Random coordinate descent: A simple alternative for optimizing parameterized quantum circuits

Zhiyan Ding,[1,*,†] Taehee Ko,[2,*,‡] Jiahao Yao,[1,§] Lin Lin,[1,3,4,∥] and Xiantao Li[5,**]

[1]*Department of Mathematics, University of California, Berkeley, CA 94720, USA*

[2]*School of Computational Sciences, Korea Institute for Advanced Study, 02455 Seoul, South Korea*

[3]*Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

[4]*Challenge Institute for Quantum Computation, University of California, Berkeley, CA 94720, USA*

[5]*Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA*

(Received 10 November 2023; accepted 11 June 2024; published 8 July 2024)

Variational quantum algorithms rely on the optimization of parameterized quantum circuits in noisy settings. The commonly used back-propagation procedure in classical machine learning is not directly applicable in this setting due to the collapse of quantum states after measurements. Thus, gradient estimations constitute a significant overhead in a gradient-based optimization of such quantum circuits. This paper introduces a random coordinate descent algorithm as a practical and easy-to-implement alternative to the full gradient descent algorithm. This algorithm only requires one partial derivative at each iteration. Motivated by the behavior of measurement noise in the practical optimization of parameterized quantum circuits, this paper presents an optimization problem setting that is amenable to analysis. Under this setting, the random coordinate descent algorithm exhibits the same level of stochastic stability as the full gradient approach, making it as resilient to noise. The complexity of the random coordinate descent method is generally no worse than that of the gradient descent and can be much better for various quantum optimization problems with anisotropic Lipschitz constants. Theoretical analysis and extensive numerical experiments validate our findings.

DOI: 10.1103/PhysRevResearch.6.033029

I. INTRODUCTION

Variational quantum algorithms have emerged as a promising application for near-term quantum devices, addressing various computational challenges with enhanced efficiency [1,2]. These algorithms encompass several notable approaches, such as the variational quantum eigensolver [3], the variational quantum simulation [4], the quantum approximate optimization algorithm [1,5,6], and quantum machine learning [7–10]. They are designed to operate in a hybrid quantum-classical fashion [11,12]. In these algorithms, the quantum component involves the implementation of parameterized quantum gate operations. By performing measurements, a cost function (and optionally, its gradient) is obtained as the output. The classical computational procedure then. utilizes an iterative method to produce updates for the parameters, which are subsequently leveraged to refine and reprogram the quantum circuits. This iterative process continues until convergence is achieved, forming a feedback loop that continues to improve the algorithm's performance.

In variational quantum algorithms, the optimizable parameters are defined within parameterized quantum circuits (PQCs) [13–16]. A PQC is a sequence of unitary operators represented by parameterized quantum gates that can be readily implemented on a quantum computer. Assuming we are working in an n-qubit Hilbert space, a parameterized quantum circuit can be expressed as follows:

$$U(\boldsymbol{\theta}) = \prod_{j=1}^{J} U_j(\boldsymbol{\theta}_j) W_j. \tag{1}$$

Here $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{J}$ are the parameters that we need to optimize, $U_j(\boldsymbol{\theta}_j) \in \mathbb{C}^{2^n \times 2^n}$ are the parameterized unitary operators, and $W_j \in \mathbb{C}^{2^n \times 2^n}$ are fixed unitary operators. For instance, a simple example of a PQC consisting only of one-qubit Pauli rotation operators takes the form

$$U_j(\boldsymbol{\theta}_j) = \bigotimes_{m=1}^{M} e^{-i\theta_{j,k_{j,m}}\sigma_{j,k_{j,m}}},$$

where $\sigma_{j,k_{j,m}} \in \mathbb{C}^{2\times 2}$ is a single-qubit Pauli matrix that acts on $k_{j,m}$-th qubit, $\theta_{j,k_{j,m}}$ represents one of the parameters in $\boldsymbol{\theta}$, and W_j's can be used to represent quantum gates that do not require parametrization, such as the controlled-NOT (CNOT) gate.

Let d be the dimension of the parameters, and we write $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_d)$. We then optimize the parameter $\boldsymbol{\theta}$ by

[*] Ding and Ko are co-first authors with equal contribution.

[†] Contact author: zding.m@math.berkeley.edu

[‡] Contact author: kthmomo@kias.re.kr;

[§] Contact author: jiahao@math.berkeley.edu

[∥] Contact author: linlin@math.berkeley.edu

[**] Contact author: xiantao.li@psu.edu

minimizing a properly chosen cost function $f(\boldsymbol{\theta})$. As an example, the variation quantum eigensolvers (VQE) finds the smallest eigenvalue (ground-state energy) and its corresponding eigenvector (ground state) of a given Hamiltonian matrix H by minimizing the energy of the state:

$$\theta^* = \mathrm{argmin}_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \mathrm{argmin}_{\boldsymbol{\theta}} \langle U(\boldsymbol{\theta})\psi_0|H|U(\boldsymbol{\theta})\psi_0\rangle. \quad (2)$$

Here $|\psi_0\rangle \in \mathbb{C}^{2^n}$ is a predetermined initial state that can be easily prepared on a quantum computer. For each given $\boldsymbol{\theta}$, $U(\boldsymbol{\theta})$ is implemented on a quantum computer to evolve $|\psi_0\rangle$, and the corresponding energy $f(\boldsymbol{\theta})$ and its gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$ can be consequently obtained with measurements. By solving the optimization problem (2), the minimum value gives an approximation to the smallest eigenvalue of H, while $U(\boldsymbol{\theta}^*)|\psi_0\rangle$ approximates the corresponding eigenvector.

A. Problem setup

Although the problem of optimizing parameters in VQAs resembles classical optimization problems in machine learning, there exist key differences, particularly in how the cost function is evaluated and the level of accuracy that can be obtained for function and gradient evaluations. First, quantum circuits used for estimating partial derivatives in various directions are typically different. This is predominantly because there is no straightforward method (in parallel to backpropagation) to estimate the entire gradient at once, given the inherent nature of quantum states. The predominant method for computing partial derivatives in a PQC is called the parameter-shift rule [17–19], which can only be applied to evaluate one component of the partial derivatives at a time. As a result, the estimation of the gradient, $\nabla f(\boldsymbol{\theta})$, typically incurs a cost that is d times greater than the cost associated with merely estimating a single partial derivative, $\partial_i f(\boldsymbol{\theta})$.

Second, the evaluation of any given quantity, a function value or a partial derivative, requires measurement from quantum computers and is subject to measurement noise. We note that this noise is associated with a finite sampling space. For example, a measurement of the Hamiltonian in (2), which is defined in a finite-dimensional Hilbert space, yields one of its eigenvalues corresponding to the ansatz. Thus, with an increased number of samples or measurements, the central limit theorem suggests that the distribution of the sample average of the function value or the partial derivative can be approximated by a Gaussian distribution, and as a result, the accuracy of function and gradient evaluations can be relatively low. Therefore, the optimization algorithm must be designed to be resilient to measure noise.

In an idealized scenario, we may assume that both the function value and the partial derivatives incorporated into the optimization routine are subject to some Gaussian noise. However, the magnitude of corresponding noises can differ up to a constant, especially in situations where the parameter shift rule is applicable (see Ref. [20]). With this consideration, the problem of optimizing PQCs can be stated as follows:

Problem 1 (Optimizing parameterized quantum circuits). Finding an efficient algorithm to solve the optimization problem,

$$\boldsymbol{\theta}^* = \mathrm{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} f(\boldsymbol{\theta}), \quad (3)$$

under the following assumptions:

(1) The cost of evaluating a partial derivative scales linearly with that of a function value.

(2) Every evaluation of the function and partial derivative is susceptible to Gaussian noise:

$$\begin{aligned} f(\boldsymbol{\theta}) &\to f(\boldsymbol{\theta}) + N\big(0, \sigma_1^2(\boldsymbol{\theta})\big), \\ \partial_i f(\boldsymbol{\theta}) &\to \partial_i f(\boldsymbol{\theta}) + N\big(0, \sigma_2^2(\boldsymbol{\theta})\big). \end{aligned} \quad (4)$$

Here $\sigma_1(\boldsymbol{\theta})$ and $\sigma_2(\boldsymbol{\theta})$ depend on the real implementation and are not necessarily the same (see Ref. [20] for example). In practical applications, it is common to adaptively adjust the number of samples for varying values of $\boldsymbol{\theta}$ when evaluating f or ∇f. This adjustment ensures that the noise strength, $\sigma_1(\boldsymbol{\theta})$ and $\sigma_2(\boldsymbol{\theta})$, remains comparable. For simplicity, in our later analysis, we assume that $\sigma_2(\boldsymbol{\theta})$ has a uniform upper bound σ_∞ (see Assumption 2).

B. Optimization methods

One widely used approach for optimizing VQA is through the application of gradient descent (GD) [20,21]. The classical gradient descent method involves iteratively updating the parameters $\boldsymbol{\theta}$ by utilizing the gradient of the cost function,

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - a_n \nabla f(\boldsymbol{\theta}_n), \quad (5)$$

where a_n denotes the learning rate. In light of the measurement process in quantum computing, we consider the noisy gradient descent: Rather than implementing Eq. (5) with exact $\nabla f(\boldsymbol{\theta}_n)$, we apply an unbiased estimator $g(\boldsymbol{\theta})$[1] [for example, (4)]. Consequently, the parameter update involves the following iteration:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - a_n g(\boldsymbol{\theta}_n). \quad (6)$$

Since $g(\boldsymbol{\theta}_n)$ is an unbiased estimation, Eq. (6) is equivalent to Eq. (5) in the expectation sense. Specifically, by taking the conditional expectation on both sides, we have

$$\mathbb{E}(\boldsymbol{\theta}_{n+1}|\boldsymbol{\theta}_n) = \boldsymbol{\theta}_n - a_n \nabla f(\boldsymbol{\theta}_n), \quad (7)$$

where $\mathbb{E}(\cdot|\boldsymbol{\theta}_n)$ denotes the conditional expectation given $\boldsymbol{\theta}_n$.

While noisy gradient descent avoids the need for precise gradient information, it still requires the approximated full gradient information at each iteration. As argued before, in the context of VQA, it is often necessary to compute d partial derivatives separately for each direction, which makes the cost of each updating step at least d. In this paper, we introduce an alternative optimization method called random coordinate descent (RCD) [22–24] for addressing Problem 1, with the goal of eliminating the cost dependency on d in each step. RCD can be viewed as a variant of GD where the full gradient in GD is approximated by a randomly selected component of $\nabla f(\boldsymbol{\theta}_n)$ in each iteration. Specifically, one RCD iteration can be expressed as:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - a_n \boldsymbol{e}_{i_n} \partial_{i_n} f(\boldsymbol{\theta}_n). \quad (8)$$

Here $\boldsymbol{e}_{i_n}$ is the i_n-th unit direction, $f'_{i_n}(\boldsymbol{\theta}_n)$ is the corresponding partial derivative of the cost function, and i_n is a random index

[1] $g(\boldsymbol{\theta})$ satisfies $\mathbb{E}[g(\boldsymbol{\theta})] = \nabla f(\boldsymbol{\theta})$.

TABLE I. Comparison of the gradient descent and the randomized coordinate descent methods with an unbiased noisy gradient estimation. d is the dimension of the parameter, and the smoothness constants L and L_{avg} are defined in (11) and (13), respectively. σ_∞^2 is a bound for the measurement noise defined in (15a). In the table, we limit our attention to the situation where the learning rate is fixed.

Algorithm	Iteration cost	Iterations to reach ϵ tolerance	Total cost
GD	$\Omega(d)$	$\tilde{\mathcal{O}}\left(\max\left\{\frac{L\sigma_\infty^2 d}{\mu^2\epsilon}, \frac{L}{\mu}\ln\left(\frac{1}{\epsilon}\right)\right\}\right)$	$\Omega\left(\frac{L\sigma_\infty^2 d^2}{\epsilon}\right)$
RCD	$\Omega(1)$	$\tilde{\mathcal{O}}\left(\max\left\{\frac{L_{\text{avg}}\sigma_\infty^2 d^2}{\mu^2\epsilon}, \frac{dL_{\max}}{\mu}\ln\left(\frac{1}{\epsilon}\right)\right\}\right)$	$\Omega\left(\frac{L_{\text{avg}}\sigma_\infty^2 d^2}{\epsilon}\right)$

uniformly drawn from $\{1, 2, \ldots, d\}$. Similarly to Eq. (6), we can write the noisy RCD as:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - a_n \boldsymbol{e}_{i_n} g_{i_n}(\boldsymbol{\theta}_n). \tag{9}$$

It is important to emphasize that in each iteration of RCD (9), only one partial derivative information is needed. Consequently, within the scope of VQA (as stated in the first assumption of Problem 1), the cost per step of RCD is d times smaller than that of GD.

C. Contribution

This paper primarily focuses on the investigation of RCD in the context of noisy gradient evaluation. Our analysis is conducted in a specific comparison with GD, and we illustrate that, under specific conditions, RCD can serve as a favorable alternative for optimizing parameterized quantum circuits. The main contributions of this study can be summarized as follows:

(i) We show that RCD is theoretically no worse than GD when measuring the complexity by the number of partial derivative calculations (Theorems 3 and 4), assuming the presence of noise and the **local** Polyak-Łojasiewicz (PL) condition, which is more appropriate for PQCs (see Remark 2). A summary of the complexities of the two methods is presented in Table I for comparison. It is important to highlight that the inequality $L_{\text{avg}} \leqslant L \leqslant dL_{\text{avg}}$ always holds. Consequently, when the optimization problem is highly anisotropic, i.e., $L \gg L_{\text{avg}}$, RCD is more cost-effective than GD. In the most extreme case when L is nearly equal to dL_{avg}, RCD can reduce the complexity by a factor of d compared to GD.

(ii) We demonstrate that (noisy) GD and RCD converge with high probability under the **local** PL condition (Assumption 3) and are stable under noisy gradient information. Specifically, if the initial parameter $\boldsymbol{\theta}_0$ resides within the basin $\mathcal{N}(\mathcal{X})$ surrounding the global minimum, then both noisy methods ensure that the subsequent parameters $\boldsymbol{\theta}_n$ will remain consistently within this basin until they converge with the same high probability (Lemmas 5 and 6). To the best of our knowledge, such stochastic stability has not been established for the optimization methods in variational quantum algorithms. We employ a, to the best of our knowledge, novel supermartingale approach and utilize Markov's inequality to quantify the average behavior of the iterates in the optimization method. We anticipate that this, as far as we know, new analytical framework would also be useful for other optimization algorithms for PQCs.

(iii) We provide extensive empirical evidence demonstrating that RCD consistently delivers superior performance compared to GD (Secs. I E and IV). Our numerical findings support the theoretical observation that RCD can take a larger learning rate than GD, leading to faster convergence.

D. Related works

1. Gradient descent with noise

The noisy gradient descent (6) is a popular optimization method in the classical machine learning community. Notable examples are the stochastic gradient descent (SGD) [25] or the perturbed gradient descent [26]. The convergence properties of the noisy gradient descent method in (6) have been extensively studied [20,27–31]. These previous works established that when the cost function is L-smooth, and μ strong convex (or PL [32]) and satisfies other mild conditions, $f(\boldsymbol{\theta}_n)$ converges linearly to an approximation of $f_{\min}$. In the recent study by Sweke *et al.* [20], a comparable theoretical result was demonstrated for the application of the noisy GD method to quantum optimization problems. The authors establish the L-smooth property of the loss function (2) and achieve the convergence of noisy gradient descent under the assumption of unbiased gradient estimation and global PL condition. Another approach to interpreting noise involves assuming that the variational quantum state is influenced by a noisy quantum map [33], ultimately resulting in a noisy loss function. In Ref. [33], the authors investigate the convergence properties of noisy GD, considering the impact of both noise and the stochastic nature of quantum measurement outcomes. They specifically illustrate that, when assuming the global PL condition, noisy GD has the capability to converge to a region where the perturbation in loss is bounded by a quantity associated with the quantum Fisher information of the variational state. With an important observation that PQCs typically do not obey the global PL condition (see Remark 2), we formulate a local PL condition, a property that is more appropriate for PQCs, and we establish the convergence of noisy GD and RCD under this weaker condition and unbiased gradient estimation. We substantiate the efficacy of both optimization algorithms with high probability. To the best of our knowledge, this result is novel and has not been previously explored.

2. Randomized coordinate descent

The RCD method has proven its efficiency over GD in many large-scale optimization problems. The convergence properties of RCD have been extensively explored in the fields of machine learning and optimization [22–24,34–36]. For example, it was shown in Ref. [22] that when f is strongly convex, the convergence complexity of RCD can be consistently lower than or equal to that of GD. Here complexity refers to the total number of partial derivative calculations required for convergence. Later, for strongly convex functions, RCD accelerations were achieved with adaptive momentum-based strategies in various regimes [35,36]. For the nonconvex optimization, recent work [34] shows the global convergence behavior of RCD with a focus on saddle-point avoidance. Nevertheless, convergence rates of RCD have been scarcely studied for nonconvex optimization problems. More importantly, most related works focused on the case where partial derivatives are computed exactly, while in this work, we deal with the case where partial derivatives are estimated, which is subject to noise, and we will refer to it as *noisy* RCD (9).

3. Locally defined convex conditions for convergence analysis

One limitation of the conventional convergence analysis is its reliance on assumptions of global convex [25] or global PL [20] conditions for the cost function $f(\boldsymbol{\theta})$. However, we show that such global assumptions are not satisfied in quantum problem applications with PQCs, as elaborated on in Remark 2. Thus, one must weaken such a global assumption to a local one in the analysis. Convergence analysis under local assumptions requires more sophisticated techniques (see Refs. [37–40] and therein), but it provides important insights that help to interpret empirical results. In our work, we make a local nonconvex condition based on the local PL condition [27]. Under this condition and suitable assumptions for the cost function, we present local convergence analysis for the optimization of PQCs.

By employing a stochastic stability argument, we demonstrate that the noisy GD and RCD methods maintain a comparable convergence rate under our local PL condition with high probability (refer to Theorem 3 and Theorem 4). To the best of the authors' knowledge, this paper is the first to provide a rigorous result for the complexity of noisy GD and RCD under a local PL condition designed for variational quantum algorithms built from PQCs. The recent work [41] also provided a local convergence analysis for the noisy GD but their local assumption and result are more suitable for deep neural networks in the classical machine learning.

