Markov-chain Monte Carlo method enhanced by a quantum alternating operator ansatz

Yuichiro Nakano,[1,*] Hideaki Hakoshima,[1,2,†] Kosuke Mitarai,[1,2,‡] and Keisuke Fujii[1,2,3,§]
[1]Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
[2]Center for Quantum Information and Quantum Biology, Osaka University, 560-0043, Japan
[3]Center for Quantum Computing, RIKEN, Wako Saitama 351-0198, Japan

(Received 28 November 2023; accepted 27 June 2024; published 25 July 2024)

Quantum computation is expected to accelerate certain computational tasks over classical counterparts. Its most primitive advantage is its ability to sample from classically intractable probability distributions. A promising approach to make use of this fact is the so-called quantum-enhanced Markov-chain Monte Carlo (qe-MCMC) method [D. Layden *et al.*, Nature (London) **619**, 282 (2023)], which uses outputs from quantum circuits as the proposal distributions. In this paper, we propose the use of a quantum alternating operator ansatz (QAOA) for qe-MCMC and provide a strategy to optimize its parameters to improve convergence speed while keeping its depth shallow. The proposed QAOA-type circuit is designed to satisfy the specific constraint which qe-MCMC requires with arbitrary parameters. Through our extensive numerical analysis, we find a correlation in a certain parameter range between an experimentally measurable value, acceptance rate of MCMC, and the spectral gap of the MCMC transition matrix, which determines the convergence speed. This allows us to optimize the parameter in the QAOA circuit and achieve quadratic speedup in convergence. Since MCMC is used in various areas such as statistical physics and machine learning, this paper represents an important step toward realizing practical quantum advantage with currently available quantum computers through qe-MCMC.

DOI: 10.1103/PhysRevResearch.6.033105

I. INTRODUCTION

The number of qubits in current quantum computers is limited. They hence lack the capacity to implement quantum error correction, rendering them vulnerable to noise. These emerging systems are known as noisy intermediate-scale quantum (NISQ) devices [1]. These devices have successfully demonstrated the superiority of quantum computers over classical computers in practice: *quantum supremacy* [2,3]. For practical applications, variational quantum algorithms (VQAs) [4] emerge as a promising approach to utilize NISQ devices. VQAs run parameterized quantum circuits (known as *variational quantum circuits*) on NISQ devices and optimize parameters using an objective function which is expressed by an expected value of an observable with respect to the output distribution. This approach keeps the quantum circuit depth shallow because the optimization is performed on classical computers. Some algorithms based on the VQA framework have been proposed for quantum chemical computation [5], combinatorial optimization [6], and machine learning [7,8].

Unfortunately, existing VQAs have not yet demonstrated a quantum advantage over the state-of-the-art classical approach for solving those problems. A possible weakness of these algorithms is their use of the expected values of operators. This requires us to run quantum circuits many times to suppress statistical errors. Furthermore, since optimization using the expected value is performed iteratively, the total runtime can be prohibitively large [9–11].

In contrast, algorithms that utilize each sampling output from quantum circuits may be more suitable for making use of NISQ devices. For example, random circuit sampling used to demonstrate quantum supremacy in NISQ devices is the task of sampling from the output distribution of a random quantum circuit [12] and is shown to be classically hard under a plausible conjecture [13]. Sampling from instantaneous quantum polynomial circuits [14] and random linear optical circuits [15] is another famous example whose classical hardness is strongly believed. These examples motivate us to develop algorithms that fully exploit each sampling outcome from NISQ devices.

The quantum-enhanced Markov-chain Monte Carlo (qe-MCMC) method [16] is one of such algorithms, which uses samples from a quantum circuit as the proposal distribution in the Metropolis-Hastings method [17]. The Markov-chain Monte Carlo (MCMC) method [17,18] is a powerful technique for sampling from computationally difficult distributions such as the Boltzmann distribution and has many applications in statistical physics [19], combinatorial optimization [20], and machine learning [21]. The Metropolis-Hastings method, one of the MCMC methods, consists of two steps: the generation of a sample by the proposal distribution and the accepting or rejecting step of this sample. Since the proposal distribution determines the efficiency of the algorithm, the proposal distribution using

[*] Contact author: u830977g@ecs.osaka-u.ac.jp
[†] Contact author: hakoshima.hideaki.es@osaka-u.ac.jp
[‡] Contact author: mitarai.kosuke.es@osaka-u.ac.jp
[§] Contact author: fujii.keisuke.es@osaka-u.ac.jp

a quantum computer, including those that are difficult to simulate on classical computers, can improve the convergence speed of MCMC over existing ones. The qe-MCMC method employs a distribution defined by a classically intractable quantum state as its proposal.

The circuit proposed in Ref. [16] is expressed as the time evolution governed by a time-independent Hamiltonian. However, when running it on quantum computers, the time evolution must be decomposed by the Suzuki-Trotter expansion [22], which increases the circuit depth when the evolution time is long.

In addition, the quantum circuits and their parameters are selected heuristically, and the strategy to construct a quantum circuit that improves the convergence speed of MCMC remains unclear.

In this paper, based on the qe-MCMC method, we propose a MCMC method called quantum alternating operator ansatz Monte Carlo (QAOA-MC). This algorithm uses a fixed-depth parameterized quantum circuit in the form of the so-called quantum alternating operator ansatz (QAOA) [23] as the proposal distribution. We thereby aim to suppress the increase in circuit depth regardless of the choice of parameters. Furthermore, we construct a systematic strategy to optimize the circuit to improve the convergence speed by examining the relationship between the absolute spectral gap and the acceptance rate (AR) of the proposal distribution. More precisely, we find that MCMC can be accelerated by minimizing AR after properly limiting the parameter range and reducing the number of circuit parameters. Through the numerical experiments, we evaluate the performance of QAOA-MC through the Boltzmann distribution for a spin-glass model, which is one of the most challenging systems to simulate due to its complex energy landscape and slow relaxation dynamics. As a result, we show that QAOA-MC achieves a near quadratic speedup in the convergence speed compared with the proposal using the uniform distribution. Additionally, we demonstrate QAOA-MC through the estimation of the average magnetization in a spin glass consisting of 15 spins. Our results suggest an acceleration of MCMC using NISQ devices and contribute to promoting the use of current NISQ devices.

