(O_p^{\dagger}(Z\otimes I)O_p)(|\mathbf{0}\rangle\otimes I)=\text{diag}(1-2p),$$

$(O_p^{\dagger}(Z\otimes I)O_p)$ is a block encoding of a diagonal matrix with diagonal elements of $(1-2p)$ [36, 50]. Finally, we use the Hamiltonian simulation technique [51] to transfer this block encoding into the phase to implement the phase oracle in Eq. (10). Then we use our proposed pure quantum gradient estimation algorithm to estimate the gradient of the objective function.

By utilizing the various techniques mentioned above, we can put the objective function in VQE into the phase of our proposed pure quantum gradient estimation algorithm, resulting in a full quantum variational eigensolver.

Appendix B: More details of constructing the oracle

Here, we will provide further details on constructing the oracle mentioned in Appendix A. Our objective is to construct such an oracle: $O_f:\mathrm{e}^{2\pi\mathrm{i}(N/ml)\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle}$.

We initiate by employing a method analogous to the Hadamard test to acquire a probability oracle pertaining to the objective function. As shown in Fig. B1, we employ the operation Prep and Tuned to prepare the variational quantum state $|\psi_{\boldsymbol{\theta}}\rangle$, where Prep corresponds to the identity operation and Tuned corresponds to $\prod_{j=1}^{d}\mathrm{e}^{-\mathrm{i}H_j x_j}$. Subsequently, utilizing a circuit similar to the Hadamard test shown in Fig. A1, we can obtain Eq. (A1) and Eq. (A2). Consequently, we successfully derive a probability oracle pertaining to the expectation value of the ground state of the model [36]

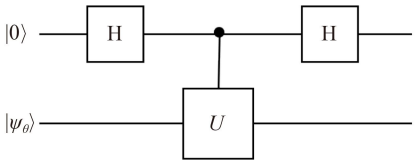


Fig. A1 Quantum circuit diagram for converting the ground state energy of the model into probability. Here, $|0\rangle$ is an auxiliary qubit, $|\psi_{\boldsymbol{\theta}}\rangle$ is the state obtained by the ansatz in the VQE algorithm, $C(U)$ is a controlled unitary operation. Using a method similar to the Hadamard test, we can obtain a probability of $(1-\langle\psi_{\boldsymbol{\theta}}|U|\psi_{\boldsymbol{\theta}}\rangle)/2$ for the outcome to be 1 on the auxiliary qubit.

$$O_p|\mathbf{0}\rangle|\theta\rangle\longrightarrow(\sqrt{p}|1\rangle|\psi_0\rangle+\sqrt{1-p}|0\rangle|\psi_1\rangle)|\theta\rangle.$$

Then observing this probability oracle, we readily deduce the relationship:

$$\begin{aligned}&(\langle\mathbf{0}|\otimes I)(O_p^{\dagger}(Z\otimes I)O_p)(|\mathbf{0}\rangle\otimes I)\\&=(\sqrt{p}\langle\psi_0|\langle 1|+\sqrt{1-p}\langle\psi_1|\langle 0|)Z(\sqrt{p}|1\rangle|\psi_0\rangle\\&\quad+\sqrt{1-p}|0\rangle|\psi_1\rangle)\\&=\text{diag}(1-2p),\end{aligned}$$

implying that $(O_p^{\dagger}(Z\otimes I)O_p)$ constitutes the block encoding of $\text{diag}(1-2p)$ [36, 50].

Finally, it remains to subject this block encoding to Hamiltonian simulation. Numerous approaches are available to achieve an optimal Hamiltonian simulation when provided with a Hamiltonian block encoding [51, 52].

For a given Hamiltonian $H=(\langle G|\otimes I)U(|G\rangle\otimes I)$, we can utilize controlled-U gates and their inverses to simulate the Hamiltonian. We can construct

$$\begin{aligned}W=&((2|G'\rangle\langle G'|-I)_a\otimes I_s)SU'\\=&\bigoplus_{\lambda}\begin{pmatrix}\lambda & -\sqrt{1-|\lambda|^2}\\ \sqrt{1-|\lambda|^2} & \lambda\end{pmatrix}_{\lambda}\\=&\bigoplus_{\lambda}\mathrm{e}^{-\mathrm{i}Y_{\lambda}\theta_{\lambda}},\end{aligned}$$