4. Other quantum optimization methods

Simultaneous perturbation stochastic approximation (SPSA) [42,43], is a zeroth-order method (i.e., it only involves function values) that has some similarity to the RCD method. Each iteration of SPSA employs a finite-difference formula to estimate one directional derivative of the loss function. SPSA has been used in Ref. [44] to update the control parameters in VQE and achieves a level of accuracy comparable to standard gradient descent methods. We highlight that when each directional or partial derivative is approximated using only two loss function values through finite differences, the cost per step of SPSA and RCD should be comparable. However, in scenarios where analytic partial derivatives are available, such as those where the parameter shift rule [20] is applicable, RCD becomes significantly more robust than SPSA, since the small parameter in finite-difference formulas tends to amplify the noise inherence in the measurement. In contrast, derivative estimations from analytic derivatives are more stable in the presence of measurement and quantum noise, as substantiated by the numerical example provided in Appendix J[2].

Besides SPSA, there are other promising zero-order methods in variational quantum optimization, more commonly known as gradient-free methods. Notably, policy gradient-based techniques have shown their effectiveness in noise robust optimization in the NISQ [45]. Sung *et al.* [46] construct models based on the previous method and further improve the sample efficiency of the methods. Furthermore, these zero-order optimization methods leverage the strengths of reinforcement learning [47–50], Monte Carlo tree search [51–53], and natural evolutionary strategies [54–56], Bayesian [57,58], as well as Gaussian processes [59].

In addition to these zero-order methods, several other optimization methods have been proposed recently [60–64]. One interesting example is the quantum natural gradient (QNG) [64], an approximate second-order method, that incorporates the quantum geometric tensor, which is similar to the natural gradient in classical machine learning. While an outcome of measurement is used as an estimate of the gradient in the QNG or the noisy gradient (6) from (1), the Jordan algorithm [62] encodes the partial derivatives as binary numbers in the computational basis. This algorithm was later improved by Gilyen *et al.* [61] using high-order finite-difference approximations, and applications to VQAs for a certain class of smooth functions were considered. However, the methods [61,62] require a significant number of ancilla qubits and complex control logics, due to the binary encoding of partial derivatives. Alternatively, Ref. [65] proposed a quantum backpropagation algorithm, which uses $\ln d$ copies of the quantum state to compute d derivatives. The overhead for computing d derivatives is $\mathrm{polylog}(d)$ times that of function evaluation (therefore mimicking backpropagation). One of the main drawbacks of their algorithm is that there is an exponential classical cost associated with the process. For a more restrictive class of cost functions (polynomial functions), Ref. [63] proposed a framework to implement the gradient descent and Newton's methods. This method also requires the coherent implementation of the cost function on a quantum computer using, e.g., sparse input oracle, and thus can be challenging to implement in near-term devices.

E. A numerical illustration: Variational quantum eigenvalue solver

As a brief illustration of the performance of noisy GD versus RCD methods, we consider the transverse-field Ising model,

$$H = J\sum_{j=1}^{N-1} Z_j Z_{j+1} + \Delta \sum_{j=1}^{N} X_j, \tag{10}$$

with the coefficient $J = 1$ and $\Delta = 1.5$. Here N denotes the number of qubits, and X_j, Z_j are Pauli operators acting on the j-th qubit. In Fig. 1, we set $N = 10$. To implement the quantum circuits, we use Qiskit Aer-simulator [66] with the command "result.get_counts" that outputs measurement outcomes as classical bitstrings. We utilize the resulting classical bitstrings to compute partial derivatives by applying the parameter shift rule [20]. Thus, the result in Fig. 1 takes into account the measurement noise.

In each experiment, 10 independent simulations are used with a fixed initialization. The parameterized quantum circuit

[2] One can also use the parameter shift rule to calculate the directional derivative in SPSA. Nevertheless, the computation of a random directional derivative always requires evaluating d partial derivatives, making it d times more expensive than the cost of RCD per step.

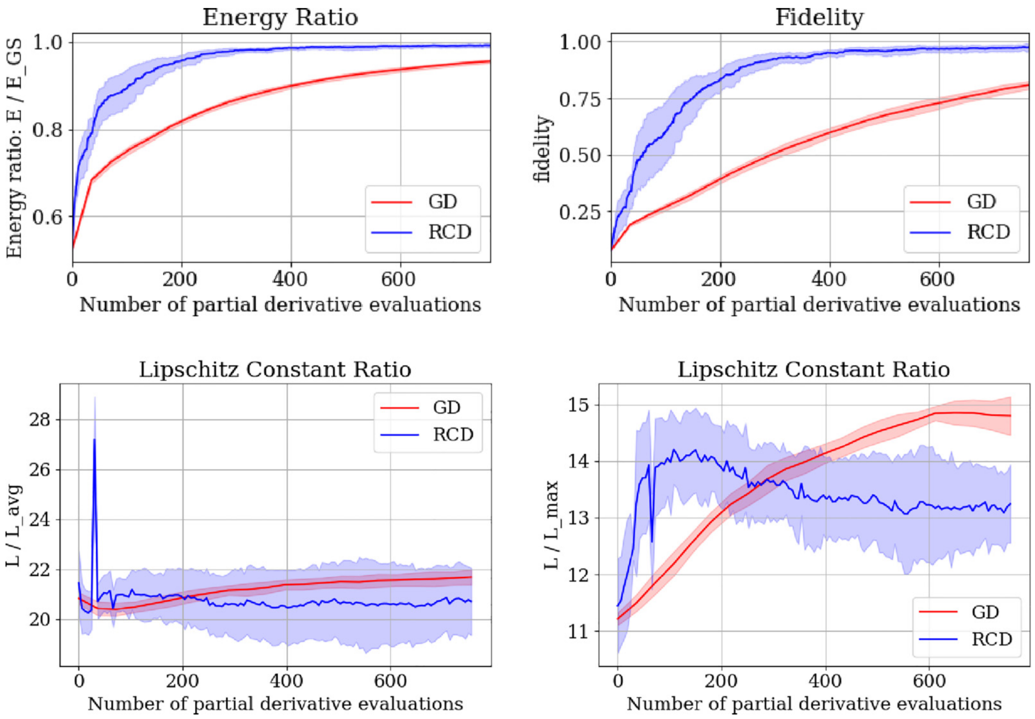


FIG. 1. The comparison of the performance of GD (red) and RCD (blue) for optimizing the Hamiltonian (10). The unit of the x axis labels the number of partial derivative evaluations as an indication of the computational complexity. The top panels show the approximation of the ground state, including the energy ratio (left) and fidelity (right). In the bottom panels, we show the ratios of Lipschitz constants obtained from the two methods are compared: $\frac{L}{L_{\mathrm{avg}}}$ (left) and $\frac{L}{L_{\max}}$ (right).

used for estimating the ground-state energy of the Hamiltonian (10) is given in Fig. 2 (Appendix D).

We compare the optimization performance of the two methods in terms of the number of partial derivative evaluations. The optimization results in Fig. 1 suggest that RCD requires nearly 4 times fewer partial derivative evaluations than GD to converge to an energy ratio of 0.99 and a fidelity of 0.97, both of which are higher than the energy ratio and the fidelity obtained from GD. This observation can be explained by the analysis in Sec. II B, i.e., RCD can be more efficient than GD when the ratio of Lipschitz constants (L/L_{avg} or $L/L_{\max}$) is significantly larger than 1. Specifically, the ratio of the total computational cost of GD to RCD can be linked to the Lipschitz ratios, as summarized in Table I. For instance, in the lower panels of Fig. 1, we observe that the ratio L/L_{avg} and $L/L_{\max}$ remains above 20 and 11 throughout the iterations on average. The faster convergence of RCD can be attributed to these large Lipschitz ratios.

II. PRELIMINARIES AND MAIN RESULTS

Before we establish results pertinent to the performance of RCD, we first establish consistent notations and assumptions, which are presented in Sec. II A. Following that, we outline our key theoretical findings in Sec. II B.

A. Notations and assumptions

Given a vector $\boldsymbol{v} \in \mathbb{R}^d$, we use standard norms for $\boldsymbol{v}$, including the 2-norm $\|\boldsymbol{v}\|_2 := \sqrt{\sum_i v_i^2}$ and the ∞-norm $\|\boldsymbol{v}\|_\infty := \max_i |v_i|$. In order to ensure the convergence of gradient-based methods, we list several technical assumptions.

We assume the cost function f satisfies the L-smoothness. Specifically, it satisfies the following assumption:

Assumption 1. The cost function f is L-smooth, in that

$$\|\nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta}')\|_2 \leqslant L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2, \quad \text{for all } \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d. \tag{11}$$

It is worth noting that assuming L-smoothness is standard when analyzing the convergence of gradient-based algorithms. Moreover, in the setting of PQC, verifying the L-smoothness of the loss function is straightforward. An example of this can be seen in Ref. [[20], Theorem 2], where the author demonstrates that a broad class of PQCs can generate loss functions that satisfy L-smoothness properties.

Since the gradient is Lipschitz continuous, the partial derivatives are Lipschitz continuous as well. We define the componentwise Lipschitz constants.

Definition 1. We say that a function f is L_i-smooth with respect to the i-th component if

$$|\partial_i f(\boldsymbol{\theta} + \boldsymbol{e}_i h) - \partial_i f(\boldsymbol{\theta})| \leqslant L_i |h| \, \forall h \in \mathbb{R}, \tag{12}$$

where $\partial_i f(\boldsymbol{\theta})$ denotes the partial derivative in the i-th direction.

From these componentwise Lipschitz constants, we denote the maximum and average of those constants as

$$L_{\max} := \max_{1 \leqslant i \leqslant d} L_i, \quad L_{\mathrm{avg}} = \frac{1}{d}\sum_{i=1}^{d} L_i. \tag{13}$$

As shown in Ref. [24], in general we have

$$L_i \leqslant L_{\text{avg}} \leqslant L_{\max} \leqslant L \leqslant dL_{\max}. \tag{14}$$

Another interpretation is through the Hessian: When f is twice continuously differentiable, the condition (11) is equivalent to $\nabla^2 f(x) \preceq LI_d$, and, similiarly, condition (12) is equivalent to $\sup_{\boldsymbol{\theta}} |\partial_i^2 f(\boldsymbol{\theta})| \leqslant L_i$. We note that both the upper and lower bounds of L in terms of $L_{\max}$ in (14) are tight. If $\nabla^2 f$ is a diagonal matrix, then $L_{\max} = L$, both being the largest diagonal element of $\nabla^2 f$. (This is the case in which all coordinates are independent of each other, for example, $f = \sum_i \lambda_i x_i^2$.) On the other hand, if $\nabla^2 f = \mathsf{e} \cdot \mathsf{e}^\top$, where $\mathsf{e} \in \mathbb{R}^d$ satisfies $\mathsf{e}_i = 1$ for all i, then $L = dL_{\max}$. This is a situation where f is highly anisotropic, e.g., $f = (\sum_i x_i)^2/2$, where $L = d$ and $L_{\max} = 1$. In addition, when $L_{\text{avg}} = L$, we see that $L_{\text{avg}} = L_{\max} = L_i$ for all i.

Next, it is important to note that the estimation of the gradients in quantum computing can be susceptible to noise, which stems from the inherent nature of quantum measurements. Consequently, in our analysis and comparative studies of different optimization methods, we will take into account the presence of noise. To facilitate such analysis, we make the following assumption:

Assumption 2 (Bounds of the noise with respect to the 2-norm). Given any $\boldsymbol{\theta} \in \mathbb{R}^d$, we assume that we can find an unbiased random estimate $\boldsymbol{g}(\boldsymbol{\theta})$ for the gradient $\nabla f(\boldsymbol{\theta})$, meaning that

$$\mathbb{E}[\boldsymbol{g}(\boldsymbol{\theta})] = \nabla f(\boldsymbol{\theta}).$$

Furthermore, we assume that there exists a constant $\sigma_\infty^2 > 0$ such that

$$\sigma_\infty^2 > \sup_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{1 \leqslant i \leqslant d} \mathbb{E}[|\partial_i f(\boldsymbol{\theta}) - g_i(\boldsymbol{\theta})|^2]. \tag{15a}$$

Here we also assume $g(\boldsymbol{\theta})$ is independent for different $\boldsymbol{\theta}$.

The constraint on the noise bounds is a standard assumption in the analysis of noisy gradient-based algorithms. A straightforward case that satisfies this assumption is $\nabla f(\boldsymbol{\theta}) \in L^\infty(\mathbb{R}^d)$, a condition frequently observed in optimizations involving PQCs. Moreover, in cases where $|\nabla f(\boldsymbol{\theta})|$ lacks an upper bound, we can adjust the sample sizes in various $\boldsymbol{\theta}$ values to effectively reduce the variance to a bounded range.

Additionally, we assume the existence of a basin encompassing the global minimum, within which f satisfies the PL condition [32] and, equivalently, the local PŁ condition [27].

Assumption 3 (Local PL condition). Define $\mathcal{X}$ as the set of global minima and $f_{\min}$ as the global minimum value evaluated over $\mathcal{X}$. Then there exists a $\delta_f, \mu > 0$ such that for any $\boldsymbol{\theta} \in \mathcal{N}(\mathcal{X}) := f^{-1}([f_{\min}, \delta_f))$,

$$\|\nabla f(\boldsymbol{\theta})\|^2 \geqslant 2\mu(f(\boldsymbol{\theta}) - f_{\min}).$$

While a similar condition was theoretically justified in classical machine learning (e.g., over-parameterized regimes [27]), analysis on the case of parameterized quantum circuits Eq. (1) is still lacking and challenging. The technical tools in Ref. [27] might be applicable for the case of PQCs. We leave this for future work. Nevertheless, it is worthwhile to highlight that the local PL condition is defined not on the entire space $\mathbb{R}^d$ but the neighborhood of global minima $\mathcal{N}(\mathcal{X})$, which is reasonable in the setting of variational quantum algorithm. We support this argument with the following remark.

Remark 2 (PQCs are neither global PL nor globally convex). Let $f(\boldsymbol{\theta})$ be a cost function defined by some parameterized quantum circuit (2). Note that f is periodic and smooth, due to its specialized form. By the extreme value theorem, we see that there exist global maximum and minimum of f, denoted by $\boldsymbol{\theta}_{\max}$ and $\boldsymbol{\theta}_{\min}$. In general, f is not constant, which means that $f_{\max} > f_{\min}$. Had f satisfied the global PL condition, it would have followed that at the global maximum $\boldsymbol{\theta}_{\max}$,

$$0 = \|\nabla f(\boldsymbol{\theta}_{\max})\|^2 \geqslant 2\mu(f_{\max} - f_{\min}) \geqslant 0, \tag{16}$$

which gives a contradiction to the general case that $f_{\max} > f_{\min}$. As another case study, if f is assumed to be convex, namely

$$f(\boldsymbol{\theta}') \geqslant f(\boldsymbol{\theta}) + (\nabla f(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d, \tag{17}$$

then setting $\boldsymbol{\theta} = \boldsymbol{\theta}_{\max}$ and $\boldsymbol{\theta}' = \boldsymbol{\theta}_{\min}$ results in a contradiction. Therefore, the cost function f that is constructed from an ansatz similar to (2), will not satisfy global PL or convex conditions in general.

B. Main result: Complexity comparison of GD and RCD

In this study, our main focus is to compare the complexity of noisy GD and RCD under the assumptions of a local PL condition 3. For the sake of simplicity, in the remaining part of this paper, we will refer to "noisy gradient descent" and "noisy randomized coordinate descent" as "GD" and "RCD," respectively, without explicitly mentioning the term "noisy."

The main theoretical results are summarized in the following two theorems:

Theorem 3 (Complexity of GD under the local PL condition). Assume f is a L-smooth function that satisfies assumption 3 and g satisfies assumption 2. Given $\epsilon > 0$ small enough, if $f(\boldsymbol{\theta}_1) \leqslant \delta_f$ and $a_n = \Theta(\min\{\mu\epsilon/(L\sigma_\infty^2 d), 1/L\})$ in GD (7), then with probability $1 - f(\boldsymbol{\theta}_1)/\delta_f - o(1)$, there exists at least one

$$n < N = \tilde{\Theta}\big(\max\big\{L\sigma_\infty^2 d/(\mu^2\epsilon), L/\mu\big\}\big) \tag{18}$$

such that $f(\boldsymbol{\theta}_n) \leqslant f_{\min} + \epsilon$.

Theorem 4 (Complexity of RCD under the local PL condition). Assume f is a L-smooth function that satisfies assumption 3 and g satisfies assumption 2. Given $\epsilon > 0$ small enough, if $f(\boldsymbol{\theta}_1) \leqslant \delta_f$ and $a_n = \Theta(\max\{\mu\epsilon/(L_{\text{avg}}\sigma_\infty^2 d), 1/L_{\max}\})$ in RCD (9), then with probability $1 - f(\boldsymbol{\theta}_1)/\delta_f - o(1)$, there exists at least one

$$n < N = \tilde{\Theta}\big(\max\big\{L_{\text{avg}}\sigma_\infty^2 d^2/(\mu^2\epsilon), L_{\max}d/\mu\big\}\big) \tag{19}$$

such that $f(\boldsymbol{\theta}_n) \leqslant f_{\min} + \epsilon$.

Based on the theorem mentioned above, to achieve $f(\boldsymbol{\theta}_n) - f_{\min} \leqslant \epsilon$, we can select the learning rate $a_n = \frac{\mu\epsilon}{L\sigma_\infty^2 d}$ for GD and $a_n = \frac{\mu\epsilon}{L_{\text{avg}}\sigma_\infty^2 d}$ for RCD. Recalling Eq. (14), we observe that $L_{\text{avg}} \leqslant L$, which means that we could use a larger learning rate for RCD. This choice aligns with the learning rates utilized in the numerical experiments presented in Sec. I E as well as those in Sec. IV.

We compare the complexity of the noisy GD and RCD methods with the estimates of the number of iterations. First, according to the above result, we conclude that the number of iterations required for GD is $N = \tilde{\Theta}(\frac{L\sigma_\infty^2 d}{\mu^2\epsilon})$[3], while for RCD, we have $N = \tilde{\mathcal{O}}(\frac{L_{\text{avg}}\sigma_\infty^2 d^2}{\mu^2\epsilon})$. Notably, in RCD, there is an additional factor of d, which can be understood in the expectation sense: During each iteration of the noisy RCD, the randomness arises from two sources: the random direction i_n and the noisy partial derivative $g_{i_n}(\boldsymbol{\theta}_n)$. By taking the conditional expectation with respect to $\boldsymbol{\theta}_n$, we obtain:

$$\mathbb{E}(\boldsymbol{\theta}_{n+1}|\boldsymbol{\theta}_n) = \boldsymbol{\theta}_n - \frac{a_n}{d}\nabla f(\boldsymbol{\theta}_n). \tag{20}$$

Compared with (7), there is an extra $1/d$ factor in the expectation of RCD. Consequently, in each iteration, the rate of decay of the cost function is smaller in RCD compared to GD. Consequently, we anticipate that RCD would necessitate more iteration steps to achieve convergence. On the other hand, it is also important to note that in certain scenarios where $L_{\text{avg}}d$ is comparable to L, the number of iterations required for RCD is comparable to that of GD.