The rest of this paper is organized as follows. First, we will explain MCMC in detail and introduce qe-MCMC in Sec. II. Furthermore, we discuss the challenges of qe-MCMC. Section III outlines our scheme: QAOA-MC. There, we propose utilizing a parameterized quantum circuit for MCMC proposals and provide guidance on optimizing the circuit. In Sec. IV, we describe the details of the numerical experiments and their results. Finally, a conclusion and future perspectives are presented in Sec. V.

II. PRELIMINARY

In this section, we provide an overview of MCMC and introduce qe-MCMC.

A. MCMC

The MCMC method is a very powerful algorithm that can sample according to an arbitrary probability distribution. This algorithm starts with a state $\mathbf{x} = [x_1, x_2, \dots, x_n]$ and changes the state according to a Markov chain, which is a stochastic process denoted by a transition probability $P(\mathbf{x}'|\mathbf{x})$. Especially an irreducible and aperiodic Markov chain has a unique stationary distribution [24]. This indicates that, after large enough transitions, the states $\mathbf{x}$ will converge to this stationary distribution, a feature exploited by the Markov chain for sampling tasks. The subsequent challenge involves designing a Markov chain, denoted by $P(\mathbf{x}'|\mathbf{x})$, so that its stationary distribution matches the desired target distribution $\pi(\mathbf{x})$. This can be achieved straightforwardly by fulfilling the detailed balance. The detailed balance is that, for any state transition from $\mathbf{x}$ to $\mathbf{x}'$, the following equation is satisfied:

$$\pi(\mathbf{x})P(\mathbf{x}'|\mathbf{x}) = \pi(\mathbf{x}')P(\mathbf{x}|\mathbf{x}') \quad \forall \mathbf{x}, \mathbf{x}'. \tag{1}$$

The Metropolis-Hastings method [17,18] realizes a transition that satisfies the detailed balance. In this method, a transition from $\mathbf{x}$ to $\mathbf{x}'$ with probability $P(\mathbf{x}'|\mathbf{x})$ is factored into a proposal distribution $Q(\mathbf{x}'|\mathbf{x})$ and an acceptance probability $A(\mathbf{x}'|\mathbf{x})$ for the proposal. The procedure is as follows: first, propose the next state according to $Q(\mathbf{x}'|\mathbf{x})$. Secondly, decide whether to accept the proposal based on the acceptance probability:

$$A(\mathbf{x}'|\mathbf{x}) = \min\left[1, \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}\frac{Q(\mathbf{x}|\mathbf{x}')}{Q(\mathbf{x}'|\mathbf{x})}\right]. \tag{2}$$

If the proposal is rejected, the next state remains the same as the state before the proposal. There are no restrictions on the choice of $Q(\mathbf{x}'|\mathbf{x})$, but the ratio $Q(\mathbf{x}|\mathbf{x}')/Q(\mathbf{x}'|\mathbf{x})$ must be in a form that can be calculated efficiently. However, if $Q(\mathbf{x}'|\mathbf{x}) = Q(\mathbf{x}|\mathbf{x}')$, then $Q(\mathbf{x}'|\mathbf{x})$ does not require explicit calculation, and the acceptance probability simplifies to $A(\mathbf{x}'|\mathbf{x}) = \min\{1, \pi(\mathbf{x}')/\pi(\mathbf{x})\}$. This approach is known as the Metropolis method [18].

The Boltzmann distribution of classical Ising models is one of the most representative distributions sampled using MCMC. The Boltzmann distribution describes the thermal equilibrium state of a system and is defined as

$$\mu(\mathbf{x}) = \frac{1}{Z}\exp[-\beta E(\mathbf{x})], \tag{3}$$

$$Z = \sum_{\mathbf{x}} \exp[-\beta E(\mathbf{x})], \tag{4}$$

where Z represents a partition function: the sum of the Boltzmann factor $\exp[-\beta E(\mathbf{x})]$ for all states $\mathbf{x}$, and $\beta = 1/k_B T$ is known as the inverse temperature, where T is the temperature of the system. In this paper, the Boltzmann constant is set to $k_B = 1$, and $E(\mathbf{x})$ is an energy function of the system. The energy function of the classical Ising model is

$$E(\mathbf{x}) = -\sum_{\langle j,k\rangle} J_{jk}x_j x_k - \sum_{j=1}^{n} h_j x_j, \tag{5}$$

where $x_j \in \{1, -1\}$ is a variable of the spin of the jth site.

In MCMC, the choice of proposal distributions is crucial, as it determines the convergence speed. The simplest method is to flip one spin in the configuration at random, which is called the *local update*. This method can be applied to any model and is easy to implement. However, it requires a large number of transitions to distant configurations at Hamming distance, increasing the probability of rejection in the process. [This is particularly true for $\mu(\mathbf{x})$ at low T.] This problem can be avoided by flipping multiple spins at once. This proposal is called the *global update*. One of the simplest global updates is to propose transitions with equal probability for all possible states. This proposal follows a uniform distribution, which we refer to as the *uniform update*. A more sophisticated global update is called the *cluster update*. In this update, we flip all spins in a group (a cluster), which is determined according to model-specific algorithms. The cluster update improves computational time for certain models [25,26]. However, generating clusters using this method is not straightforward, and it can only be applied to specific models.

Finally, we will now describe the convergence speed of MCMC. The convergence speed of a Markov chain can be represented by the eigenvalues of the transition matrix $\mathbf{P}$ [27,28]. The transition matrix $\mathbf{P}$ is defined by a $2^n \times 2^n$ matrix, in which each element is the transition probability $P(\mathbf{x}'|\mathbf{x})$. In this paper, we use a quantity called the absolute spectral gap as the metric to evaluate convergence speed following Ref. [16]. The *absolute spectral gap* [24] is defined as the absolute difference between the first two largest eigenvalues (λ_1 and λ_2, which satisfy $1 = \lambda_1 \geqslant \lambda_2$) of $\mathbf{P}$, which is represented by

$$\delta = 1 - |\lambda_2|, \tag{6}$$

where δ is in the range from 0 to 1. The larger the value of δ, the faster the convergence speed becomes.