using G and U, where S is an arbitrary unitary operation. Specifically, if we let controlled-U be represented as $V_1=|0\rangle\langle 0|\otimes I+|1\rangle\langle 1|\otimes U^{\dagger}$, $V_2=|0\rangle\langle 0|\otimes U+|1\rangle\langle 1|\otimes I$, then we can set $U'=V_1V_2=|0\rangle\langle 0|\otimes U+|1\rangle\langle 1|\otimes U^{\dagger}$ and choose $|G'\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)G$. We can choose $S=(|0\rangle\langle 1|+|1\rangle\langle 0|)\otimes I$ to construct W using these operations [51]. In this paper, $H=(\langle\mathbf{0}|\otimes I)(O_p^{\dagger}(Z\otimes I)O_p)(|\mathbf{0}\rangle\otimes I)=\text{diag}(1-2p)$, $|G\rangle=|\mathbf{0}\rangle$ and $U=O_p^{\dagger}(Z\otimes I)O_p$.

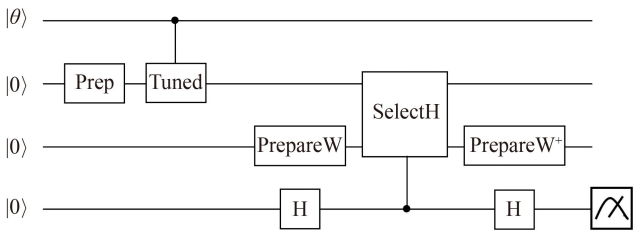


Fig. B1 Quantum circuit diagram for transformation of ground state energy into probability. The probability of measuring 1 in the ancillary qubit is $(1-\langle\psi_{\boldsymbol{\theta}}|H_h|\psi_{\boldsymbol{\theta}}\rangle)/2$.

Utilizing the obtained W in single-ancilla quantum signal processing shown in Ref. [51], with W, $H=\langle G|U|G\rangle=(\langle\mathbf{0}|\otimes I)(O_p^{\dagger}(Z\otimes I)O_p)(|\mathbf{0}\rangle\otimes I)$, and real polynomials $A(\lambda)$ and $C(\lambda)$ of degree $Q/2$ and opposite parity satisfying the following conditions: $A(0)=1;\forall\subseteq[-1,1],A^2(\lambda)+C^2(\lambda)\leq 1$ as input, we can achieve the standard form encoding of $A[H]+\mathrm{i}C[H]$ [51].

This allows us to precisely encode any function $A[H]+\mathrm{i}C[H]$ using $2Q$ queries, where $A(\lambda)$ and $C(\lambda)$ are bounded polynomials of opposite parity and degree Q. Hamiltonian simulation is accomplished by selecting a suitable polynomial of degree Q to approximate $A(\lambda)+\mathrm{i}C(\lambda)=\mathrm{e}^{-\mathrm{i}\lambda t}$ [51].

Building upon the aforementioned principles, we are able to effectively map the ground state energy expectation value, which serves as the objective function $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ in the VQE algorithm, onto a phase. This transformation can be successfully integrated into the pure quantum gradient estimation algorithm that we have proposed.

Appendix C: Complexity analysis

For the pure quantum gradient estimation algorithm, we employ each gradient estimate only once with the oracle. Consequently, the query complexity of the algorithm remains $O(1)$. In comparison to standard classical methods, achieving exponential time savings can be achieved with just a single oracle query [24]. However, it is essential to acknowledge that in order to facilitate arbitrary gradient estimation, the algorithm necessitates the use of a quantum adder to shift quantum states. As a consequence, a minimum of $d\times(2n+2)$ qubits is required to represent a function with d-dimensional variables, where each variable is encoded using n qubits.

Building upon the foundation of the pure quantum gradient estimation algorithm, we introduce the quantum gradient descent algorithm. In contrast to classical gradient descent algorithms, our quantum gradient descent algorithm replaces the gradient estimation part with the pure quantum gradient estimation algorithm. This key distinction is the source of our algorithm's advantage. The remaining parts of the algorithm are consistent with classical gradient descent, resulting in equivalent complexity between our quantum gradient descent algorithm and the classical counterpart. Optimization algorithms are inherently iterative, and often, analyzing the computational workload of each iteration is relatively straightforward [53]. However, there is no well-defined representation for the overall complexity of optimization algorithms. In this paper, the complexity of the classical gradient descent algorithm is represented as $O(d\times l)$, where d represents the computational effort for each gradient computation, and l is the number of iterations. Due to the $O(1)$ query complexity for each gradient computation in the quantum gradient descent algorithm, the overall query complexity for l iterations is $O(l)$.