Meanwhile, it is important to point out that a more practical criterion for comparing the two methods is the cumulative cost of each method, which is represented by the number of partial derivative calculations from the quantum circuits. This is because quantum algorithms for estimating the gradient have a cost proportional to d. Since each iteration of GD needs to calculate the full gradient (d partial derivatives), the total number of partial derivative estimations in GD is

$$N_{\text{partial,GD}} = \tilde{\Theta}\left(\frac{L\sigma_\infty^2 d^2}{\mu^2\epsilon}\right).$$

In contrast, the number of partial derivative estimations in RCD is

$$N_{\text{partial,RCD}} = \tilde{\mathcal{O}}\left(\frac{L_{\text{avg}}\sigma_\infty^2 d^2}{\mu^2\epsilon}\right).$$

From Eq. (14), we can deduce that:

$$\tilde{\Omega}(N_{\text{partial,RCD}}) = N_{\text{partial,GD}} = d\tilde{\mathcal{O}}(N_{\text{partial,RCD}}).$$

This suggests that the computational cost of RCD is L/L_{avg} times cheaper than that of GD. In an extreme case where f is highly skewed, i.e., $L/L_{\text{avg}} \approx d$, RCD can reduce the computational cost by a factor of the dimension d, which will be a significant reduction for large quantum circuits.

In addition to the complexity result, it is worth noting that the two methods exhibit similar success probability, which is approximately $1 - f(\boldsymbol{\theta}_1)/\delta_f$, as indicated by the two aforementioned theorems. This observation is quite surprising, as each iteration of RCD appears noisier due to the random selection of the updating direction i_n. Intuitively, this suggests that we might need to choose a smaller learning rate a_n to ensure stability in RCD, which would consequently increase its complexity. However, our theory unveils that choosing a similar learning rate a_n is adequate to stabilize RCD. To elucidate this point, it is important to recognize that, on average, RCD behaves equivalently to GD. By conducting more iterations, RCD can approximate its average behavior (expectation), effectively mitigating the extra randomness introduced by i_n. This compensation mechanism ensures that the success probabilities remain consistent between the two methods.

III. PROOF OF MAIN RESULTS

In this section, we provide the proofs for Theorems 3 and 4. We will start by showing the stochastic stability of the two methods in Sec. III A. This will guarantee that the parameter $\boldsymbol{\theta}$ is likely to stay close to the global minimum until attaining a small loss. Following that, in Sec. III B, we utilize the local PL condition around the global minimum to establish the convergence of $f(\boldsymbol{\theta}_n)$. In all of the following theoretical results and the corresponding proofs in the Appendixes, we assume $f_{\min} = 0$ without loss of generality by modifying the original function as

$$f(\boldsymbol{\theta}) \leftarrow f(\boldsymbol{\theta}) - f_{\min}. \tag{21}$$

Thus, all results in this section can be reformulated for the original cost function by the substitution (21), which will yield Theorems 3 and 4.

A. Stochastic stability

In the context of optimization, stability and convergence are not separate properties. In a deterministic algorithm, convergence immediately guarantees stability. However, this connection does not hold for stochastic processes in general. For instance, when optimization methods such as noisy GD, SGD, or noisy RCD are applied, discrete-time stochastic processes are generated. In such cases, a convergence theory must be developed for a collection of random paths, which can exhibit different convergence behaviors among themselves.

In our specific case, we anticipate that when $\boldsymbol{\theta}_n$ remains within the basin $\mathcal{N}(\mathcal{X})$ and the learning rate is correctly chosen, both the GD and the RCD methods, when the gradient is exactly calculated, converge to a global minimum due to the local PL condition stated in Assumption 3. However, in the presence of noise in the gradient and the use of a constant learning rate, it is generally impossible to ensure that $\boldsymbol{\theta}_n \in \mathcal{N}(\mathcal{X})$ almost surely, unless a different strategy is adopted such as the decreasing learning rates [37,38,40]. On the other hand, the purpose of the optimization algorithm is to minimize the loss function, which means that it suffices to ensure stability until a small loss is achieved. To quantify such a likelihood, in this section, we demonstrate that when $\boldsymbol{\theta}_0 \in \mathcal{N}(\mathcal{X})$, there exists a finite probability that $\boldsymbol{\theta}_n$ obtained from GD and RCD remain within $\mathcal{N}(\mathcal{X})$ until achieving a small loss. This provides a high probability of convergence for the two methods.

We summarize the result for noisy GD in the following lemma.

Lemma 5. Assume that f is a L-smooth function that satisfies the Assumption 3 and g satisfies the Assumption 2. If $f(\boldsymbol{\theta}_1) \leqslant \delta_f$ and the learning rate is chosen as follows:

$$a_n = a < \min\left\{\frac{1}{L}, \frac{2\mu\delta_f}{L\sigma_\infty^2 d}\right\},$$

[3] This complexity aligns with the classical theoretical results for GD, which typically assume the presence of strong convexity or a local PL condition for the function f.

then, with high probability, iterations of noisy GD (7) remain in $f^{-1}([0, \delta_f))$ until a small loss is achieved. Specifically,

$$\mathbb{P}\Bigg\{\exists N > 0 \text{ such that } f(\boldsymbol{\theta}_N) \notin \mathcal{N} \text{ and } f(\boldsymbol{\theta}_n) > \frac{La\sigma_\infty^2 d}{\mu},\ \forall n < N\Bigg\} \leqslant \frac{f(\boldsymbol{\theta}_1)}{\delta_f}. \tag{22}$$

In light of Eq. (22), if we select the learning rate a_n to be sufficiently small, then with a probability of $1 - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}$, the parameters are guaranteed to achieve a small loss before escaping the basin.

Despite infrequent updates of the gradient components, RCD still demonstrates a similar level of stochastic stability. This key observation is summarized in the following lemma:

Lemma 6. Assume that f is a L-smooth function that satisfies Assumption 3 and g satisfies Assumption 2. Given any $f(\boldsymbol{\theta}_1) < \delta_f$, if one chooses the learning rate

$$a_n = a < \min\left\{\frac{1}{L_{\max}}, \frac{d}{\mu}, \frac{2\mu\delta_f}{L_{\text{avg}}\sigma_\infty^2 d}\right\},$$

then, with high probability, iterations from the noisy RCD (9) stay at $f^{-1}([0, \delta_f))$ until achieving a small loss. Specifically,

$$\mathbb{P}\Bigg\{\exists N > 0 \text{ such that } f(\boldsymbol{\theta}_N) \notin \mathcal{N} \text{ and } f(\boldsymbol{\theta}_n) > \frac{L_{\text{avg}}a\sigma_\infty^2 d}{\mu},\ \forall n < N\Bigg\} \leqslant \frac{f(\boldsymbol{\theta}_1)}{\delta_f}.$$

The proofs of Lemma 5 and 6 are provided in Appendixes A and B, respectively. The core concept of these proofs is based on the construction of a specialized supermartingale and the utilization of Markov's inequality. For example, to prove Lemma 5, we define a stochastic process

$$V_n = \begin{cases} f(\boldsymbol{\theta}_n)\mathbb{I}_n, & n < \tau \\ f(\boldsymbol{\theta}_\tau)\mathbb{I}_\tau, & n \geqslant \tau \end{cases}.$$

where the indicator random variable is given by

$$I_n = \begin{cases} 1, & \text{if } \{\boldsymbol{\theta}_k\}_{k=1}^{n-1} \subset f^{-1}([0, \delta_f)) \\ 0, & \text{otherwise.} \end{cases},$$

and the stopping time

$$\tau = \inf\left\{k : f(\boldsymbol{\theta}_k) \leqslant \frac{La\sigma_\infty^2 d}{\mu}\right\}.$$

We observe that V_n is a meticulously crafted supermartingale, allowing us to distinguish between stable and unstable events. In particular, we demonstrate that if $\boldsymbol{\theta}_n$ exits the basin before it reaches $f(\boldsymbol{\theta}_n) = \frac{La\sigma_\infty^2 d}{\mu}$ (an unstable event), then $\sup_n V_n \geqslant \delta_f$. Therefore, we can employ V_n as a categorizer and the probability of failure of GD can be characterized by the value of V_n. More specifically,

$$\mathbb{P}\Bigg\{\exists N > 0 \text{ such that } f(\boldsymbol{\theta}_N) \notin \mathcal{N} \text{ and } f(\boldsymbol{\theta}_n) > \frac{La\sigma_\infty^2 d}{\mu},\ \forall n < N\Bigg\} \leqslant \mathbb{P}\{\sup_n V_n \geqslant \delta_f\}.$$

Except for its use as a categorizer, we have designed V_n in such a way that it is a supermartingale, meaning $\mathbb{E}(V_{n+1}|\boldsymbol{\theta}_{k\leqslant n}) \leqslant V_n$. Therefore, we can use Markov's inequality for supermartingales to bound the supremum of V_n and achieve the desired result.

B. Convergence analysis

In this section, we present the convergence properties of noisy GD and RCD methods. It is important to note that Theorems 3 and 4 directly follow from Theorems 7 and 8, respectively.

Our first theorem shows the convergence performance of the noisy GD method,

Theorem 7. Assume f is a L-smooth function that satisfies Assumption 3 and g satisfies Assumption 2. Given any precision $0 < \epsilon < \delta_f$, the initial guess $f(\boldsymbol{\theta}_1) < \delta_f$, and the probability of failure $\eta \in (\frac{f(\boldsymbol{\theta}_1)}{\delta_f^{-1}}, 1)$, we choose the learning rate in (5)

$$a_n = a = \mathcal{O}\left(\min\left\{\frac{1}{L}, \frac{\mu\epsilon}{L\sigma_\infty^2 d}\right\}\right),$$

and the total number of iterations

$$N = \Omega\left(\frac{1}{\mu a\eta}\ln\left\{\frac{f(\boldsymbol{\theta}_{\mathbf{1}})}{\left[\eta - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}\right]\epsilon}\right\}\right).$$

Then, with probability $1 - \eta$, we can find at least one $\boldsymbol{\theta}_m$ with $1 \leqslant m \leqslant N$ such that $f(\boldsymbol{\theta}_m) \leqslant \epsilon$. In particular,

$$\mathbb{P}\{\exists m \leqslant N,\ f(\boldsymbol{\theta}_m) \leqslant \epsilon\} \geqslant 1 - \eta,$$

Next, we state the convergence property of the noisy RCD method in the following theorem,

Theorem 8. Assume f is a L-smooth function that satisfies Assumption 3 and g satisfies Assumption 2. Given any precision $0 < \epsilon < \delta_f$, the initial guess $f(\boldsymbol{\theta}_1) < \delta_f$, and the probability of failure $\eta \in (\frac{f(\boldsymbol{\theta}_1)}{\delta_f^{-1}}, 1)$, we choose the learning rate in (9)

$$a_n = a = \mathcal{O}\left(\min\left\{\frac{1}{L_{\max}}, \frac{d}{\mu}, \frac{\mu\epsilon}{L_{\text{avg}}\sigma_\infty^2 d}\right\}\right),$$

and the total number of iterations

$$N = \Omega\left(\frac{d}{\mu a\eta}\ln\left\{\frac{f(\boldsymbol{\theta}_{\mathbf{1}})}{\left[\eta - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}\right]\epsilon}\right\}\right).$$

Then, with probability $1 - \eta$, we can find at least one $\boldsymbol{\theta}_m$ with $1 \leqslant m \leqslant N$ such that $f(\boldsymbol{\theta}_m) \leqslant \epsilon$. In particular,

$$\mathbb{P}\{\exists m \leqslant N,\ f(\boldsymbol{\theta}_m) \leqslant \epsilon\} \geqslant 1 - \eta.$$

The proofs of these theorems can be found in the Appendix C.

Remark 9. We emphasize that Theorem 7 and theorem Theorem 8 are general convergence results that require only mild conditions. Specifically, Theorem 7 can be used to demonstrate the stability and convergence of the traditional SGD algorithm when the right assumptions are in place. A convergence result analogous to the one previously discussed has been investigated in Ref. [[27], Theorem 7], where the authors impose a more stringent requirement on the cost function [67]. In our work, we demonstrate the convergence of noisy GD using more sophisticated techniques in probability theory and adopt a weak version of probabilistic convergence [68]. In addition, our approach can be directly extended to show the convergence of noisy RCD as in theorem Theorem 8, which to the best of our knowledge, has not been established before. These two theorems suggest that with a high probability, the loss function can achieve a small loss during the training process. In other words, it is likely that the parameter $\boldsymbol{\theta}$ remains in the basin $\mathcal{N}$ until the precision ϵ is attained at some point. After that, the optimization algorithm could diverge unless a certain strategy is applied, for example, a schedule of decreasing learning rates or an early stopping criterion.

Remark 10. Our theoretical result clarifies a relation between the learning rate and the desired precision in optimization. For example, the precision ϵ is manifested in the upper bounds of the learning rates in Theorem 7 and theorem Theorem 8. Thus, to reach precision ϵ, it is suggested to use an $\mathcal{O}(\epsilon)$ learning rate. Otherwise, due to the stability issue, the trajectory is no longer guaranteed to converge to the precision with positive probability.

We present the roadmap for proving Theorem 7 as follows: Define the stopping time

$$\tau = \inf\{k : f(\boldsymbol{\theta}_k) \leqslant \epsilon\}.$$

To prove Theorem 7, it suffices to demonstrate that the probability of failure $\mathbb{P}(\tau > N)$ is small. Since the learning rate a_n is selected to be sufficiently small and, according to the Lemma 5, it is likely that $\boldsymbol{\theta}_n$ will remain within the basin until the ϵ loss is achieved.[4] Thus, informally, it suffices for us to assume $\boldsymbol{\theta}_n \in \mathcal{N}$. The next step is to find an upper bound for the probability of failure $p_{\text{fail}} = \mathbb{P}(\tau > N)$. Using the local PL condition, we can show that when $\epsilon < f(\boldsymbol{\theta}_n) < \delta_f$,

$$\mathbb{E}(f(\boldsymbol{\theta}_{n+1})|\boldsymbol{\theta}_n) \leqslant \left(1 - \frac{\mu a}{2}\right) f(\boldsymbol{\theta}_n),$$

meaning that the conditional expectation of $f(\boldsymbol{\theta}_{n+1})$ decays to zero with rate $(1 - \frac{\mu a}{2})$. Inspired by this observation, we can construct a supermartingale to show that, if $\tau > N$, then, with high probability, we have $\inf_{1\leqslant n\leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon$. We note that this event is complementary to the failure event $\{\tau > N\}$. Consequently, we obtain an upper bound for p_{fail}.

IV. NUMERICAL RESULTS

In Sec. I E, depicted in Fig. 1, we have demonstrated that the noisy RCD leads to faster convergence than the noisy GD for VQE problems. In this section, we extend our investigation to gauge the efficiency of noisy RCD applied to various other variational quantum algorithms, especially those involving nonconvex optimization problems. The implementation of these algorithms is executed on classical computers. To emulate quantum measurement noise, the partial derivatives undergo perturbation through additive Gaussian noise as outlined in Sec. I A.[5] Subsequently, we substantiate this approximation through a numerical experiment on a quantum simulator. This experiment further also proposes suitable values for the strength of the Gaussian noise that we will introduce in the upcoming numerical tests to appropriately mimic the measurement noise.

In the experiment presented in Sec. IV B, we utilize Qiskit-0.44.1 [66].

The algorithms for subsequent examples are implemented using Numpy [69] and Jax [70]. We conducted each experiment 10 times, employing different random initializations for each run. All tests are executed on an Intel Xeon CPU @ 2.20 GHz, complemented by a T4 GPU.

A. Analyzing the noise distribution

Building on the numerical experiment detailed in Sec. I E and executed in Qiskit, we analyze the statistics of the partial derivatives derived from the quantum circuit. Figure 2 showcases the histograms representing 10 000 estimates of partial derivatives with respect to the initial 12 directions, while the histograms for the remaining directions are presented in Appendix F. Each estimate of the partial derivatives is averaged over 1000 shots. From all histograms, we can clearly see that the distribution is closely approximated by a Gaussian distribution. In addition, the magnitude of the standard deviation of partial derivative estimates across all directions is comparable. These observations support assumptions of the noise model in Problem 1. For simplicity, we will employ the Gaussian noise model in our subsequent investigations to compare the performance of the noisy GD and RCD methods.

In the next two sections, we conduct a comprehensive comparison between noisy RCD and GD across a broad spectrum of variational quantum algorithms and applications.

B. VQE with a varied circuit structure

In Sec. I E, we utilize the VQE for the TFIM (10) employing both the noisy GD and the noisy RCD. In this section, we tackle the same optimization task but with a modified setup. Specifically, Fig. 3 depicts the PQC [44] utilized in the experiments showcased in Fig. 4, different from the one used in Fig. 1.

In the experiments illustrated in Fig. 4, each optimization outcome derives from 10 identical simulations with the same initial condition. We set the learning rates for the RCD and GD at 0.3 and 0.05, respectively. Each experiment utilizes 10,000 shots, with 18 trainable parameters. Results shown in Fig. 4 demonstrate that, compared to GD, RCD requires nearly three times fewer partial derivative evaluations to converge.

[4] Rigorously, we must also take into account the possibility that the optimization algorithm does not reach ϵ loss in a finite number of iterations.

[5] The derivative with noise is computed by adding Gaussian noise to the original derivative: $\partial_i f(x) \leftarrow \partial_i f(x) + \epsilon$, where ϵ follows a Gaussian distribution, denoted as $\mathcal{N}(0, \sigma)$. In this notation, σ signifies the standard deviation, defining the intensity of the Gaussian noise.