B. qe-MCMC

The qe-MCMC algorithm, developed by Layden *et al.* [16], samples the Boltzmann distribution $\mu(\mathbf{x})$ for the classical Ising model using NISQ devices. The qe-MCMC algorithm uses a quantum circuit to sample a proposal distribution to realize sampling $\mu(\mathbf{x})$. More concretely, the proposal is executed by applying the quantum circuit U to $|\mathbf{x}\rangle$, which encodes $\mathbf{x}$ as a quantum state, and then measuring $U|\mathbf{x}\rangle$ to obtain $\mathbf{x}'$. The proposal distribution is therefore given by $Q(\mathbf{x}'|\mathbf{x}) = |\langle\mathbf{x}'|U|\mathbf{x}\rangle|^2$. At first sight, Eq. (2) seems to require us to compute the exact value of $Q(\mathbf{x}'|\mathbf{x})$. The computation of $|\langle\mathbf{x}'|U|\mathbf{x}\rangle|^2$ for a general quantum circuit U generally requires exponential time and cannot be performed efficiently even with quantum computers. However, we can avoid its computation by imposing the symmetry $U = U^\top$ on the quantum circuit, which leads to

$$Q(\mathbf{x}'|\mathbf{x}) = \left|\langle\mathbf{x}'|U|\mathbf{x}\rangle\right|^2 = \left|\langle\mathbf{x}|U|\mathbf{x}'\rangle\right|^2 = Q(\mathbf{x}|\mathbf{x}'). \tag{7}$$

This eliminates the Q term in Eq. (2), simplifying it to the Metropolis method. When the target distribution $\pi(\mathbf{x}) = \mu(\mathbf{x})$, Eq. (2) becomes $A(\mathbf{x}'|\mathbf{x}) = \min[1, \exp\{-\Delta E/T\}]$, where $\Delta E = E(\mathbf{x}') - E(\mathbf{x})$ is an energy difference between two configurations. It can be efficiently calculated on a classical computer so that the decision of accepting/rejecting the proposal is done on classical computers. The quantum circuit is only used for the proposal $\mathbf{x} \rightarrow \mathbf{x}'$. More importantly, while VQAs necessitate multiple runs and measurements of the quantum circuit to calculate the objective function and optimize the circuit, qe-MCMC requires only a single run and measurement of the quantum circuit for each MCMC step.

A variety of quantum circuits can be used in the qe-MCMC algorithm if $U = U^\top$ is satisfied. In Ref. [16], the time evolution is used under a time-independent Hamiltonian H as U, and the performance of this algorithm is evaluated. More concretely, their choice of U is given by

$$U = \exp(-iHt), \tag{8}$$

$$H = (1-u)\alpha H_{\text{prob}} + uH_{\text{mix}}, \tag{9}$$

where

$$\alpha = ||H_{\text{mix}}||_{\text{F}}/||H_{\text{prob}}||_{\text{F}} \tag{10}$$

is the normalization factor for H_{mix} [$\|A\|_f = \text{tr}(\text{A}^\dagger \text{A})^{1/2}$ is the Frobenius norm of a matrix A], and $u \in [0, 1]$ is a parameter that controls the relative weights of both H_{mix} and H_{prob}. The H_{mix} and H_{prob} are given by

$$H_{\text{mix}} = \sum_{j=1}^{n} X_j, \tag{11}$$

$$H_{\text{prob}} = -\sum_{\langle j,k\rangle} J_{jk} Z_j Z_k - \sum_{j=1}^{n} h_j Z_j, \tag{12}$$

where X_j and Z_j are the Pauli operators acting on the jth qubit, and H_{prob} is the target Hamiltonian from whose Boltzmann distribution we wish to sample. The coefficients $\{J_{jk}\}$ and $\{h_j\}$ are defined by couplings and external fields of the target Hamiltonian. The algorithm flow is shown in Algorithm 1.

ALGORITHM 1. Quantum-enhanced MCMC [16].

```
x = initial spin configuration
while not converged do
  Propose jump (quantum step)
  $u$ = random.uniform(0.25, 0.6)
  $t$ = random.uniform(2, 20)
  $|\psi\rangle = \exp[-iH(u)t]\,|\mathbf{x}\rangle$ on quantum device
  $\mathbf{x}'$ = result of measuring $|\psi\rangle$ in computational basis
  Accept/reject jump (classical step)
  $A = \min(1, \exp\{[E(\mathbf{x}) - E(\mathbf{x}')]/T\})$
  if $A \geqslant$ random.uniform(0, 1) then
    $\mathbf{x} = \mathbf{x}'$
  end if
end while
```

However, U in Eq. (8) must be implemented by the Suzuki-Trotter decomposition, which increases the circuit depth depending on the choice of the time parameter t. In addition, the parameters (u, t) are chosen randomly, and no optimization method has been established. In this paper, we aim to resolve these challenges.

III. QAOA-MC: MCMC WITH VARIATIONALLY TRAINED QUANTUM SAMPLING

In this section, we propose using a parameterized quantum circuit with a fixed depth as the proposal distribution of qe-MCMC and optimizing this circuit based on the MCMC AR. We call this method QAOA-MC. An overall view of this algorithm is shown in Fig. 1.