When applying the pure quantum gradient estimation algorithm to quantum gradient descent, similar to the iterative process in classical gradient descent, the quantum gradient descent algorithm also requires the use of a quantum multiplier. For the multiplication of two states, each consisting of n qubits, it can be understood as performing n^2 AND operations followed by $n-1$ addition operations. Therefore, for a d-dimensional variable (with each variable encoded using n qubits), when utilizing the quantum gradient descent algorithm, a minimum of $d\times(2n^2+n+2)$ qubits are needed.

In the context of block encoding, assuming A is an operation involving s-qubit and U is an operator on $(s+a)$-qubit, if the condition $\|A-\alpha(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)\|\leq\varepsilon$ is satisfied, U is denoted as the (α,a,ε) block encoding of A. In the context of this paper, we can use a technique similar to the Hadamard test as shown in Fig. A1 to transform the objective function into a probability [36]. We can obtain a probability oracle O_p for the expected ground state energy of the model, resulting in:

$$O_p|\mathbf{0}\rangle|\theta\rangle\longrightarrow(\sqrt{p}|1\rangle|\psi_0\rangle+\sqrt{1-p}|0\rangle|\psi_1\rangle)|\theta\rangle.$$

Observe that A corresponds to $(\langle\mathbf{0}|\otimes I)(O_p^{\dagger}(Z\otimes I)O_p)(|\mathbf{0}\rangle\otimes I)=\mathrm{diag}(1-2p)$ and U to $(O_p^{\dagger}(Z\otimes I)O_p)$. Consequently, by employing a 2 ancillary qubits, we can effectively achieve the block encoding of the desired expectation value [36, 50].

Regarding the simulation of block-encoded Hamiltonians, assuming U represents the block encoding of Hamiltonian H, we can utilize 2 additional qubits. Through the application of controlled-U or its inverse gates, we can achieve an ε-precision approximation of the Hamiltonian simulation $\mathrm{e}^{\mathrm{i}tH}$. This simulation approach maintains a computational complexity of $O(|t|+\log(1/\varepsilon))$ [36, 51].

References

1. E. Farhi and J. Goldstone, A quantum approximate optimization algorithm, arXiv: 1411.4028 (2014)
2. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser,