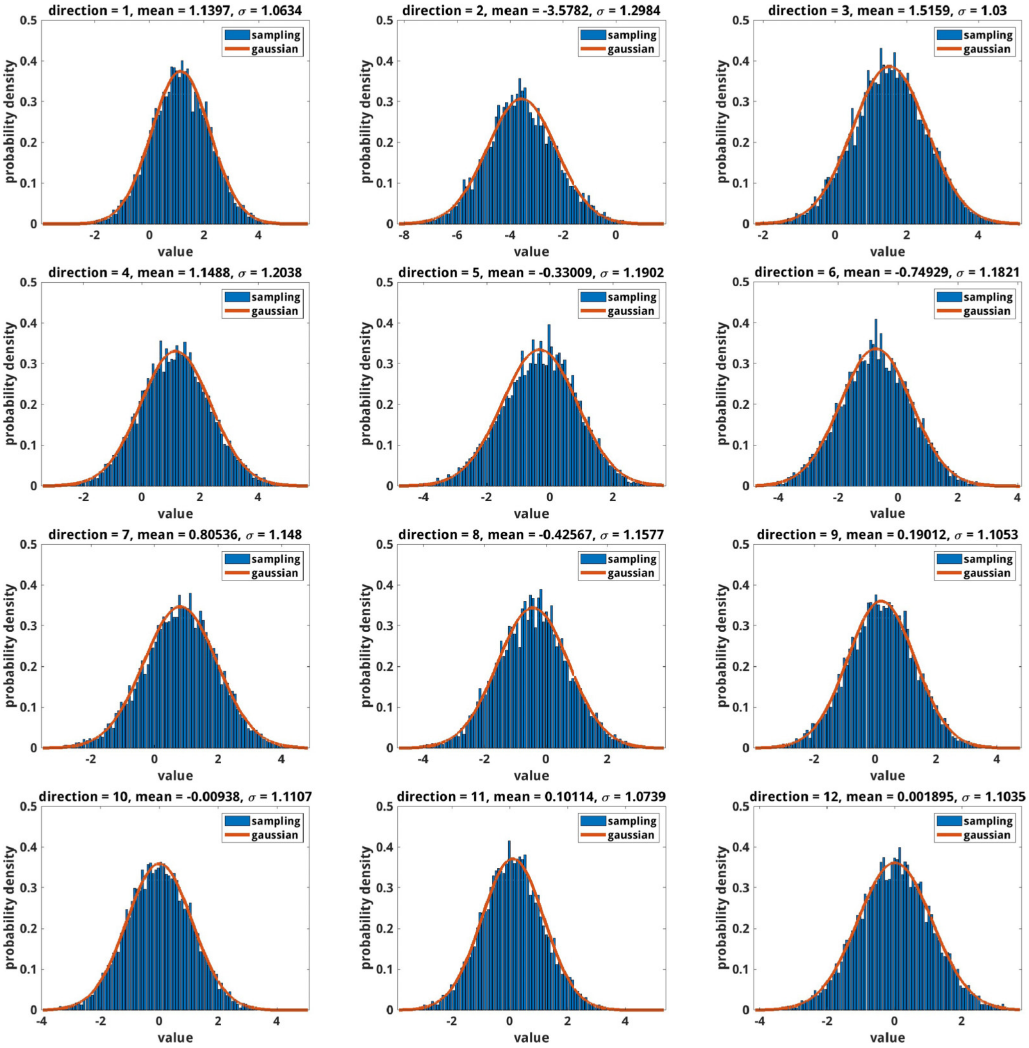


FIG. 2. Histograms of the estimated partial derivatives: Each panel displays the histogram of 10 000 partial derivative estimates in one of the first 12 directions using the parameter-shift rule. The sampling of the partial derivatives is carried out at a suboptimal point chosen from one simulation used in Fig. 1, where the fidelity is about 0.883.

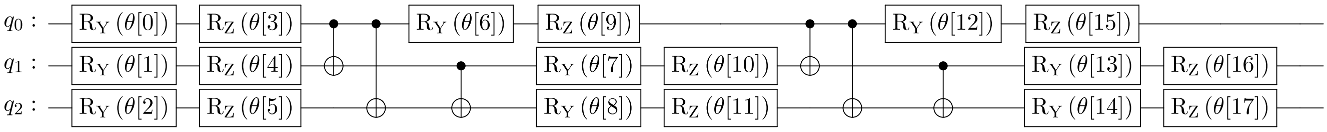

q_0 :
q_1 :
q_2 :
$\mathrm{R_Y}\,(\theta[0])$
$\mathrm{R_Y}\,(\theta[1])$
$\mathrm{R_Y}\,(\theta[2])$
$\mathrm{R_Z}\,(\theta[3])$
$\mathrm{R_Z}\,(\theta[4])$
$\mathrm{R_Z}\,(\theta[5])$
$\mathrm{R_Y}\,(\theta[6])$
$\mathrm{R_Y}\,(\theta[7])$
$\mathrm{R_Y}\,(\theta[8])$
$\mathrm{R_Z}\,(\theta[9])$
$\mathrm{R_Z}\,(\theta[10])$
$\mathrm{R_Z}\,(\theta[11])$
$\mathrm{R_Y}\,(\theta[12])$
$\mathrm{R_Y}\,(\theta[13])$
$\mathrm{R_Y}\,(\theta[14])$
$\mathrm{R_Z}\,(\theta[15])$
$\mathrm{R_Z}\,(\theta[16])$
$\mathrm{R_Z}\,(\theta[17])$

FIG. 3. A variational circuit ansatz is employed for the Transverse-field Ising model expressed in Eq. (10), utilizing three qubits. This circuit is a parameterized construct comprised of alternating rotation and entanglement layers. Each rotation layer involves the application of single qubit gates, specifically Rotation-y and Rotation-z gates, to all qubits. In contrast, the entanglement layer employs two-qubit gates, namely the controlled-X gate, to facilitate entanglement among the qubits. The ansatz is designated with 18 parameters.

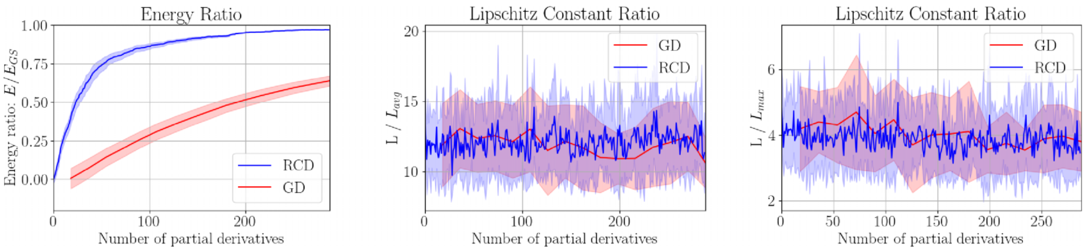


FIG. 4. Performance comparison between GD (red) and RCD (blue) in terms of energy ratio and Lipschitz constant ratios for optimizing the Hamiltonian (10). The energy ratio E/E_{GS} is presented in the left panel, while the Lipschitz constant ratios, denoted as $\frac{L}{L_{\mathrm{avg}}}$ and $\frac{L}{L_{\mathrm{max}}}$, are shown in the middle and right panels, respectively. The shaded areas in each panel represent variations observed across multiple trials.

C. Quantum approximate optimization algorithm for quantum Hamiltonians

The quantum approximate optimization algorithm (QAOA) [5], originally devised for solving combinatorial problems, is a leading example for demonstrating quantum advantage on near-term quantum computers. As introduced in Ref. [5], the QAOA utilizes a PQC, which naturally enables optimization through the variational quantum algorithm.

In a generalized QAOA model, we begin with an initial quantum state $|\psi_i\rangle$, which can be easily prepared in experiments, and let it evolve by a parameterized unitary transformation,

$$\begin{aligned}|\psi(\boldsymbol{\alpha},\boldsymbol{\beta})\rangle &= U\left(\{\alpha_j,\beta_j\}_{j=1}^{p}\right)|\psi_i\rangle\\ &= e^{-iH_2\beta_p}e^{-iH_1\alpha_p}\cdots e^{-iH_2\beta_1}e^{-iH_1\alpha_1}|\psi_i\rangle,\end{aligned} \tag{23}$$

where the vector $\boldsymbol{\alpha}$ (or $\boldsymbol{\beta}$) enumerates the parameters α_j (or β_j), and thus the total number of parameters is $2p$ and the unitary transformation alternates between two kinds of parameterized unitary transformations. With this ansatz, the optimization is performed with the parameters $\{\alpha_j, \beta_j\}$ associated with the application-dependent Hamiltonian matrices H_1 and H_2, respectively.

In the subsequent sections, we will consider optimization problems based on the QAOA (23). We will conduct a comparative analysis of the noisy GD and RCD for various QAOA models that will span a range of systems, including the Ising model (refer to Sec. IV C 1), the Heisenberg model (refer to Sec. IV C 2), and variational auantum factoring (refer to Sec. IV D 3).

1. *QAOA: Ising model*

In this section, we parametrize the transverse-field Ising model by a Hamiltonian

$$H[h] = \sum_{j=1}^{N-1} Z_{j+1}Z_j + \sum_{j=1}^{N}(Z_j + hX_j), \tag{24}$$

where N denotes the total number of qubits. The global control field $h \in \{\pm 4\}$ takes two discrete values, corresponding to the two alternating QAOA generators $H_1 = H[-4]$ and $H_2 = H[+4]$ [45,48]. The initial state $|\psi_i\rangle$ corresponds to the ground state of $H[-2]$, while the desired target state $|\psi_*\rangle$ is selected as the ground state of $H[+2]$. The variational problem aims to optimize the fidelity,[6]

$$\max_{\{\alpha_i,\beta_i\}_{i=1}^{p}} \mathcal{F}\left(\{\alpha_i,\beta_i\}_{i=1}^{p}\right) = \max_{\{\alpha_i,\beta_i\}_{i=1}^{p}} \left|\langle\psi_*|U(\{\alpha_i,\beta_i\}_{i=1}^{p})|\psi_i\rangle\right|^2, \tag{25}$$

where

$$U\left(\{\alpha_i,\beta_i\}_{i=1}^{p}\right)|\psi_i\rangle = e^{-iH_2\beta_p}e^{-iH_1\alpha_p}\cdots e^{-iH_2\beta_1}e^{-iH_1\alpha_1}|\psi_i\rangle. \tag{26}$$

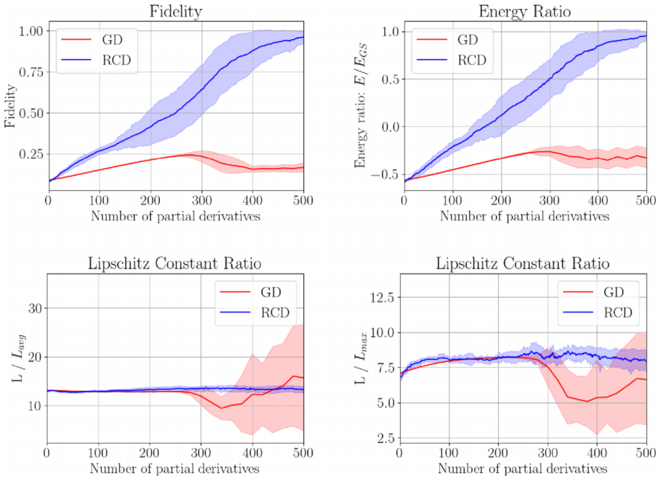


FIG. 5. Performance comparison between the noisy GD and RCD for the Ising model (24). The corresponding Lipschitz constant ratios, denoted as $\frac{L}{L_{\text{avg}}}$ and $\frac{L}{L_{\text{max}}}$, are presented in the bottom figures. The shaded areas within the figures represent variations that have been observed across 10 random realizations. The optimization is performed for parameters with dimension equal to 20.

We note that the fidelity optimization (25) is equivalent to the optimization of the form (2) by letting the Hamiltonian be $|\psi_*\rangle\langle\psi_*|$.

In the numerical test, we choose a system from (24) with three qubits ($N = 3$) and then apply both GD and RCD methods in the optimization. Figure 5 shows the optimization results obtained from the noisy GD and RCD with the respective learning rates of 0.0045 and 0.015 by using an ansatz defined with 20 parameters. By adjusting the learning rate and tracking the stability. We observe that RCD permits a larger learning rate in comparison to GD, while maintaining the stability. Similarly to the results presented in Fig. 1, we compare the performance of the two methods in terms of the number of partial derivative evaluations. From Fig. 5, We observe that noisy RCD converges much faster than noisy GD. While RCD achieves a fidelity near 1 with 500 partial derivative evaluations, GD only attains a fidelity below 0.25 with an equivalent number of evaluations. This computational effectiveness of RCD can be attributed to the large ratios of Lipschitz constants shown in Fig. 5, which are obtained along the optimization trajectories.

2. *QAOA: Heisenberg model*

Our second test problem with QAOA is the (anisotropic) spin-1 Heisenberg model, $H = H_1 + H_2$, with the alternating Hamiltonians given by

$$H_1 = J\sum_{j=1}^{N}(X_{j+1}X_j + Y_{j+1}Y_j), \quad H_2 = \Delta\sum_{j=1}^{N} Z_{j+1}Z_j,$$

with anisotropic parameter $\Delta/J = 0.5$ (topological-Haldane [71–74]). For the Heisenberg model, we consider a system consisting of eight qubits ($N = 8$) and choose the fidelity as a measure for optimization, similarly to the setup for the results in Fig. 5. We set the antiferromagnetic initial state to $|\psi_i\rangle = |10101010\rangle$. The target state is the ground state of

[6] Fidelity serves as a metric for optimization. However, one caveat of utilizing fidelity is its reliance on the ground state. In this context, we assume the presence of an oracle capable of producing the fidelity value. Subsequently, we also employ energy as an observable metric for optimization purposes.

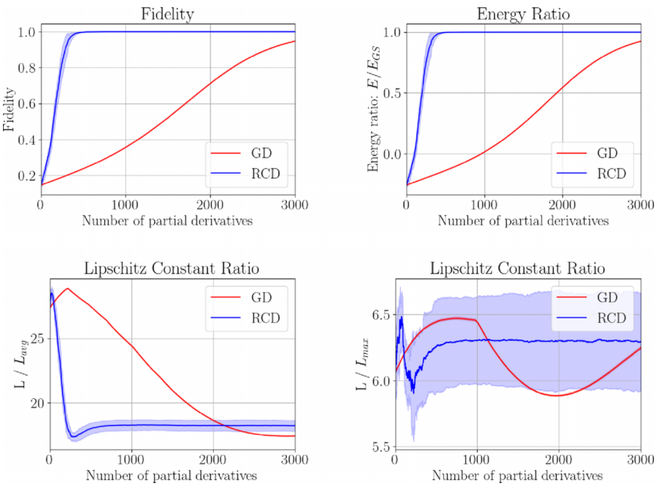


FIG. 6. Performance comparison between noisy GD and RCD for the Heisenberg model. The corresponding Lipschitz constant ratios, denoted as $\frac{L}{L_{\text{avg}}}$ and $\frac{L}{L_{\max}}$, are presented in the middle and right. The shaded areas within the figure represent variations that have been observed across 10 random realizations. The optimization is performed in dimensions of 28.

the Hamiltonian $H = H_1 + H_2$. We employ the QAOA ansatz represented by Eq. (26) and carry out the fidelity optimization detailed in Eq. (25).

Figure 6 showcases the performance outcomes from noisy GD and RCD simulations with learning rates set to 0.01 and 0.1, respectively. This QAOA model involves 28 parameters. The fidelity result shows that RCD converges to the target state much faster than GD. This phenomenon can be elucidated by noting that the ratios of Lipschitz constants derived from both noisy methods, $\frac{L}{L_{\text{avg}}}$ and $\frac{L}{L_{\max}}$, average around 10 and 6 along the trajectories, respectively. Especially, the magnitude of the ratio $\frac{L}{L_{\max}}$ is similar to that of the ratio of the numbers of partial derivative evaluations to reach a high fidelity >0.8 from both noisy methods, as shown in Fig. 6. Based on the observed numerical results, a high ratio of $\frac{L}{L_{\max}}$ is responsible for the efficiency of RCD in this optimization problem.

D. QAOA for classical combinatorial optimization problems

Quadratic unconstrained binary optimization (QUBO) problems have significant applications in fields such as finance, logistics, and machine learning, etc. Recognized as one prominent optimization model in quantum computing, QUBO consolidates a wide range of combinatorial optimization problems [75–78] and translates them into identifying the ground state of classical Ising models [79].

The goal of QUBO is to identify a sequence of binary variables (0 or 1) that minimize a quadratic function. Specifically, a cost function f_Q is constructed over the set of binary vectors, $\mathbb{B}^n$:

$$f_Q(x) = x^\top Q x = \sum_{i,j=1}^{n} Q_{ij} x_i x_j. \tag{27}$$

In this context, $\mathbb{B} = \{0, 1\}$ signifies the set of binary values (or bits), and $\mathbb{B}^n$ represents the collection of binary vectors with length $n > 0$. A symmetric, real-valued matrix $Q \in \mathbb{R}^{n\times n}$ is introduced, with each element Q_{ij} determining the weight for the corresponding pair of indices $i, j \in 1, \dots, n$. For example, if $i = j$, then the term $Q_{ii}x_i^2$ contributes Q_{ii} to the function value when $x_i = 1$. On the other hand, if $i \neq j$, then the term $Q_{ij}x_ix_j$ contributes Q_{ij} to the function value when both $x_i = 1$ and $x_j = 1$.

Overall, QUBO seeks to minimize the function f_Q over the set of binary vectors by determining an optimal minimizer x^*,

$$x^* = \arg\min_{x\in\mathbb{B}^n} f_Q(x). \tag{28}$$

Incorporating the variational quantum algorithm into QUBO, we reformulate the cost function using the following substitution:

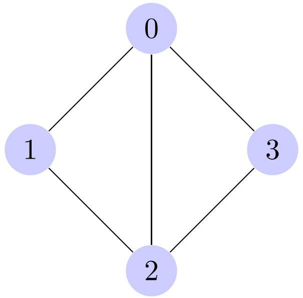
$$x_i = \frac{1 - Z_i}{2} \text{ or } \frac{1 + Z_i}{2}, \tag{29}$$

where the variable x_i is supplanted by the Pauli Z matrix operating on the i-th qubit. This replacement facilitates the formulation of a model Hamiltonian. Its ground state can be approximated by minimizing the expected energy via the variational quantum algorithm, as elaborated on in Sec. IV D 1.

In the following sections, we evaluate the performance of the noisy GD and RCD across various QUBO applications, focusing on the ground-state energy estimation. These applications encompass Max-Cut in Sec. IV D 1, the traveling salesman problem in Sec. IV D 2, and variational quantum factoring in Sec. IV D 3.