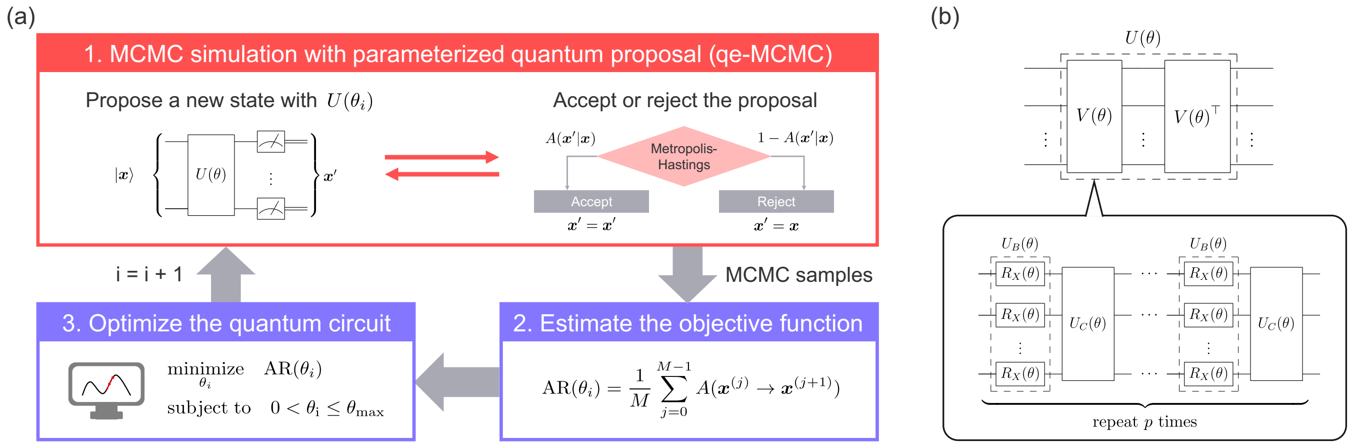

(a)
1. MCMC simulation with parameterized quantum proposal (qe-MCMC)
Propose a new state with $U(\theta_i)$
$|\boldsymbol{x}\rangle$
$U(\theta)$
$\boldsymbol{x}'$
Accept or reject the proposal
$A(\boldsymbol{x}'|\boldsymbol{x})$
Metropolis-Hastings
$1 - A(\boldsymbol{x}'|\boldsymbol{x})$
Accept
Reject
$\boldsymbol{x}' = \boldsymbol{x}'$
$\boldsymbol{x}' = \boldsymbol{x}$
i = i + 1
MCMC samples
3. Optimize the quantum circuit
$\underset{\theta_i}{\text{minimize}} \quad \text{AR}(\theta_i)$
subject to $0 < \theta_i \leq \theta_{\max}$
2. Estimate the objective function
$\text{AR}(\theta_i) = \frac{1}{M}\sum_{j=0}^{M-1} A(\boldsymbol{x}^{(j)} \to \boldsymbol{x}^{(j+1)})$
(b)
$U(\theta)$
$V(\theta)$
$V(\theta)^{\top}$
$U_B(\theta)$
$R_X(\theta)$
$U_C(\theta)$
repeat p times

FIG. 1. An overview of the quantum alternating operator ansatz (QAOA)-Monte Carlo (MC) algorithm. (a) Schematic illustration of the QAOA-MC algorithm. (b) Parameterized quantum circuit used in the Markov-chain MC (MCMC) proposal. A structure of the circuit is inspired by the QAOA but is designed to satisfy the symmetry constraint which quantum-enhanced MCMC requires regarding arbitrary parameters, which is slightly different from the original QAOA.

A. Parameterized quantum circuit

We apply QAOA [23] to the structure of the circuit that generates MCMC proposals. Concretely, this circuit is defined as follows:

$$U = V(\boldsymbol{\beta}, \boldsymbol{\gamma})^{\top} V(\boldsymbol{\beta}, \boldsymbol{\gamma}), \tag{13}$$

where,

$$V(\boldsymbol{\beta}, \boldsymbol{\gamma}) = U_C(\gamma_p)U_B(\beta_p)\cdots U_C(\gamma_1)U_B(\beta_1), \tag{14}$$

$$U_B(\beta) = \exp(-iH_{\text{mix}}\beta), \quad U_C(\gamma) = \exp(-i\alpha H_{\text{prob}}\gamma), \tag{15}$$

and p is a hyperparameter that determines the depth of the circuit. It is shown in Fig. 1(b). The circuit has $2p$ parameters: $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_p\}$ and $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_p\}$. Here, α, H_{mix}, and H_{prob} are as defined in Eqs. (10)–(12), respectively. Note that the circuit defined by Eq. (13) always satisfies $U = U^{\top}$ by construction.

The quantum circuit U in Eq. (13) has a similar structure to Eq. (8). However, unlike the circuit implementation in Eq. (8), this circuit is more NISQ friendly because the circuit depth is fixed. The initial state $|\mathbf{x}\rangle$ is expected to be updated globally through H_{mix}, and H_{prob} is responsible for proposing a transition respecting the energy landscape of the system. The generated probability distribution $Q(\mathbf{x}'|\mathbf{x})$ includes those that are classically difficult to simulate and may realize acceleration of the convergence compared with existing proposal distributions.

B. Optimization of circuit

Next, we explain how to optimize the proposal distribution generated by the proposed circuit [Eq. (13)] to achieve faster convergence. One might think that we can use the absolute spectral gap δ [Eq. (6)] which directly determines the convergence speed as an objective function to maximize. However, computing δ requires solving for the eigenvalues of a transition probability matrix $\mathbf{P}$ with a size of $2^n \times 2^n$ for a system of size n and is not feasible. The objective function must be a quantity that reflects the convergence speed of MCMC and is easily computable. We find that the MCMC AR can be used as the objective function after some numerical experiments. The AR [29] is defined as

$$\text{AR} = \sum_{\mathbf{x},\mathbf{x}'} \pi(\mathbf{x})Q(\mathbf{x}'|\mathbf{x})A(\mathbf{x}'|\mathbf{x}). \tag{16}$$

This formula includes $\pi(\mathbf{x})$, making it difficult to calculate directly. However, it can efficiently be estimated by performing MCMC on $\pi(\mathbf{x})$. We estimate AR using samples generated by M samples of MCMC as follows:

$$\text{AR} \approx \frac{1}{M}\sum_{j=0}^{M-1} A[\mathbf{x}^{(j+1)}|\mathbf{x}^{(j)}], \tag{17}$$

where $\mathbf{x}^{(j)}$ represents the state at the jth step of the MCMC.