Quantum computation by adiabatic evolution, arXiv: quant-ph/0001106 (2000)
3. G. G. Guerreschi and A. Y. Matsuura, Qaoa for max-cut requires hundreds of qubits for quantum speed-up, *Sci. Rep.* 6(1), 6903 (2019)
4. M. A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015
5. M. Schuld and I. Sinayskiy, The quest for a quantum neural network, arXiv: 1408.7005 (2014)
6. V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-enhanced machine learning, *Phys. Rev. Lett.* 117(13), 130501 (2016)
7. T. Sakuma, Application of deep quantum neural networks to finance, arXiv: 2011.07319 (2020)
8. R. Orus, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, arXiv: 1807.03890v2 (2018)
9. D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State-of-the-art and future prospects, *IEEE Trans. Quant. Eng.* 1, 3101724 (2020)
10. R. P. Feynmen, Forces in molecules, *Phys. Rev. Lett.* 56, 340 (1939)
11. D. A. Fedorov, M. J. Otten, S. K. Gray, and Y. Alexeev, *Ab initio* molecular dynamics on quantum computers, *J. Chem. Phys.* 154(16), 164103 (2021)
12. V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. M. McGinnity, Quantum neural network-based EEG filtering for a brain–computer interface, *IEEE Trans. Neural Netw. Learn. Syst.* 25(2), 278 (2014)
13. A. Peruzzo, J. McClean, P. Shadbolt, et al., A variational eigenvalue solver on a quantum processor, arXiv: 1304.3061 (2013)
14. S. Wei, H. Li, and G. Long, A full quantum eigensolver for quantum chemistry simulations, *Research* 2020, 1486935 (2020)
15. S. Ruder, An overview of gradient descent optimization algorithms, arXiv: 1609.04747 (2016)
16. J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantumclassical algorithms, *New J. Phys.* 18(2), 023023 (2016)
17. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, arXiv: 2012.09265 (2020)
18. S. Y. Hou, G. Feng, Z. Wu, H. Zou, W. Shi, J. Zeng, C. Cao, S. Yu, Z. Sheng, X. Rao, B. Ren, D. Lu, J. Zou, G. Miao, J. Xiang, and B. Zeng, Spinq gemini: A desktop quantum computing platform for education and research, *EPJ Quantum Technol.* 8(1), 20 (2021)
19. G. Yuan, T. Li, and W. Hu, A conjugate gradient algorithm and its application in large-scale optimization problems and image restoration, *J. Inequal. Appl.* 2019(1), 247 (2019)
20. C. G. Broyden, The convergence of a class of double-rank minimization algorithms (1): General considerations, *IMA J. Appl. Math.* 6(1), 76 (1970)
21. R. Fletcher, A new approach to variable metric algorithms, *Comput. J.* 13(3), 317 (1970)
22. D. Goldfarb, A family of variable-metric methods derived by variational means, *Math. Comput.* 24(109), 23 (1970)
23. D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comput.* 24(111), 647 (1970)
24. P. Gao, K. Li, S. Wei, J. Gao, and G. Long, Quantum gradient algorithm for general polynomials, *Phys. Rev. A* 103(4), 042403 (2021)
25. M. Nielsen and I. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010
26. D. P. DiVincenzo, Quantum computation, *Science* 270(5234), 255 (1995)
27. J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* 2, 79 (2018)
28. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp 124–134
29. T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Realization of a scalable Shor algorithm, *Science* 351(6277), 1068 (2016)
30. L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* 79(2), 325 (1997)
31. G. L. Long, Grover algorithm with zero theoretical failure rate, *Phys. Rev. A* 64(2), 022307 (2001)
32. A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* 103(15), 150502 (2009)
33. B. Duan, J. Yuan, C. H. Yu, J. Huang, and C. Y. Hsieh, A survey on HHLalgorithm: From theory to application in quantum machine learning, *Phys. Lett. A* 384(24), 126595 (2020)
34. S. P. Jordan, Fast quantum algorithm for numerical gradient estimation, *Phys. Rev. Lett.* 95(5), 050501 (2005)
35. R. Wiersema, D. Lewis, D. Wierichs, J. Carrasquilla, and N. Killoran, Here comes the $SU(n)$: multivariate quantum gates and gradients, arXiv: 2303.11355 (2023)
36. A. Gilyén, S. Arunachalam, and N. Wiebe, Optimizing quantum optimization algorithms via faster quantum gradient computation, in: Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms, 2019, pp 1425–1444
37. J. Li, General explicit difference formulas for numerical differentiation, *J. Comput. Appl. Math.* 183(1), 29 (2005)
38. W. H. Press, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes in C, Cambridge University Press, 1992
39. Y. Li and M. Dou, A quantum addition operation method, device, electronic device and storage medium, *Origin Quantum* (2021)
40. C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, Algorithm 778: L-BFGS-B, *ACM Trans. Math. Softw.* 23(4), 550 (1997)
41. G. Aleksandrowicz, et al., Qiskit: An open-source framework for quantum computing, 2019
42. R. M. Parrish, E. G. Hohenstein, P. L. McMahon, and

T. J. Martínez, Quantum computation of electronic transitions using a variational quantum eigensolver, *Phys. Rev. Lett.* 122(23), 230401 (2019)
43. A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardwareefficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* 549(7671), 242 (2017)
44. C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Quantum chemistry calculations on a trapped-ion quantum simulator, *Phys. Rev. X* 8(3), 031022 (2018)
45. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, *Phys. Rev. A* 98(3), 032309 (2018)
46. M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* 99(3), 032331 (2019)
47. J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, Generalized unitary coupled cluster wave functions for quantum computation, *J. Chem. Theory Comput.* 15(1), 311 (2019)
48. D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, *Phys. Rev. A* 92(4), 042303 (2015)
49. R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring entanglement and optimization within the hamiltonian variational ansatz, *PRX Quantum* 1(2), 020319 (2020)
50. A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, arXiv: 1806.01838 (2018)
51. G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, arXiv: 1610.06546 (2016)
52. A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, arXiv: 1202.5822 (2012)
53. L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, edited by Y. Lechevallier and G. Saporta, Physica-Verlag HD, Heidelberg, 2010, pp 177–186