1. Max-Cut

For the Max-Cut problem, the graph employed in our numerical experiments is presented as follows:



The global cost function is designed to maximize $C = \sum_{(i,j)\in E} x_i(1 - x_j)$, where E represents the edges in the graph. For the given graph, the QUBO problem can be formulated as:

$$\min_{x_i\in\{0,1\}} -3x_0^2 + 2x_0x_1 + 2x_0x_2 + 2x_0x_3 - 2x_1^2 + 2x_1x_2 - 3x_2^2 + 2x_2x_3 - 2x_3^2.$$

In order to construct the corresponding Hamiltonian, we associate the binary variables x_i with the Pauli Z matrices, denoted as Z_i, which act on individual qubits. Taking into account the relationship between the binary variables x_i and the Pauli matrices Z_i, defined by the equation $x_i = \frac{1-Z_i}{2}$, the cost function is articulated by the Hamiltonian:

$$H = \tfrac{1}{2}I - 3Z_0 + \tfrac{1}{2}Z_0Z_1 + \tfrac{1}{2}Z_0Z_2 + \tfrac{1}{2}Z_0Z_3 + \tfrac{1}{2}Z_1Z_2 + \tfrac{1}{2}Z_2Z_3. \tag{30}$$

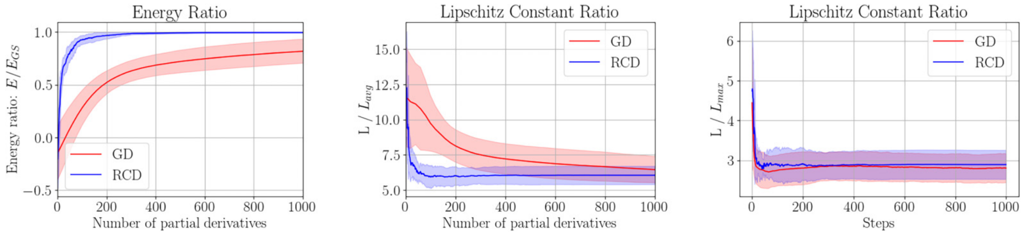


FIG. 7. Performance comparison between noisy GD and RCD for the Max-Cut problem. The corresponding Lipschitz constant ratios, denoted as $\frac{L}{L_{\text{avg}}}$ and $\frac{L}{L_{\max}}$, are presented in the middle and right panels. The shaded areas within the figure represent variations that have been observed across 10 random realizations. The optimization process has been performed in 20 dimensions.

Using this Hamiltonian, we construct a parameterized quantum circuit with four qubits ($N = 4$) and 20 parameters. The circuit consists of alternating single-gate rotations, denoted as $U_{\text{single}}\,(\boldsymbol{\theta}) = \prod_{i=1}^{n} \text{RY}(\theta_i)$[7] and entangler gate $U_{\text{entangler}}$.[8] This structure resembles the variational quantum circuit of the QAOA, with the ansatz given by $|\psi(\boldsymbol{\theta})\rangle = [U_{\text{single}}\,(\boldsymbol{\theta})U_{\text{entangler}}\,]^m|+\rangle$. For the optimization process, we assign a learning rate of 0.1 for GD and 3.0 for RCD and select energy as the optimization metric.

As illustrated in Fig. 7, the RCD also outperforms GD in this case, as it converges to an energy ratio of 1 with roughly 200 partial derivative evaluations. In contrast, the GD achieves only an average of 0.75 with 1000 derivative evaluations. The superior performance of RCD in Fig. 7 can again be attributed to the significant values of $\frac{L}{L_{\text{avg}}}$ and $\frac{L}{L_{\max}}$, both exceeding an order of magnitude of 3. As observed from the optimization result, a high ratio of $\frac{L}{L_{\text{avg}}}$ is indicative of the rapid convergence of RCD in this application.

2. *Traveling salesman problem*

We have designed a numerical test for the traveling salesman problem (TSP) using three cities as an example. The intercity costs for these cities are 48, 63, and 91, respectively. The cost of TSP is defined as

$$C(\mathbf{x}) = \sum_{i,j} w_{ij} \sum_{p} x_{i,p}x_{j,p+1} + A\sum_{p}\left(1 - \sum_{i} x_{i,p}\right)^2 + A\sum_{i}\left(1 - \sum_{p} x_{i,p}\right)^2,$$

where i labels the node, p indicates its order, and $x_{i,p}$ is in the set {0, 1} and the penalty parameter A is set sufficiently large to effectively enforce constraints. More details regarding the expansion of $C(\mathbf{x})$ can be found in Appendix G.

Utilizing the defined cost function, we establish a model Hamiltonian in the same manner as presented in Sec. IV D 1. We aim to prepare its ground state to address the QUBO problem. A detailed representation of the Hamiltonian is available in Appendix G. We construct a parameterized quantum circuit comprising alternating single-gate rotations, represented by $U_{\text{single}}\,(\boldsymbol{\theta}) = \prod_{i=1}^{n} \text{R}_{\text{Y}}(\theta_i)$ and entangler gate $U_{\text{entangler}}$. The total number of trainable parameters is 90, which requires nine qubits ($N = 9$) and 10 alternating layers. We employ energy as the measure for the optimization cost function.

In the left panel in Fig. 8, the optimization results obtained from the noisy RCD and GD are plotted. Notably, GD exhibits slower convergence compared to RCD in achieving an energy ratio of 1. The employment of 90 parameters in the optimization, a number markedly greater than those in prior applications, might account for this disparity. This increased parameter count likely requires additional iterations and partial derivative evaluations when applying GD. Similarly to previous results, the two types of Lipschitz constant ratios are obtained and shown along with the iterations in Fig. 8. Again, we can see that the values of the ratios are considerably large, especially during the initial stage of the optimization, underlining the efficiency of RCD in the optimization process.

3. *Variational quantum factoring*

Our next QUBO problem is designed as a variational quantum factoring task. For this task, we formulated the optimization problem within the framework of quantum adiabatic computation [80,81]. For example, to factorize 143 into the product of two prime numbers, let $143 = pq$, where

$$p = 8 + 4p_2 + 2p_1 + 1,$$
$$q = 8 + 4q_2 + 2q_1 + 1.$$

On direct computation, the relations are simplified to

$$p_1 + q_1 - 1 = 0, \tag{31}$$

$$p_2 + q_2 - 1 = 0, \tag{32}$$

$$p_2q_1 + p_1q_2 - 1 = 0. \tag{33}$$

To solve this system of equations, we introduce a cost function,

$$c(p_1, q_1, p_2, q_2) = (p_1 + q_1 - 1)^2 + (p_2 + q_2 - 1)^2 + (p_2q_1 + p_1q_2 - 1)^2. \tag{34}$$

[7] Each layer of rotation gates includes a rotation-Y gate applied to every qubit.

[8] The entanglement layer incorporates two-qubit gates for qubit entanglement without tunable parameters. In this experiment, the entangler gate employs controlled-Z gates. For a comprehensive explanation, refer to the circuit architecture in Appendix E.

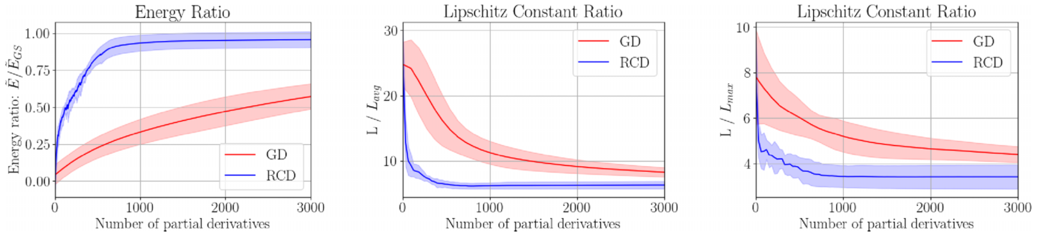


FIG. 8. Performance comparison between noisy GD and RCD for the TSP problem. The corresponding Lipschitz constant ratios, denoted as $\frac{L}{L_{\rm avg}}$ and $\frac{L}{L_{\max}}$, are presented in the middle and right panels. The shaded areas within the figure represent variations that have been observed across 10 random realizations. The optimization process has been performed in 90 dimensions. In the first panel, $\tilde{E}/\tilde{E}_{\rm GS}$ is defined as $(E-c)/(E_{\rm GS}-c)$, where $c/E_{\rm GS}=3000$. For clarity in the presentation, we adjust the energy by a constant.

By borrowing techniques (see Appendix H for more details) from Refs. [82,83], the cost function can be reduced to

$$\begin{aligned} &c(p_1,q_1,p_2,q_2)\\ &\quad=5-3p_1-p_2-q_1+2p_1q_1-3p_2q_1\\ &\qquad+2p_1p_2q_1-3q_2+p_1q_2+2p_2q_2+2p_2q_1q_2. \end{aligned} \tag{35}$$

Following the methods detailed in the QUBO, we treat (p_1,q_1,p_2,q_2) as Boolean functions and substitute each Boolean with $\frac{1}{2}(1-Z_i)$ as we did in previous sections. Then the problem can be reformulated into the Ising Hamiltonian,

$$\begin{aligned} H=&-3\mathbb{I}+\tfrac{1}{2}Z_0+\tfrac{1}{4}Z_1+\tfrac{3}{4}Z_0Z_2+\tfrac{1}{2}Z_2-\tfrac{1}{4}Z_1Z_2+\tfrac{1}{4}Z_0Z_1\\ &-\tfrac{1}{4}Z_0Z_1Z_2+\tfrac{1}{2}Z_3+\tfrac{1}{4}Z_0Z_3+\tfrac{3}{4}Z_1Z_3+\tfrac{1}{4}Z_2Z_3\\ &-\tfrac{1}{4}Z_1Z_2Z_3. \end{aligned} \tag{36}$$

The ground states of this Hamiltonian are $|0110\rangle$ and $|1001\rangle$, which respectively correspond to the solutions for the factorization of the number 143. We summarize it as follows:

$$(p_1,p_2,q_1,q_2)=(0,1,1,0)\longleftrightarrow(p,q)=(13,11), \tag{37}$$

$$(p_1,p_2,q_1,q_2)=(1,0,0,1)\longleftrightarrow(p,q)=(11,13), \tag{38}$$

$$\begin{aligned} p=8+4p_2+2p_1+1 \text{ and } q=8+4q_2\\ +2q_1+1 \quad \text{Boolean functions.} \end{aligned} \tag{39}$$

In our numerical experiment, we select the mixer Hamiltonian $H_2=\sum X_i$ and set up a 20-layer QAOA, which corresponds to 40 parameters.[9] We set the learning rates to 0.0001 for GD and 0.005 for RCD and choose the energy as a measure for optimization. Even with a small step size, the variance of GD is notably large. Employing a larger step size for GD further exacerbates the results.

In Fig. 9, the optimization results of the Hamiltonian (36) are depicted, showing that the number of partial derivative evaluations for the RCD to reach an energy ratio of 1 is about 400 whereas the GD seems to require more than 1000 to the same tolerance. As discussed previously, this observation aligns with prior discussions, particularly given the pronounced magnitude of the Lipschitz constant ratios evident in Fig. 9.

V. CONCLUSION

In this paper, we propose that the RCD method is a simple and effective algorithm for optimizing parameterized quantum circuits. The effectiveness of RCD is demonstrated using extensive numerical experiments for a variety of quantum optimization problems (all are nonconvex problems). In a noisy environment, we find that RCD's computational cost (measured by the number of partial derivative calculations) can be significantly lower than that of the gradient descent method. This suggests that in optimizing parameterized quantum

[9] The QAOA ansatz builds the variational circuit by alternating between the parametrized unitary evolution associated with the problem Hamiltonian H and the mixer Hamiltonian H_2.

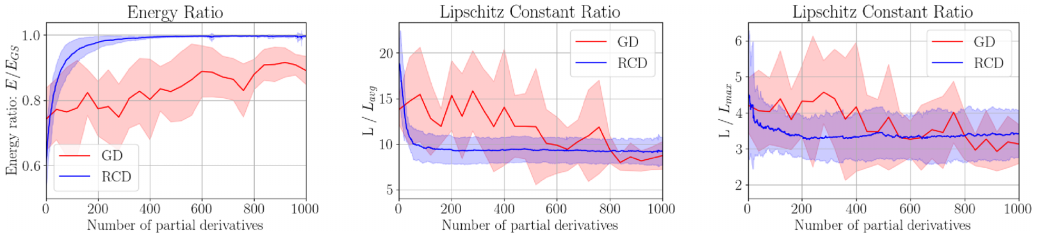


FIG. 9. Performance comparison between noisy GD and RCD for the quantum factoring problem. The corresponding Lipschitz constant ratios, denoted as $\frac{L}{L_{\rm avg}}$ and $\frac{L}{L_{\max}}$, are presented in the middle and right panels. The shaded areas within the figure represent variations that have been observed across 10 random realizations. The optimization process has been performed in 40 dimensions.

circuits, the benefits of developing an asymptotically efficient and yet complex quantum circuit for evaluating all partial derivatives, as discussed in Refs. [61,62,65], may be limited and demand re-evaluation.

From a theoretical perspective, most previous works on randomized coordinate descent algorithms studied the case of convex cost functions or functions satisfying the global PL condition, which do not fit into most variational quantum applications that involve nonconvex cost functions. In this work, we generalized the conventional convergence analysis of randomized coordinate descent to local convergence analysis under a local-PL condition that can capture a large class of nonconvex optimization. In particular, we proved that noisy randomized coordinate descent can converge faster than noisy gradient descent in terms of the total cost, measured in terms of the total number of partial derivative estimations. Relaxing the global PL condition to a local one leads to additional theoretical difficulties, namely the inherent noise can potentially displace parameters from the local basin. In our study, we introduce, to the best of our knowledge, a novel supermartingale approach to show that, with a proper initial condition, the parameters can stay within the basin and converge to a low-loss point. We believe that this analysis technique may also benefit the analysis of other quantum optimization algorithms. We also emphasize that our analysis does not guarantee the global convergence of the algorithm and, specifically, does not address its performance in the presence of barren plateaus [84].

Our analysis noise comes from quantum measurements. However, in real quantum devices, the presence of generic quantum noise may introduce additional noise to the problem. For example, circuit noise will have two effects on our optimization methods. First, it will increase the variance σ^2 associated with the gradient estimators and, as a result, increase the sampling complexity [85] or the number of iteration steps (Theorem 3 and Theorem 4). Second, noise may induce a bias that is proportional to the noise strength (λ) [85]. Although the bias can be improved by error mitigation techniques (see Refs. [86] and [87]), e.g., the zero noise extrapolation method, the variance overhead will persist and thus increase the optimization complexity, as indicated by Theorems 3 and 4. Meanwhile incorporating small bias into the convergence analysis seems possible, as suggested by previous works (refer, for instance, to Ref. [88]).

From an optimization standpoint, variational quantum optimization as outlined in Problem 1 also raises many interesting questions. For instance, in our experiment, we employ a fixed number of measurement shots for each step. However, in practical scenarios, implementing an adaptive approach to determine the number of shots could potentially reduce the overall shot count and increase the accuracy. When the estimation for $f(\theta)$ and $\nabla f(\theta)$ reveals significant magnitudes, increasing the number of shots can effectively reduce variance, ensuring high accuracy in the estimation process. Another interesting direction is to consider non-gradient-based optimization algorithm. For example, can second-order or zeroth-order optimization methods (i.e., methods using only function evaluation) be more efficient compared to the current gradient-based algorithms? In a technical viewpoint, another question is whether the stability result Lemma 5 can be generalized so that the event covers the case that the iteration diverges at some time instances, but it remains in the entire basin until then, $f^{-1}[0,\delta_f)$, not necessarily in the region above the set of global minima, $f^{-1}(\frac{La\sigma_\infty^2 d}{\mu},\delta_f)$. If this is possible to show, then it will provide a stronger result such as the stability of the noisy GD and RCD within the entire basin as the stability of Markov chain in Ref. [89]. Furthermore, the barren plateau [84] (i.e., vanishing gradients) poses a significant challenge for training algorithms for PQCs. Moreover, there remains a gap in the theoretical understanding of how the observed vanishing gradients relate to the assumptions underlying the analysis of optimization algorithms. A simple speculation is that the presence of barren plateau will make some of the constants (e.g., the local PL constant) in our analysis vanishingly small as the number of qubits and/or the circuit depth increases. It may be possible to combine RCD with some existing novel optimization algorithms, e.g., layerwise training [90], to partially mitigate the difficulty caused by the barren plateau in specific applications.

ACKNOWLEDGMENTS

T.K. and X.L.'s research is supported by the National Science Foundation Grants No. DMS-2111221 and No. CCF-2312456. T.K. is supported by a KIAS Individual Grant No. CG096001 at Korea Institute for Advanced Study. This material is based on work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator (Z.D.). Additional funding is provided by the SciAI Center funded by the Office of Naval Research (ONR), under Grant No. N00014-23-1-2729 (L.L.). L.L. is a Simons investigator in Mathematics.

APPENDIX A: STOCHASTIC STABILITY OF NOISY GD

In this section, we prove Lemma 5.

Proof of Lemma 5 Define the probability filtration: $\mathcal{F}_n = \sigma(\boldsymbol{\theta}_k | k \leqslant n)$ and the stopping time[10]

$$\tau = \inf\left\{k : f(\boldsymbol{\theta}_k) \leqslant \frac{La\sigma_\infty^2 d}{\mu}\right\},$$

which is the smallest timepoint where the noisy GD achieves $f(\boldsymbol{\theta}_k) \leqslant \frac{La\sigma_\infty^2 d}{\mu}$.

Define the indicator function $\mathbb{I}_n$:

$$\mathbb{I}_n = \begin{cases} 1, & \text{if} \quad \{\boldsymbol{\theta}_k\}_{k=1}^{n-1} \subset f^{-1}([0,\delta_f)) \\ 0, & \text{otherwise} \end{cases}, \tag{A1}$$

and the stochastic process

$$V_n = \begin{cases} f(\boldsymbol{\theta}_n)\mathbb{I}_n, & n < \tau \\ f(\boldsymbol{\theta}_\tau)\mathbb{I}_\tau, & n \geqslant \tau \end{cases}.$$

[10] It is straightforward to see

$$\{\tau \leqslant n\} \in \mathcal{F}_n, \quad \{\tau > n\} \in \mathcal{F}_n.$$

According to the definition of V_n, there are complementary and exclusive events (cases):

(i) Case 1: If there exists $0 < n < \infty$ such that: 1. $\boldsymbol{\theta}_n \notin \mathcal{N}$; 2. For any $m < n$, $\boldsymbol{\theta}_m \in \mathcal{N}$ and $f(\boldsymbol{\theta}_m) > \frac{La\sigma_\infty^2 d}{\mu}$. Then

$$V_n \geqslant \delta_f \Rightarrow \sup_n V_n \geqslant \delta_f .$$

(ii) Case 2: For any $n < \tau$, $f(\boldsymbol{\theta}_n) \in \mathcal{N}$.