After experimenting with the Boltzmann distributions for various Ising models, we have discovered a relationship between AR and the absolute spectral gap δ. Figure 2 illustrates this relationship in a typical Ising model instance. In this experiment, we use a single-parameter circuit $U(\theta)$ which is defined by setting the parameters in Eq. (13) as

$$\theta = \beta_1 = \cdots = \beta_p = \gamma_1 = \cdots = \gamma_p.$$

Figure 2 shows that, although there is usually no correlation between AR and δ, a correlation exists for small θ, where δ increases as AR decreases. It continues until the AR reaches a local minimum, which is often a local maximum value of δ. Based on these observations, we optimize $U(\theta)$ by searching for a small θ that achieves the locally minimal AR.

There are several reasons for AR minimization in QAOA-MC optimization. Firstly, the optimization process initiates near $\theta = 0$. At this point, for the Metropolis-Hastings method, the proposal distribution is maximized with AR = 1, yet it corresponds to the delta-function proposal which proposes the same state as before, rendering it the most inefficient proposal [30]. Consequently, in most instances, directing the search toward minimizing AR tends to significantly enhance convergence speed. Moreover, the practically desirable AR values are generally considered to fall between

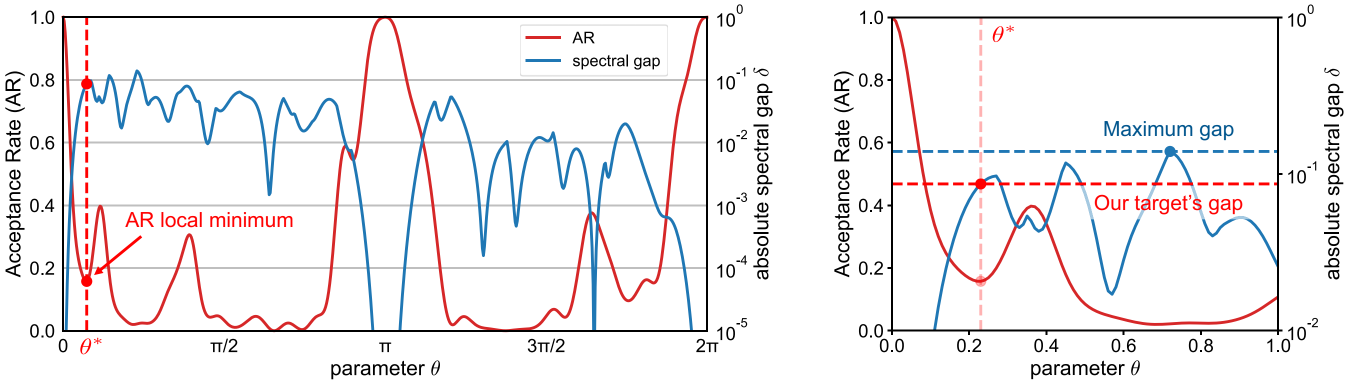


FIG. 2. The relationship between the circuit parameter θ, acceptance rate (AR), and the absolute spectral gap δ in a typical instance. The right figure is an enlarged image of the area around the smallest parameter θ^* among those taking AR minima in the left figure. In many cases, θ^* gives a large absolute spectral gap. In these figures, we use a $n = 5$ fully connected Ising model [described by Eq. (5)]. $\{J_{jk}\}$ and $\{h_j\}$ are randomly generated from a standard normal distribution.

0.1 and 0.6 [31]. Given these observations, it is reasonable to optimize the system with the aim of moderately reducing the AR value. The reason AR local minima near the origin lead to effective global convergence speed remains an open question. While we expect that these considerations will remain valid when multiple parameters are involved, such an optimization becomes increasingly challenging as the dimensionality of the parameter space grows. Therefore, in this paper, to maintain simplicity, the single parametrization is employed. It is important to note that our observation is given for the classical Ising model; its applicability to other models remains open.

Finally, the overview of QAOA-MC is shown in Fig. 1(a). We perform a search for the local minimum value of AR by restricting the search range to $\theta \in (0, \theta_{\max}]$. Here, $\theta_{\max}$ is set as a hyperparameter in our optimization method. We find that a fixed $\theta_{\max}$ can be used for different model instances without deteriorating the performance if we fix the depth parameter p (see Appendix).