We observe that Case 2 is the stable situation, indicating that $f(\boldsymbol{\theta}_n)$ remains in the basin of the global minimum until it achieves a small loss.[11] To prove (22), it suffices to show that

$$\mathbb{P}(\Omega_1) \leqslant \frac{f(\boldsymbol{\theta}_1)}{\delta_f}, \tag{A2}$$

where Ω_1 denotes the event associated with Case 1.

Now we show that V_n is a supermartingale to bound $\sup_n V_n$. Taking the conditional expectation, we obtain

$$\begin{aligned}\mathbb{E}(V_{n+1}|\mathcal{F}_n) = {} & \mathbb{E}(V_{n+1}|\mathcal{F}_n, \mathbb{I}_n = 1, \tau \leqslant n)\mathbb{P}(\tau \leqslant n)\\ & + \mathbb{E}(V_{n+1}|\mathcal{F}_n, \mathbb{I}_n = 1, \tau > n)\mathbb{P}(\tau > n),\end{aligned}$$

where we use $\mathbb{I}_{n+1} \leqslant \mathbb{I}_n$. There are two terms in the above equation:

(i) For the first term, when $\tau \leqslant n$, we obtain $V_{n+1} = V_\tau = V_n$. This implies

$$\mathbb{E}(V_{n+1}|\mathcal{F}_n, \mathbb{I}_n = 1, \tau \leqslant n) = V_n. \tag{A3}$$

(ii) For the second term, when $\tau > n$, we have $f(\boldsymbol{\theta}_n) > \frac{La\sigma_\infty^2 d}{\mu}$. Then taking the conditional expectation yields

$$\begin{aligned}&\mathbb{E}[V_{n+1}|\mathbb{I}_n = 1, \boldsymbol{\theta}_1, \tau > n]\\ &\quad = \mathbb{E}[f(\boldsymbol{\theta}_{n+1})\mathbb{I}_{n+1}|\mathbb{I}_n = 1, \boldsymbol{\theta}_1, \tau > n]\\ &\quad \leqslant f(\boldsymbol{\theta}_n) - a\|\nabla f(\boldsymbol{\theta}_n)\|^2 + \frac{La^2}{2}(\|\nabla f(\boldsymbol{\theta}_n)\|^2 + \sigma_\infty^2 d)\\ &\quad \leqslant (1-\mu a)f(\boldsymbol{\theta}_n) + \frac{La^2\sigma_\infty^2 d}{2}\\ &\quad < (1-\mu a)f(\boldsymbol{\theta}_n) + \frac{\mu a}{2}f(\boldsymbol{\theta}_n)\\ &\quad \leqslant \left(1 - \frac{\mu a}{2}\right)f(\boldsymbol{\theta}_n)\mathbb{I}_n = \left(1 - \frac{\mu a}{2}\right)V_n,\end{aligned} \tag{A4}$$

where we use Assumption 3 and $a < \frac{1}{L}$ in the second inequality, $\tau > n$ in the third inequality.

Combining (A3) and (A4), we obtain

$$\mathbb{E}(V_{n+1}|\mathcal{F}_n) = V_n\mathbb{P}(\tau \leqslant n) + \left(1 - \frac{\mu a}{2}\right)V_n\mathbb{P}(\tau > n) \leqslant V_n. \tag{A5}$$

Thus, V_n is a supermartingale.

Now, we consider the Case 1 event:

$$\begin{aligned}&\Omega_1 = \{\exists n > 1, \boldsymbol{\theta}_n \notin \mathcal{N} \text{ and } f(\boldsymbol{\theta}_m) > \epsilon \text{ with } \boldsymbol{\theta}_m \in \mathcal{N},\\ &\forall 1 \leqslant m < n\} \subset \{\sup_n V_n \geqslant \delta_f\}.\end{aligned}$$

Because V_n is a supermartingale, we obtain Case 1 happens with small probability:

$$\mathbb{P}(\Omega_1) \leqslant \frac{V_1}{\delta_f} = \frac{f(\boldsymbol{\theta}_1)}{\delta_f}.$$

This concludes the proof. ■

APPENDIX B: STOCHASTIC STABILITY OF NOISY RCD

In this section, we prove Lemma 6 with a slight modification of the proof in Appendix A. From a theoretical viewpoint, the difference between the noisy GD and RCD methods is made by the construction of gradient estimate [e.g., see (5) and (8)]. Compared to GD, the additional randomness of RCD comes with the random selection of a component as in (8). This difference affects the recursive inequality (A4) mainly in the previous proof, where we considered the properties of the gradient estimator. From this observation, it suffices to derive a recursive inequality similar to (A4) to prove Lemma 6.

Note that the sampling of a component within RCD is performed before estimating a partial derivative. Thus, the first step is to take expectation on the partial derivative estimate,

$$\begin{aligned}\mathbb{E}_{\xi_{i_n}}[f(\boldsymbol{\theta}_{n+1})] \leqslant {} & f(\boldsymbol{\theta}_n) - a\mathbb{E}_{\xi_{i_n}}[\partial_{i_n}f(\boldsymbol{\theta}_n)g_{i_n}(\boldsymbol{\theta}_n)]\\ & + \frac{L_{i_n}a^2}{2}\mathbb{E}_{\xi_{i_n}}[|g_{i_n}(\boldsymbol{\theta}_n)|^2],\end{aligned} \tag{B1}$$

where i_n is uniformly sampled index and g_{i_n} is the corresponding unbiased estimate for the partial derivative. Let $\mathcal{F}_n$, τ, $\mathbb{I}_n$, and V_n be as defined in the previous section. By considering the inequality (B1) and the conditional expectation in (A4), we achieve the following result by taking expectations with respect to the random index i_n:

$$\begin{aligned}&\mathbb{E}[V_{n+1}|\mathbb{I}_n = 1, \boldsymbol{\theta}_1, \tau > n]\\ &\quad = \mathbb{E}[f(\boldsymbol{\theta}_{n+1})\mathbb{I}_{n+1}|\mathbb{I}_n = 1, \boldsymbol{\theta}_1, \tau > n]\\ &\quad \leqslant \left[f(\boldsymbol{\theta}_n) - \frac{a}{d}\|\nabla f(\boldsymbol{\theta}_n)\|^2 + \frac{L_{\max}a^2}{2d}\|\nabla f(\boldsymbol{\theta}_n)\|^2\right.\\ &\quad\quad \left. + \frac{L_{\text{avg}}\sigma_\infty^2 d a^2}{2d}\right]\mathbb{I}_{n+1}\\ &\quad \leqslant \left[\left(1 - \frac{\mu a}{d}\right)f(\boldsymbol{\theta}_n) + \frac{L_{\text{avg}}a^2\sigma_\infty^2}{2}\right]\mathbb{I}_{n+1}\\ &\quad < \left[\left(1 - \frac{\mu a}{d}\right)f(\boldsymbol{\theta}_n) + \frac{\mu a}{2d}f(\boldsymbol{\theta}_n)\right]\mathbb{I}_{n+1}\\ &\quad = \left(1 - \frac{\mu a}{2d}\right)f(\boldsymbol{\theta}_n)\mathbb{I}_n = \left(1 - \frac{\mu a}{2d}\right)V_n,\end{aligned} \tag{B2}$$

provided that $f(\boldsymbol{\theta}_n) > \frac{L_{\text{avg}}a\sigma_\infty^2 d}{\mu}$ and $a_n = a < \min\{\frac{1}{L_{\max}}, \frac{d}{\mu}, \frac{2\mu\delta_f}{L_{\text{avg}}\sigma_\infty^2 d}\}$.

Similarly to (A5), in the case of RCD, (B2) implies

$$\mathbb{E}(V_{n+1}|\mathcal{F}_n) = V_n\mathbb{P}(\tau \leqslant n) + \left(1 - \frac{\mu a}{2d}\right)V_n\mathbb{P}(\tau > n) \leqslant V_n, \tag{B3}$$

which implies V_n forms a supermartingale. The remaining proof of Lemma 6 follows the same steps as the proof of Lemma 5, so we will not include them here.

[11] We emphasize that Case 2 also includes the situation where $f(\boldsymbol{\theta}_n)$ remains in the basin and never achieves the small loss.

APPENDIX C: THE PROOFS OF THEOREM 7 AND THEOREM THEOREM 8

We first show the convergence rate of the noisy GD method, followed by a similar analysis for the noisy RCD method. The following proofs are similar to those in Appendixes A and B with minor differences.

Proof of Theorem 7. Define the probability filtration: $\mathcal{F}_n = \sigma(\boldsymbol{\theta}_k | k \leqslant n)$ and the stopping time

$$\tau = \inf\{k : f(\boldsymbol{\theta}_k) \leqslant \epsilon\},$$

which is the smallest timepoint where the noisy GD achieves $f(\boldsymbol{\theta}_k) \leqslant \epsilon$. Then, our ultimate goal is to show that the event that $\inf_{1\leqslant n\leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon$ occurs with high probability, say, at least $1-\eta$. Then, our goal is to show that for any $\eta \in (\frac{f(\boldsymbol{\theta}_1)}{\delta_f}, 1)$, there exists a sufficiently large N such that

$$p_{\text{fail}} := \mathbb{P}(\tau > N) \leqslant \eta. \tag{C1}$$

Define the indicator function $\mathbb{I}_n$:

$$\mathbb{I}_n = \begin{cases} 1, & \text{if} \quad \{\boldsymbol{\theta}_k\}_{k=1}^{n-1} \subset f^{-1}([0, \delta_f)) \\ 0, & \text{otherwise} \end{cases},$$

and the stochastic process

$$V_n = \begin{cases} f(\boldsymbol{\theta}_n)\mathbb{I}_n, & n < \tau \\ f(\boldsymbol{\theta}_\tau)\mathbb{I}_\tau, & n \geqslant \tau \end{cases}.$$

Define the unstable event:

$$\Omega = \{\exists n > 1, \boldsymbol{\theta}_n \notin \mathcal{N} \text{ and } f(\boldsymbol{\theta}_m) > \epsilon, \; \forall 1 \leqslant m < n\} \subset \{\sup_n V_n \geqslant \delta_f\}.$$

According to Lemma 5 and the proof in Appendix A, for learning rate a with $\frac{La\sigma_\infty^2 d}{\mu} < \epsilon$, we obtain Ω happens with small probability:

$$\mathbb{P}(\Omega) \leqslant \frac{V_1}{\delta_f} = \frac{f(\boldsymbol{\theta}_1)}{\delta_f}. \tag{C2}$$

Recalling (C1), we note that, for any $n \leqslant N$,

$$\mathbb{P}(\tau > n) \geqslant p_{\text{fail}}.$$

Plugging this into (A5), we obtain that

$$\begin{aligned} \mathbb{E}(V_{n+1}|\mathcal{F}_n) &= \left[1 - \mathbb{P}(\tau > n) + \left(1 - \frac{\mu a}{2}\right)\mathbb{P}(\tau > n)\right]V_n \\ &= \left[1 - \frac{\mu a \mathbb{P}(\tau > n)}{2}\right]V_n \\ &\leqslant \left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)V_n. \end{aligned} \tag{C3}$$

By taking the total expectation on both sides and using a telescoping trick, we achieve that

$$\mathbb{E}(V_{n+1}) \leqslant \left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^n V_1 = \left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^n f(\boldsymbol{\theta}_1). \tag{C4}$$

This means that if the probability of failure, p_{fail}, is large, the expectation of V_{n+1} decreases quickly. By Markov's inequality, we have

$$\mathbb{P}(V_N > \epsilon) \leqslant \frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^{N-1} f(\boldsymbol{\theta}_1)}{\epsilon},$$

equivalently,

$$\mathbb{P}(V_N \leqslant \epsilon) \geqslant 1 - \frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^{N-1} f(\boldsymbol{\theta}_1)}{\epsilon}.$$

Now, if we consider the event $\{V_N \leqslant \epsilon\}$, then it is the union of the following two events (not necessarily exclusive and complementary), which are slightly different from the ones in Appendix A:

(i) Ω_1: There exists $n \leqslant N$ such that $f(\boldsymbol{\theta}_n) \leqslant \epsilon$ and $\boldsymbol{\theta}_n \in \mathcal{N}$. This means

$$\inf_{1\leqslant n\leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon.$$

We want to show that Ω_1 happens with high probability.

(ii) Ω_2: There exists $n < N$ such that $f(\boldsymbol{\theta}_n) > \delta_f$ and $f(\boldsymbol{\theta}_m) > \epsilon$ for any $m < n$.

We note that, when Ω_2 happens, we have $V_{n+1} = 0$ with $f(\boldsymbol{\theta}_n) > \delta_f$, which implies $\Omega_2 \subset \Omega$. According to (C2), we obtain

$$\mathbb{P}(\Omega_2) \leqslant \mathbb{P}(\Omega) \leqslant \frac{f(\boldsymbol{\theta}_1)}{\delta_f}.$$

Now we give a lower bound for the event Ω_1:

$$\begin{aligned} \mathbb{P}[\inf_{1\leqslant n\leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon] &= \mathbb{P}(\Omega_1) \geqslant \mathbb{P}(V_N \leqslant \epsilon) - \mathbb{P}(\Omega_2) \\ &\geqslant 1 - \frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^{N} f(\boldsymbol{\theta}_1)}{\epsilon} - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}. \end{aligned} \tag{C5}$$

Notice

$$\mathbb{P}[\inf_{1\leqslant n\leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon] \leqslant \mathbb{P}(\tau \leqslant N) = 1 - p_{\text{fail}}.$$

Combining the above two inequalities, we have

$$p_{\text{fail}} \leqslant \frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^{N} f(\boldsymbol{\theta}_1)}{\epsilon} + \frac{f(\boldsymbol{\theta}_1)}{\delta_f}. \tag{C6}$$

Next we show (C1) using the proof by contradiction. Assume that the conclusion of the theorem is not true, meaning that for some $\eta \in (\frac{f(\boldsymbol{\theta}_1)}{\delta_f}, 1)$ and every N,

$$p_{\text{fail}} > \eta.$$

When $p_{\text{fail}} > \eta$ and $N = \frac{2}{\mu a \eta} \ln\{\frac{f(\boldsymbol{\theta}_1)}{[\eta - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}]\epsilon}\}$, then

$$\begin{aligned} &\frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2}\right)^{N} f(\boldsymbol{\theta}_1)}{\epsilon} + \frac{f(\boldsymbol{\theta}_1)}{\delta_f} \\ &< \frac{\left(1 - \frac{\mu a \eta}{2}\right)^{N} f(\boldsymbol{\theta}_1)}{\epsilon} + \frac{f(\boldsymbol{\theta}_1)}{\delta_f} \\ &\leqslant \frac{\exp\left(-\frac{\mu a \eta N}{2}\right) f(\boldsymbol{\theta}_1)}{\epsilon} + \frac{f(\boldsymbol{\theta}_1)}{\delta_f} = \eta < p_{\text{fail}}, \end{aligned}$$

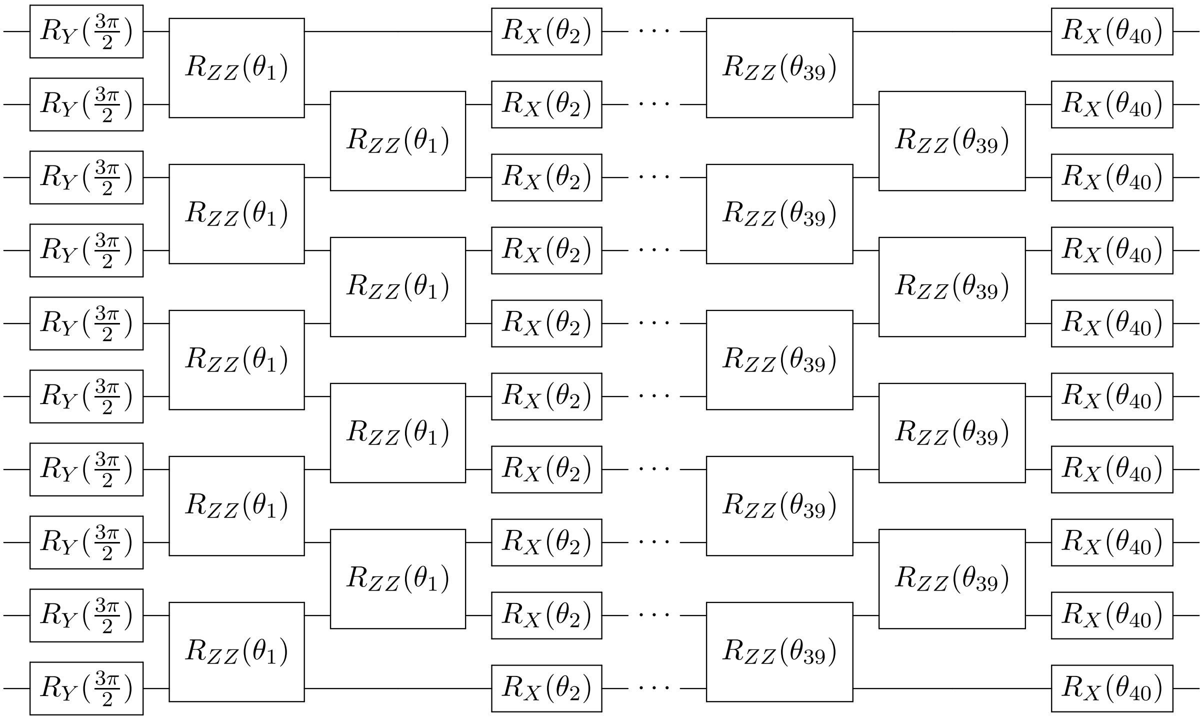


FIG. 10. A QAOA-like ansatz motivated by Ref. [91] is used for the TFIM model (10) with 10 qubits. For the result in Fig. 1, 36 parameters are assigned with 18 layers of alternating rotation ZZ gates and rotation X gates.

where we use $p_{\text{fail}} > \eta$ in the first inequality and $(1 - x)^N \leqslant \exp(-xN)$ in the second inequality. This contradicts to (C6). Thus, (C1) must be true and we conclude the proof. ■

Proof of Theorem 8. Denote the probability of failure

$$p_{\text{fail}} = \mathbb{P}(\tau > N).$$

Similarly to the calculation in the previous proof, from (B3), we have

$$\mathbb{P}[\inf_{1 \leqslant n \leqslant N} f(\boldsymbol{\theta}_n) \leqslant \epsilon] \geqslant 1 - \frac{\left(1 - \frac{\mu a p_{\text{fail}}}{2d}\right)^N f(\boldsymbol{\theta}_1)}{\epsilon} - \frac{f(\boldsymbol{\theta}_1)}{\delta_f}. \tag{C7}$$

With the same logic below (C5), we conclude the proof of theorem Theorem 8. ■

APPENDIX D: PARAMETERIZED CIRCUIT FOR THE VQE

The quantum circuit described in Fig. 10 is utilized for the numerical result in Fig. 1.

APPENDIX E: PARAMETERIZED CIRCUIT FOR THE VQE IN QUBO EXPERIMENTS

Figure 11 depicts the quantum circuit for the QUBO experiments in Secs. IV D 1, IV D 2, and IV D 3.

APPENDIX F: ADDITIONAL HISTOGRAMS OF PARTIAL DERIVATIVE ESTIMATES

Figure 1 plots the histograms with respect to the first 12 parameters among 36. The rest of 24 histograms are shown in

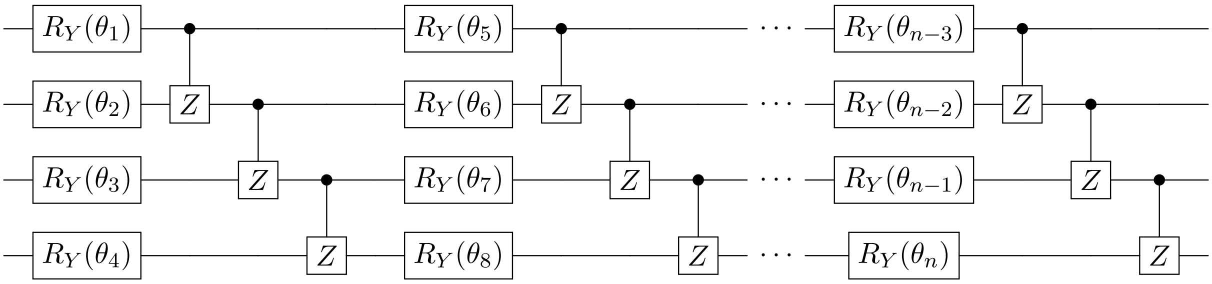


FIG. 11. A parametrized quantum circuit is employed in the QUBO experiments. This circuit features alternating layers of single rotation gates and entangling controlled-z gates. The adjustable parameters are exclusively found in the single rotation gates, and these parameters vary across different layers and qubits.

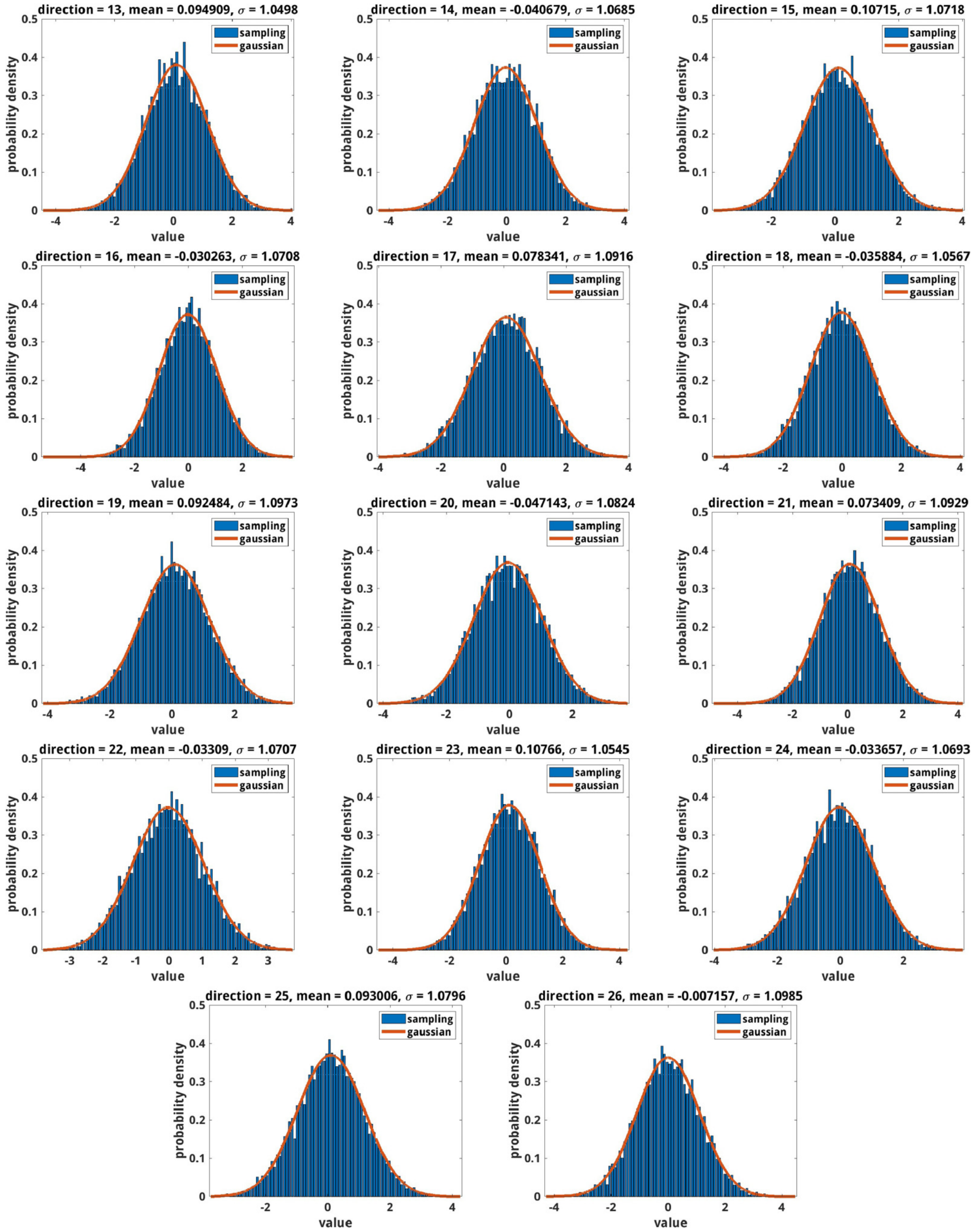


FIG. 12. The histogram of partial derivative estimates with respect to the 13th to the 26th parameters are plotted with the same setup as in Fig. 2.

Figs. 12 and 13. It is observed in all figures that the variances of partial derivative estimates in all directions are a similar magnitude of value.

APPENDIX G: COST FUNCTION FOR THE TSP

First, the cost function is defined as

$$C(\mathbf{x}) = \sum_{i,j} w_{ij} \sum_{p} x_{i,p} x_{j,p+1} + A \sum_{p} \left(1 - \sum_{i} x_{i,p}\right)^2 + A \sum_{i} \left(1 - \sum_{p} x_{i,p}\right)^2,$$

where $A = 10\,000$, $w_{12} = w_{21} = 48$, $w_{13} = w_{31} = 91$, and $w_{23} = w_{32} = 63$, and $w_{ii} = 0$, $i = 1, 2, 3$.

We can introduce a new Boolean variable, denoted by $\tilde{x}_{3i+j-4} = x_{i,j}$, where $i, j = 1, 2, 3$. For simplicity, in the following formula, we will use $x_0, \dots, x_8$ to represent $\tilde{x}_0, \dots, \tilde{x}_8$. With this notation, the expanded form of the cost function can be expressed as:

$$\begin{aligned} C(\boldsymbol{x}) = & -200000x_0 - 200000x_1 - 200000x_2 - 200000x_3 \\ & - 200000x_4 - 200000x_5 - 200000x_6 - 200000x_7 \\ & - 200000x_8 + [200000x_0x_1 + 200000x_0x_2 \\ & + 200000x_0x_3 + 48x_0x_4 + 48x_0x_5 + 200000x_0x_6 \end{aligned}$$

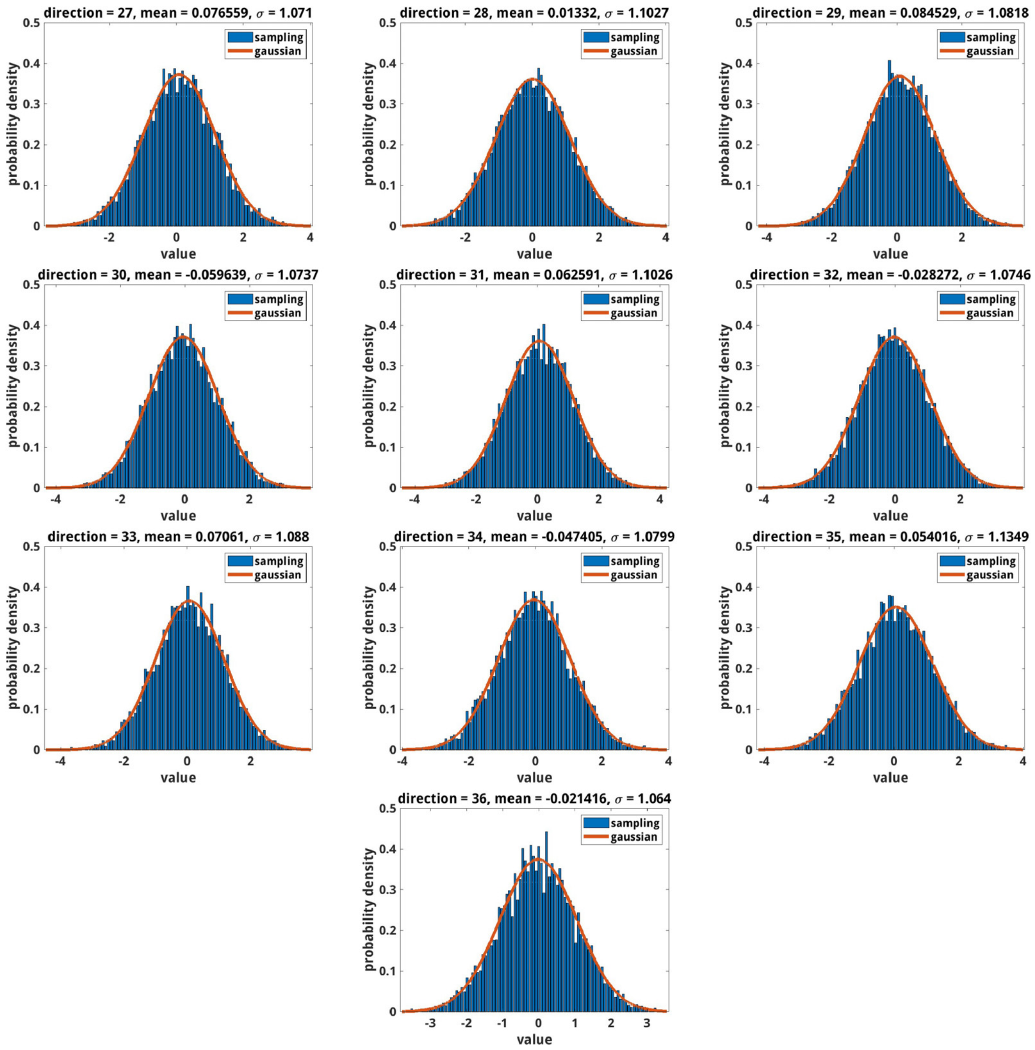


FIG. 13. The histogram of partial derivative estimates with respect to the 27th to the 36th parameters are plotted with the same setup as in Fig. 2.

$$
\begin{aligned}
&+ 91x_0x_7 + 91x_0x_8 + 200000x_1x_2 + 48x_1x_3\\
&+ 200000x_1x_4 + 48x_1x_5 + 91x_1x_6 + 200000x_1x_7\\
&+ 91x_1x_8 + 48x_2x_3 + 48x_2x_4 + 200000x_2x_5\\
&+ 91x_2x_6 + 91x_2x_7 + 200000x_2x_8 + 200000x_3x_4\\
&+ 200000x_3x_5 + 200000x_3x_6 + 63x_3x_7 + 63x_3x_8\\
&+ 200000x_4x_5 + 63x_4x_6 + 200000x_4x_7 + 63x_4x_8\\
&+ 63x_5x_6 + 63x_5x_7 + 200000x_5x_8 + 200000x_6x_7\\
&+ 200000x_6x_8 + 200000x_7x_8] + 600000.
\end{aligned}
$$

In order to build the corresponding Hamiltonian, we align the binary variables x_i with the Pauli Z matrices, which operate on individual qubits, and are represented by Z_i. Taking into account the relationship between the binary variables x_i and the Pauli Z matrices, defined by the equation $x_i = \frac{1-Z_i}{2}$, we can express the Hamiltonian for QUBO as follows,

$$
\begin{aligned}
H_{\text{TSP}} = {} & 600303.0 - 100069.5Z_0 - 100055.5Z_4 + 12.0Z_4Z_0\\
& - 100069.5Z_1 - 100055.5Z_5 + 12.0Z_5Z_1\\
& - 100069.5Z_2 - 100055.5Z_3 + 12.0Z_3Z_2\\
& - 100077.0Z_7 + 22.75Z_7Z_0 - 100077.0Z_8\\
& + 22.75Z_8Z_1 - 100077.0Z_6 + 22.75Z_6Z_2\\
& + 12.0Z_3Z_1 + 12.0Z_4Z_2 + 12.0Z_5Z_0\\
& + 15.75Z_7Z_3 + 15.75Z_8Z_4 + 15.75Z_6Z_5\\
& + 22.75Z_6Z_1 + 22.75Z_7Z_2 + 22.75Z_8Z_0\\
& + 15.75Z_6Z_4 + 15.75Z_7Z_5 + 15.75Z_8Z_3\\
& + 50000.0Z_3Z_0 + 50000.0Z_6Z_0 + 50000.0Z_6Z_3\\
& + 50000.0Z_4Z_1 + 50000.0Z_7Z_1 + 50000.0Z_7Z_4\\
& + 50000.0Z_5Z_2 + 50000.0Z_8Z_2 + 50000.0Z_8Z_5\\
& + 50000.0Z_1Z_0 + 50000.0Z_2Z_0 + 50000.0Z_2Z_1\\
& + 50000.0Z_4Z_3 + 50000.0Z_5Z_3 + 50000.0Z_5Z_4\\
& + 50000.0Z_7Z_6 + 50000.0Z_8Z_6 + 50000.0Z_8Z_7
\end{aligned}
$$

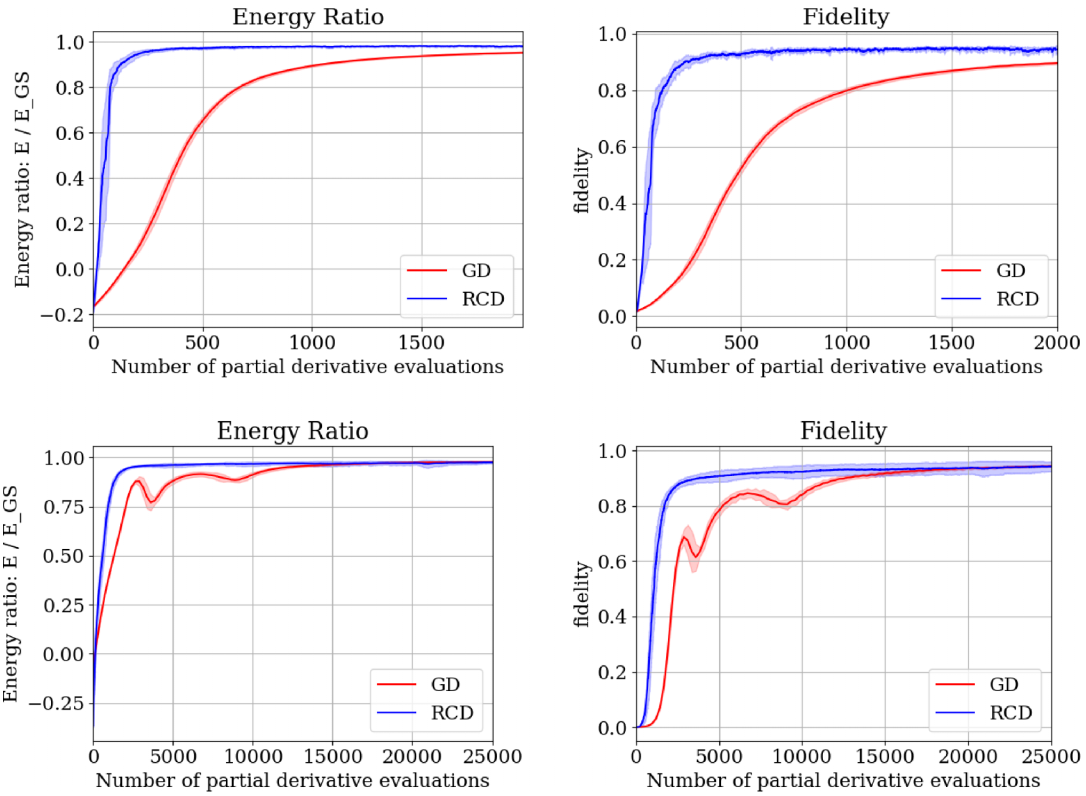


FIG. 14. Comparison of the performance of GD (red) and RCD (blue) for optimizing the Hamiltonian (10). The result in the top panels is obtained from a slight modification of the ansatz in Ref. [92]. For the result in the bottom panels, we used an ansatz [93].

APPENDIX H: TECHNIQUE USED IN QUANTUM FACTORING

The introduced technique proposes an alternative formulation for equations of the type $AB + S = 0$. Here A and B represent Boolean variables, while S denotes integers with $S \in \mathbb{Z}$. The optimization algorithm targets the minimization of the quadratic version of this equation.

Given the problem Hamiltonian, defined as $H = (AB + S)^2$, it can be restructured as:

$$H = 2\left[\frac{1}{2}\left(A + B - \frac{1}{2}\right) + S\right]^2 - \frac{1}{8}. \tag{H1}$$

While the two Hamiltonians are not generally equivalent, they do share the same minimizer due to their underlying Boolean function properties. For instance:

(i) When $AB = 1$: The minimizer for the first Hamiltonian dictates $S = -1$. In the reformulated version, we get

$$\begin{aligned} H &= 2\left[\frac{1}{2}\left(1 + 1 - \frac{1}{2}\right) - 1\right]^2 - \frac{1}{8} \\ &= 0. \end{aligned}$$

(ii) When $AB = 0$: At least one of A or B is zero. Assuming $A = 0$ (without loss of generality) and due to the minimizer, we get $S = 0$. This also minimizes the reformulated Hamiltonian since, regardless of whether B is 0 or 1, the result remains 0.