IV. NUMERICAL EXPERIMENTS

A. Average convergence speed

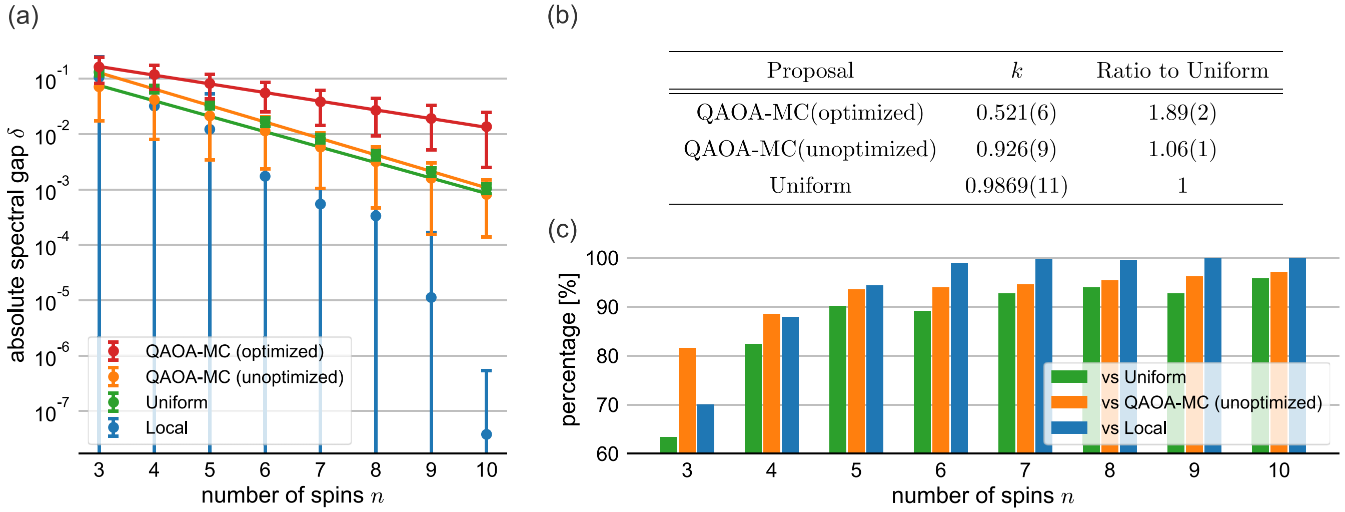
First, to investigate the performance of QAOA-MC, we analyze the absolute spectral gap δ of the Boltzmann distribution $\mu(\mathbf{x})$ for fully connected Ising model instances of various sizes n. The temperature of the Boltzmann distribution is set to $T = 0.1$. We prepare 500 random spin-glass instances by randomly choosing $\{J_{jk}\}$ and $\{h_j\}$ from a standard normal distribution and calculate δ for each $\mu(\mathbf{x})$. The average convergence speed $\langle\delta\rangle$ for a model size of n is obtained from these 500 δ values. This is done for each $3 \leqslant n \leqslant 10$ to investigate the relationship between n and $\langle\delta\rangle$. We use the circuit in Eq. (13) [Fig. 1(b)] with $p = 5$ and set the hyperparameter $\theta_{\max} = 0.3$ (see Appendix). In this numerical experiment, we compare the QAOA-MC proposal to three proposal distributions: local update, uniform update, and random circuit. This *random circuit* corresponds to a distribution defined by Eq. (13) with a randomly chosen parameter $\theta \in [0, 2\pi]$ to verify the improvement of convergence speed through optimization. This numerical experiment is simulated entirely on a classical computer using Python. Qulacs [32] is utilized to simulate the quantum circuit. The optimization method used is L-BFGS-B [33], which is implemented by SciPy [34]. AR is calculated exactly by Eq. (16).

Figure 3(a) shows the relationship between n and $\langle\delta\rangle$ obtained from the numerical experiment. The points represent $\langle\delta\rangle$ computed using 500 random instances at each value of n. The error bars represent the standard deviations computed over 500 δ values. Although $\langle\delta\rangle$ decreases as n increases for all methods, QAOA-MC shows a slower rate of decrease than others and is superior in terms of $\langle\delta\rangle$. We fit $\langle\delta\rangle$ by 2^{-kn} with a parameter k and show the result as the straight lines in Fig. 3(a). The approximation curves fit the data well except for the local update. The fitting is calculated using the least squares method. Figure 3(b) displays the scaling factor k for these curves. Uncertainties in Fig. 3(b) are from the covariance matrices obtained in the fitting process. QAOA-MC has a scaling factor $k \sim \frac{1}{1.89}$ times that of the uniform update, which represents an approximately quadratic acceleration concerning $\langle\delta\rangle$. On the other hand, the results of the random circuit are almost identical to those of the uniform update, suggesting that this acceleration is due to the optimization of the circuit.

Although QAOA-MC optimizes a parameter based on the observation that a local minimum of AR often gives a local maximum of δ, it does not always hold for all instances. To see the effect of this imperfect assumption, we next show the percentage of instances for which QAOA-MC surpasses δ of other methods for each size n in Fig. 3(c). The percentage of instances in which QAOA-MC is dominant increases with increasing n. For $n \geqslant 7$, QAOA-MC outperforms the others in >90% of the 500 instances, making it the best-performing proposal in this experiment. This result indicates that our optimization method, which searches for a local minimum of AR for small values of θ, works for many instances.

B. Optimization with MCMC estimator of AR

We now examine the impact of MCMC estimation of AR on the performance of QAOA-MC. Since QAOA-MC must use the MCMC estimate for obtaining AR in practice, the objective function contains statistical errors that could adversely affect the convergence performance. We analyze the



Proposal	k	Ratio to Uniform
QAOA-MC(optimized)	0.521(6)	1.89(2)
QAOA-MC(unoptimized)	0.926(9)	1.06(1)
Uniform	0.9869(11)	1

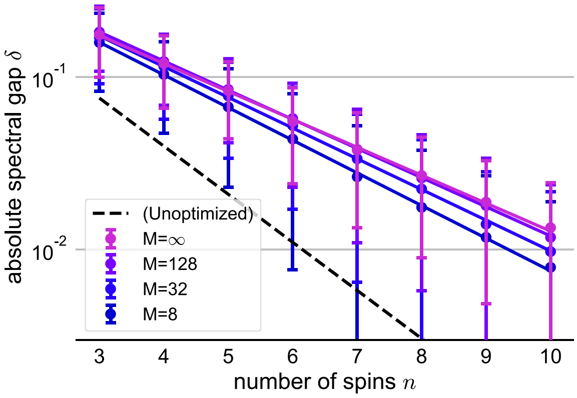
FIG. 3. Numerical simulation results for the absolute spectral gap δ of the Boltzmann distribution at $T = 0.1$ for fully connected Ising model instances. (a) Relationship between model size n and average convergence rate $\langle\delta\rangle$. "QAOA-MC (unoptimized)," where QAOA-MC stands for quantum alternating operator ansatz Monte Carlo, uses the randomly chosen parameter $\theta \in [0, 2\pi]$ in the parameterized quantum circuit [Fig. 1(b)]. (b) The value of the scaling factor k obtained by fitting $\langle\delta\rangle$ with 2^{-kn}. (c) Percentage of cases where QAOA-MC surpasses other methods in convergence speed δ.

relationship between the number of samples used in AR estimation and the performance. The numerical experiments performed here are under the same setup as discussed in Sec. IV A unless otherwise stated. AR is estimated from M samples, which are obtained through MCMC as described by Eq. (17). We set M to 8, 32, 128, and ∞ [where AR is calculated directly from the target distribution via Eq. (16)] and optimize θ. Note that, in the AR estimation, the MCMC simulations start from a random initial configuration, and no postprocessing techniques, such as burn-in, are employed. Then using the optimized θ, we calculate the absolute spectral gap δ for the same instances used in Sec. IV A. When using MCMC estimators, the L-BFGS-B method cannot be used as the optimization method because the objective function contains statistical errors. Here, we employ the bisection method for optimization, taking advantage of the fact that we only have a single parameter θ. In the numerical experiments, we used Brent's method [35], which is implemented by SciPy. The circuit and the hyperparameter settings are the same as in Sec. IV A.

Figure 4 displays the relationship between M and the resulting $\langle\delta\rangle$. Figure 4 shows the scaling factor k for the approximate curves obtained by the same fitting as Fig. 3. As M becomes smaller, the standard deviation of $\langle\delta\rangle$ increases, and the scaling factor k deteriorates at the same time. This is because decreasing M results in a less accurate AR estimate. It then leads to poor optimization, the result approaching random outcomes. On the other hand, if M is large enough, Brent's method can be used to achieve a performance that is nearly the same as that attained by the L-BFGS-B method. Note that the size of a sufficient M in QAOA-MC is much less than the number of measurements used for a single evaluation of an expected value in VQAs. Additionally, once the parameters are optimized, only a single-shot measurement from the optimized circuit is required for each step of the MCMC.

C. Magnetization estimation

Finally, we conduct numerical experiments of QAOA-MC by estimating a physical quantity, the average magnetization



M	k	Ratio to Uniform
∞	0.538(6)	1.834(19)
128	0.5615(26)	1.758(9)
32	0.592(6)	1.668(17)
8	0.629(6)	1.568(15)

FIG. 4. (Upper) Relationship between model size n and average convergence rate $\langle\delta\rangle$ toward the number of Markov-chain Monte Carlo (MCMC) samples M using each estimation of the acceptance rate (AR). The dotted line represents the "QAOA-MC (unoptimized)," where QAOA-MC stands for quantum alternating operator ansatz Monte Carlo, result in Fig. 3(a). (Lower) The value of the scaling factor k of the approximate curve $\langle\delta\rangle \approx 2^{-kn}$ in QAOA-MC, as M varies.

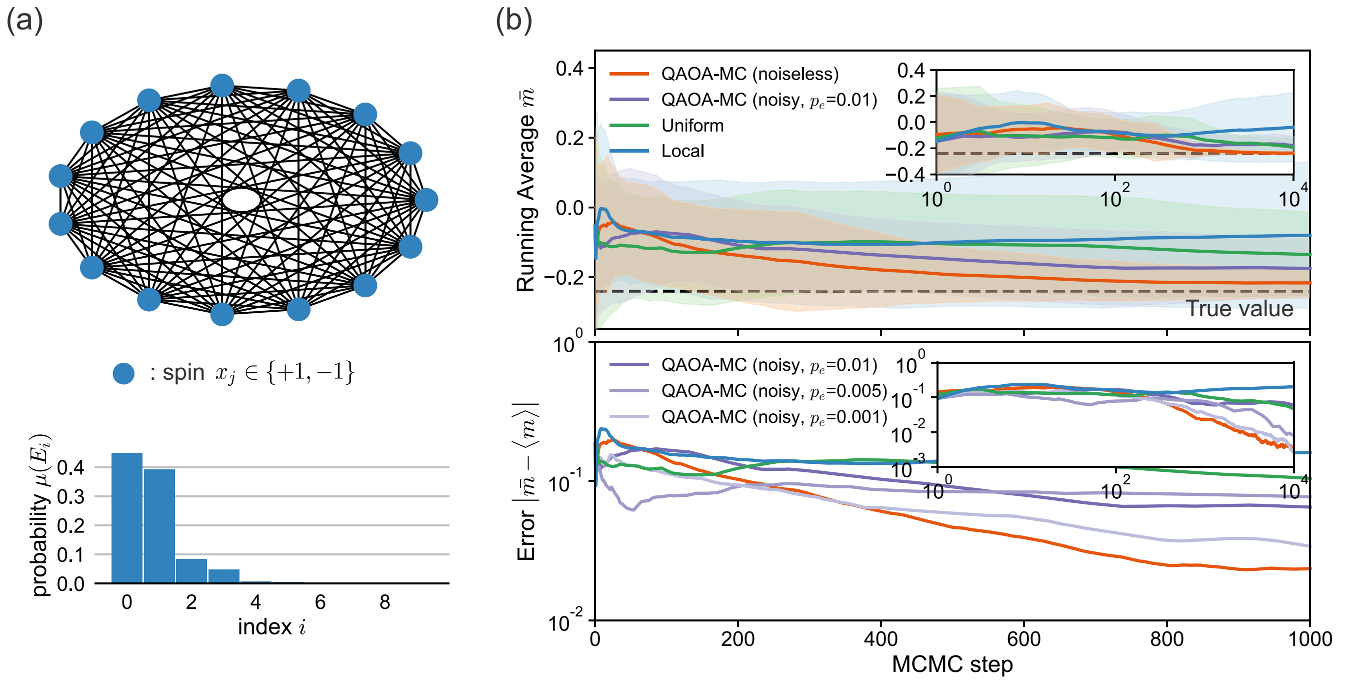


FIG. 5. Estimation of the average magnetization $\langle m\rangle$ using the Markov-chain Monte Carlo (MCMC) method. (a) (Top) A graph of a $n = 15$ fully connected instance used in this experiment. (Bottom) Histogram of the Boltzmann distribution for the instance. The horizontal axis is sorted in ascending order of energy E_i ($E_0 \leqslant E_1 \leqslant \cdots \leqslant E_{2^n-1}$). (b) (Top) The relationship between the number of MCMC steps and the estimated value $\bar{m}$. (Bottom) The absolute difference between the mean of the estimated value $\bar{m}$ and the true value $\langle m\rangle$.