Thus, the reformulated version can be employed interchangeably in certain scenarios. However, this updated representation leads to a significant reduction in the many-body interactions observed experimentally. Specifically, the quartic terms in the Ising Hamiltonian are eliminated, simplifying experimental realizations. As a result, the third Hamiltonian term $(p_2q_1 + p_1q_2 - 1)^2$ in Eq. (34) is reformulated as:

$$H' = 2\left[\frac{1}{2}\left(p_1 + q_2 - \frac{1}{2}\right) + p_2q_1 - 1\right]^2 - \frac{1}{8}.$$

APPENDIX I: ADDITIONAL RESULT FOR THE TFIM WITH DIFFERENT ANSATZES

In addition to the QAOA-like ansatz depicted in Fig. 10, we incorporate hardware-efficient ansatzes [92,93] to evaluate the GD and RCD methods within the context of TFIM (10). The ansatzes, named HEA and HEA2 following the works of Refs. [92] and [93], respectively, were modified in our approach. Notably, we replaced the CNOT gates with rotation ZZ gates in the HEA configuration. For our numerical analysis in Fig. 14, HEA was configured with 36 parameters while HEA2 was equipped with 240 parameters, which correspond to 18 layers of HEA and 12 layers of HEA2. We also did numerical tests for the two ansatzes with more parameters, for example, 30 layers of HEA and 18 layers of HEA2. We found that performance with RCD method applied to the ansatzes with more layers was degraded in comparison to the case of the smaller numbers of layers.

Our analysis revealed a notable performance disparity between the two ansatzes, with the RCD method demonstrating superior efficiency over GD, as depicted in Fig. 14.

Specifically, the optimization of HEA2 required a significantly larger number of derivative computations in comparison to HEA. This discrepancy is largely attributed to the structural differences between HEA and HEA2. In HEA2, each parameter influences a singular qubit rotation Pauli gate, whereas in HEA, a single parameter adjustment can concurrently modify multiple gates. Thus, parameter alterations in HEA2 affect only one parameterized gate at a time, contrasting with HEA where multiple gates are simultaneously updated with each parameter change.

From the results shown in Figs. 1 and 14, our findings indicate that achieving a fidelity threshold of 0.9 with HEA2 demands a substantially higher number of partial derivative calculations than with HEA and the QAOA-like ansatz illustrated in Fig. 10. We also see that the QAOA-like ansatz performs slightly better than HEA. Precisely, the maximum and average fidelities at the last iterates of 10 simulations with the QAOA-like ansatz are 0.99 and 0.97, whereas those with HEA are 0.95 and 0.94, respectively. This comparison underscores the relative efficacy of QAOA-like ansatz in approximating the ground state of the TFIM (10) over the hardware-efficient designs HEA and HEA2, despite their intended hardware optimization benefits.

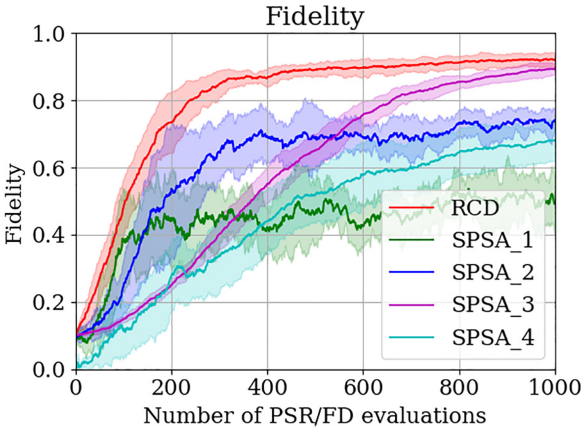


FIG. 15. Comparison of the performance of RCD (red) and SPSA (other colors) for optimizing the Hamiltonian of the TFIM (10) defined in Sec. I E. The x axis labels the number of parameter shift rule (PSR) or finite-difference (FD) evaluations as an indication of the computational complexity, which corresponds to RCD and SPSA, respectively. Both evaluations involve the same measurement complexity. The y axis indicates the fidelity with the ground state from the iterations.

APPENDIX J: COMPARISON OF RCD WITH SPSA USING THE TFIM MODEL

We now present results from another numerical experiment where we compare the performance of RCD and SPSA. The numerical tests are based on the TFIM model (10) using the ansatz shown in Fig. 10. Specifically, we consider the problem (10) with $n = 12$, the number of qubits. In the implementation, both methods are subject to measurement noise, and we set the number of shots to 1000 to estimate gradients within both optimization methods. We assigned 80 trainable parameters to the ansatz, that is, 40 layers within the circuit in Fig. 10. Figure 15 shows results from five numerical experiments. All results are obtained from 10 independent simulations with the same fixed initial configuration, but different learning rates, specifically $a = 0.008$ for RCD, $a = 0.001, 0.0005, 0.00001$ for SPSA_1, SPSA_2, SPSA_3, and a default setting in Qiskit for SPSA_4. Note that the two methods have the same complexity of derivative estimation, employing the finite-difference formula (SPSA) and the parameter shift rule (RCD).

From Fig. 15, it is evident that RCD outperforms SPSA significantly, even after fine-tuning the parameters of SPSA. We observe that hyperparameter tuning in RCD is easier than that in SPSA: RCD requires only a constant learning rate, while SPSA requires scheduling a decreasing learning rate and a diminishing perturbation parameter for the finite-difference scheme. To obtain successful SPSA training, a small learning rate must be selected. Compared to RCD, to achieve stable training for SPSA, the learning rate must be selected 80 times smaller, rendering the algorithm much less efficient than RCD. This can be clearly observed in Fig. 15, where RCD (with learning rate $a = 8 \times 10^{-3}$) attains a fidelity exceeding 0.95 at the end, already reaching over 0.9 between 200 and 400 parameter shift rule evaluations. In contrast, only SPSA_3 (with learning rate $a = 10^{-4}$) achieves a fidelity greater than 0.9, but with a much slower rate.

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[2] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, Quantum optimization using variational algorithms on near-term quantum devices, Quant. Sci. Technol. **3**, 030503 (2018).

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

[4] M. Benedetti, M. Fiorentini, and M. Lubasch, Hardware-efficient variational quantum algorithms for time evolution, Phys. Rev. Res. **3**, 033083 (2021).

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

[6] J. Leng, Y. Peng, Y.-L. Qiao, M. Lin, and X. Wu, Differentiable analog quantum computing for optimization and control, in *NeurIPS 2022*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates Inc., Red Hook, NY, 2022).

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

[8] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, Nat. Comput. Sci. **2**, 567 (2022).

[9] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, Nat. Comput. Sci. **3**, 542 (2023).

[10] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, Contemp. Phys. **56**, 172 (2015).

[11] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, J. Phys. Soc. Jpn. **90**, 032001 (2021).

[12] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[13] S. Altares-López, A. Ribeiro, and J. J. García-Ripoll, Automatic design of quantum feature maps, Quant. Sci. Technol. **6**, 045015 (2021).

[14] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quant. Sci. Technol. **4**, 043001 (2019).

[15] M. Ostaszewski, E. Grant, and M. Benedetti, Structure optimization for parameterized quantum circuits, Quantum **5**, 391 (2021).

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

[17] L. Banchi and G. E. Crooks, Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule, Quantum **5**, 386 (2021).

[18] G. E. Crooks, Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition, arXiv:1905.13311.

[19] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).

[20] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).

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

[22] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM J. Opt. **22**, 341 (2012).

[23] P. Richtárik and M. Takááž, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function, Math. Program. **144**, 1 (2014).

[24] S. J. Wright, Coordinate descent algorithms, Math. Program. **151**, 3 (2015).

[25] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. **60**, 223 (2018).

[26] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan, On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points, J. ACM **68**, 1 (2021).

[27] C. Liu, L. Zhu, and M. Belkin, Loss landscapes and optimization in over-parameterized non-linear systems and neural networks, Astron. Astrophys. **666**, A85 (2022).

[28] E. Moulines and F. R. Bach, Non-asymptotic analysis of stochastic approximation algorithms for machine learning, in *NeurIPS 2011*, edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Curran Associates, Inc., Red Hook, NY, 2011), Vol. 24.

[29] L. M. Nguyen, P. Ha Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč, SGD and Hogwild! Convergence without the bounded gradients assumption, in *ICML 2018*, edited by J. Dy and A. Krause (PMLR, 2018), Vol. 80.

[30] P. Ha Nguyen, L. M. Nguyen, and M. van Dijk, Tight dimension independent lower bound on optimal expected convergence rate for diminishing step sizes in SGD, in *NeurIPS 2019*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates Inc., Red Hook, NY, 2019).

[31] A. Rakhlin, O. Shamir, and K. Sridharan, Making gradient descentoptimal for strongly convex stochastic optimization, in *ICML 2012*, edited by J. Langford and J. Pineau (Omnipress, Madison, WI, 2012), pp. 1571–1578.

[32] B. Polyak, Gradient methods for the minimisation of functionals, USSR Comput. Math. **3**, 864 (1963).

[33] L. Gentini, A. Cuccoli, S. Pirandola, P. Verrucchi, and L. Banchi, Noise-resilient variational hybrid quantum-classical optimization, Phys. Rev. A **102**, 052414 (2020).

[34] Z. Chen, Y. Li, and J. Lu, On the global convergence of randomized coordinate gradient descent for nonconvex optimization, SIAM J. Opt. **33**, 713 (2023).

[35] Y. T. Lee and A. Sidford, Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems, in *FOCS 2013*, edited by D. Shmoys and O. Reingold (IEEE Computer Society, Los Alamitos, CA, USA, 2013), pp. 147–156.

[36] Y. Nesterov and S. U. Stich, Efficiency of the accelerated coordinate descent method on structured optimization problems, SIAM J. Opt. **27**, 110 (2017).

[37] B. Fehrman, B. Gess, and A. Jentzen, Convergence rates forthe stochastic gradient descent method for non-convex objective functions, J. Mach. Learn. Res. **21**, 5354 (2020).

[38] T. Ko and X. Li, A local convergence theory for the stochastic gradient descent method in non-convex optimization with non-isolated local minima, JML **2**, 138 (2023).

[39] P. Mertikopoulos, N. Hallak, A. Kavis, and V. Cevher, On the almost sure convergence of stochastic gradient descent in non-convex problems, in *NeurIPS 2020*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H.-T. Lin (Curran Associates, Inc., Red Hook, NY, 2020), Vol. 33, pp. 1117–1128.

[40] V. Patel, S. Zhang, and B. Tian, Global convergence and stability of stochastic gradient descent, in *NeurIPS 2022*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., Red Hook, NY, USA, 2022).

[41] J. An and J. Lu, Convergence of stochastic gradient descent under a local Lojasiewicz condition for deep neural networks, arXiv:2304.09221.

[42] J. C. Spall, A stochastic approximation technique for generating maximum likelihood parameter estimates, in *1987 American Control Conference*, edited by T. F. Edgar and J. L. Speyer (IEEE, New York City, NY, USA, 1987), pp. 1161–1167.

[43] J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Trans. Autom. Contr. **37**, 332 (1992).

[44] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. Chow, and J. Gambetta, Hardware-efficient variational quantum

eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[45] J. Yao, M. Bukov, and L. Lin, Policy gradient based quantum approximate optimization algorithm, in *MSML 2020*, edited by J. Lu and R. Ward (PMLR, 2020), Vol. 107, pp. 605–634.

[46] K. J. Sung, J. Yao, M. P. Harrigan, N. C. Rubin, Z. Jiang, L. Lin, R. Babbush, and J. R. McClean, Using models to improve optimizers for variational quantum algorithms, Quant. Sci. Technol. **5**, 044008 (2020).

[47] M. Bukov, Reinforcement learning for autonomous preparation of floquet-engineered states: Inverting the quantum Kapitza oscillator, Phys. Rev. B **98**, 224305 (2018).

[48] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, Phys. Rev. X **8**, 031086 (2018).

[49] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li, Quantum circuit optimization with deep reinforcement learning, arXiv:2103.07585.

[50] J. Yao, P. Kottering, H. Gundlach, L. Lin, and M. Bukov, Noise-robust end-to-end quantum control using deep autoregressive policy networks, in *MSML 2021*, edited by J. Bruna, J. Hesthaven, and L. Zdeborova (PMLR, 2021), Vol. 107, pp. 1044–1081.

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

[52] B. Rosenhahn and T. J. Osborne, Monte Carlo graph search for quantum circuit optimization, Phys. Rev. A **108**, 062615 (2023).

[53] J. Yao, H. Li, M. Bukov, L. Lin, and L. Ying, Monte Carlo tree search based hybrid optimization of variational quantum circuits, in *MSML 2022*, edited by B. Dong, Q. Li, L. Wang, and Z.-Q. J. Xu (PMLR, 2022), Vol. 190, pp. 49–64.

[54] A. Anand, M. Degroote, and A. Aspuru-Guzik, Natural evolutionary strategies for variational quantum computation, Mach. Learn. Sci. Technol. **2**, 045012 (2021).

[55] A. Giovagnoli, Y. Ma, and V. Tresp, Qneat: Natural evolution of variational quantum circuit architecture, arXiv:2304.06981.

[56] T. Zhao, G. Carleo, J. Stokes, and S. Veerapaneni, Natural evolution strategies and variational Monte Carlo, Mach. Learn.: Sci. Technol. **2**, 02LT01 (2020).

[57] S. Tamiya and H. Yamasaki, Stochastic gradient line bayesian optimization for efficient noise-robust optimization of parameterized quantum circuits, npj Quant. Inf. **8**, 90 (2022).

[58] S. Tibaldi, D. Vodola, E. Tignone, and E. Ercolessi, Bayesian optimization for QAOA, arXiv:2209.03824.

[59] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque *et al.*, Training of quantum circuits on a hybrid quantum computer, Sci. Adv. **5**, eaaw9918 (2019).

[60] P. Gao, K. Li, S. Wei, J. Gao, and G. Long, Quantum gradient algorithm for general polynomials, Phys. Rev. A **103**, 042403 (2021).

[61] A. Gilyén, S. Arunachalam, and N. Wiebe, Optimizing quantum optimization algorithms via faster quantum gradient computation, in *SODA 2019*, edited by T. M. Chan (SIAM, Philadelphia, PA, USA, 2019), pp. 1425–1444.

[62] S. P. Jordan, Fast quantum algorithm for numerical gradient estimation, Phys. Rev. Lett. **95**, 050501 (2005).

[63] P. Rebentrost, M. Schuld, L. Wossnig, F. Petruccione, and S. Lloyd, Quantum gradient descent and Newton's method for constrained polynomial optimization, New J. Phys. **21**, 073023 (2019).

[64] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).

[65] A. Abbas, R. King, H.-Y. Huang, W. J. Huggins, R. Movassagh, D. Gilboa, and J. R. McClean, On quantum backpropagation, information reuse, and cheating measurement collapse, in *NeurIPS 2023* (Curran Associates Inc., Red Hook, NY, 2023).

[66] Qiskit Community, Qiskit: An open-source framework for quantum computing, Zenodo (2017), doi:10.5281/zenodo.2562110.

[67] R. Bassily, M. Belkin, and S. Ma, On exponential convergence of SGD in non-convex over-parametrized learning, arXiv:1811.02564.

[68] B. Kleinberg, Y. Li, and Y. Yuan, An alternative view: When does SGD escape local minima? in *ICML 2018* (PMLR, 2018), pp. 2698–2707.

[69] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Array programming with NumPy, Nature (London) **585**, 357 (2020).

[70] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python+NumPy programs (2018) [http://github.com/google/jax].

[71] W. Chen, K. Hida, and B. C. Sanctuary, Ground-state phase diagram of $S = 1XXZ$ chains with uniaxial single-ion-type anisotropy, Phys. Rev. B **67**, 104401 (2003).

[72] A. Langari, F. Pollmann, and M. Siahatgar, Ground-state fidelity of the spin-1 Heisenberg chain with single ion anisotropy: Quantum renormalization group and exact diagonalization approaches, J. Phys.: Condens. Matter **25**, 406002 (2013).

[73] F. Pollmann, A. M. Turner, E. Berg, and M. Oshikawa, Entanglement spectrum of a topological phase in one dimension, Phys. Rev. B **81**, 064439 (2010).

[74] J. Yao, L. Lin, and M. Bukov, Reinforcement learning for many-body ground-state preparation inspired by counterdiabatic driving, Phys. Rev. X **11**, 031070 (2021).

[75] M. Anthony, E. Boros, Y. Crama, and A. Gruber, Quadratic reformulations of nonlinear binary optimization problems, Math. Program. **162**, 115 (2017).

[76] P. Díez-Valle, D. Porras, and J. José García-Ripoll, Quantum variational optimization: The role of entanglement and problem hardness, Phys. Rev. A **104**, 062426 (2021).

[77] F. Glover, G. Kochenberger, and Y. Du, Quantum bridge analytics I: A tutorial on formulating and using QUBO models, Ann. Oper. Res. **314**, 141 (2022).

[78] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, The unconstrained binary quadratic programming problem: A survey, J. Comb. Opt. **28**, 58 (2014).

[79] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[80] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, Variational quantum factoring, in *QTOP 2019* (Springer, Berlin, 2019), pp. 74–85.

[81] N. S. Dattani and N. Bryans, Quantum factorization of 56153 with only 4 qubits, arXiv:1411.6758.

[82] G. Schaller and R. Schützhold, The role of symmetries in adiabatic quantum algorithms, Quantum Inf. Comput. **10**, 109 (2010).

[83] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system, Phys. Rev. Lett. **108**, 130501 (2012).

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

[85] J. Sun, X. Yuan, T. Tsunoda, V. Vedral, S. C. Benjamin, and S. Endo, Mitigating realistic noise in practical noisy intermediate-scale quantum devices, Phys. Rev. Appl. **15**, 034026 (2021).

[86] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, Phys. Rev. X **8**, 031027 (2018).

[87] S. Wang, P. Czarnik, A. Arrasmith, M. Cerezo, L. Cincio, and P. J. Coles, Can error mitigation improve trainability of noisy variational quantum algorithms? Quantum **8**, 1287 (2024).

[88] A. Ajalloeian and S. U. Stich, On the convergence of SGD with biasedgradients, arXiv:2008.00051.

[89] H. J. Kushner and G. George Yin, *Applications to Learning, State Dependent Noise, and Queueing* (Springer, Berlin, 1997), pp. 25–46.

[90] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, Layerwise learning for quantum neural networks, Quant. Mach. Intell. **3**, 5 (2021).

[91] 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**, 020319 (2020).

[92] B. Choy and D. J. Wales, Molecular energy landscapes of hardware-efficient ansatze in quantum computing, J. Chem. Theory Comput. **19**, 1197 (2023).

[93] N. Yoshioka, H. Hakoshima, Y. Matsuzaki, Y. Tokunaga, Y. Suzuki, and S. Endo, Generalized quantum subspace expansion, Phys. Rev. Lett. **129**, 020502 (2022).