$\langle m\rangle$ of the Ising model. The average magnetization, defined in the context of the Boltzmann distribution $\mu(\mathbf{x})$, is expressed as

$$\langle m\rangle = \sum_{\mathbf{x}} \mu(\mathbf{x}) m(\mathbf{x}), \tag{18}$$

where $m(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} x_j$ denotes the magnetization of a state $\mathbf{x}$. In this numerical experiment, we estimate the average magnetization of an $n = 15$ fully connected spin-glass instance [Fig. 5(a)] with respect to the Boltzmann distribution at $T = 1.0$. As shown in Fig. 5(a), the Boltzmann probabilities for the ground, first, second, and third excited states (E_0, E_1, E_2, and E_3) are ∼45.0, 39.3, 8.6, and 4.8%, respectively, resulting in a multimodal distribution with large probabilities corresponding to several low-energy configurations. Notably, this instance features several energy minima, each separated by considerable Hamming distances. As a result, MCMC for such an instance faces significant challenges in transitioning frequently between these distant energy minima, a task that proves difficult for traditional classical proposal distributions.

We simulate not only an ideal quantum computer but also examine the effects of noise on the convergence performance of MCMC. Various implementation methods for quantum computers exist, and noise models can vary significantly among them. For our simulations, for simplicity, we employ one of the most commonly employed noise model, depolarizing error, and observe how noise impacts the MCMC convergence performance. For instance, a single-qubit gate is followed by a single-qubit depolarizing error:

$$\mathcal{D}(\rho) = (1 - p_e)\rho + \frac{p_e}{3}(X\rho X + Y\rho Y + Z\rho Z). \tag{19}$$

Here, p_e denotes the gate error probability, with Pauli operators X, Y, and Z acting with equal probability. The two-qubit gate error is also similarly introduced by using the two-qubit depolarizing noise with error probability p_e. While actual devices encounter various other types of noise, in this paper, we simplify by not delving into additional details. In this experiment, we set the error probability to $p_e = 1.0 \times 10^{-2}$, 5.0×10^{-3}, and 1.0×10^{-3}, which is feasible with current quantum computers.

In the experiment, optimization is conducted as previously described, utilizing AR estimated from MCMC samples. We employ Brent's method for this optimization. The number of samples used for AR estimation is set at $M = 1000$. All optimization processes are computed through noise-free simulations, while noise simulations are conducted on the MCMC using the optimized parameter θ^*. The MCMC simulations begin from a spin configuration randomly selected according to a uniform distribution, and each of the 10 simulations runs for 10 000 steps, with each simulation starting from a new initial configuration.

Figure 5(b) presents the MCMC estimation results for the average magnetization $\langle m\rangle$. The solid lines depict the average of running averages $\bar{m}$, derived from 10 independent Markov chains at each step. The shaded bands around these lines indicate their standard deviations, illustrating the variability among the chains. The dotted line in Fig. 5(b) represents the true value of $\langle m\rangle$, which has been calculated from the target distribution $\mu(\mathbf{x})$. It is clear from the QAOA-MC results that the running average $\bar{m}$ converges more rapidly to the true value than other methods, under both noisy and noise-free conditions. Additionally, the standard deviation of $\bar{m}$ within this algorithm remains low and stable throughout the simulations. In the presence of noise, observations indicate that the convergence speed decreases with the increasing error probability p_e. Simultaneously, it is also observed that, when p_e is sufficiently small, the advantage over classical proposals, such as the uniform proposal, remains. In practical devices, there may be additional factors not accounted for in this experiment that could adversely affect the performance of the algorithm. Notably, the presence of biased noise could disrupt the symmetry of the circuit. Nevertheless, this issue can be mitigated through the use of a noise-averaging technique known as Pauli twirling [16,36–38]. While we do not consider such factors in this experiment, and the implementation of twirling could potentially further impact performance, it is anticipated that this method would still substantially improve upon traditional classical proposal distributions.

V. CONCLUSIONS

In this paper, we proposed QAOA-MC, which uses samples from quantum circuits in the form of QAOA as the proposal for MCMC. Quantum computation is used only for proposing transitions, and the other parts of the algorithm are executed on classical computers, which makes the algorithm feasible on current NISQ devices. We introduced the use of a QAOA-type circuit to realize the algorithm with shallow circuits. Furthermore, we showed that the convergence speed of MCMC can be improved by finding a local minimum of the AR. As shown in numerical experiments, QAOA-MC confirmed an approximately quadratic speedup in the absolute spectral gap for the Boltzmann distribution in spin glass, when compared with the uniform distribution.

Some future directions are in order. First, the circuit in Fig. 1(b) has multiple parameters that could be further tuned to achieve a better proposal distribution. However, the optimization of multiple parameters using AR did not work in our trials. Building more advanced optimization methods remains for possible future work. Additionally, the results of the numerical experiments in this paper are based on the assumption of a quantum computer operating under an ideal or simplified noise model. It is unclear whether the acceleration can be achieved on real NISQ devices. If this advantage can be maintained despite the noise, QAOA-MC has the potential to become a practical algorithm for current quantum computers. This algorithm leaves much room for improvement; in any case, it represents a step toward implementing MCMC with quantum computers and facilitates the use of current NISQ devices.

ACKNOWLEDGMENTS

This paper is supported by MEXT Quantum Leap Flagship Program Grants No. JPMXS0118067394 and No. JPMXS0120319794, and JST COI-NEXT Grant No. JPMJPF2014. K.M. is supported by JST PRESTO Grant No. JPMJPR2019.

APPENDIX: THE CHOICE OF HYPERPARAMETER $\theta_{\max}$

In this section, we analyze the distribution of parameters θ^* that give the locally minimal AR for various instances to determine the hyperparameter $\theta_{\max}$. We define θ^* as θ that achieves the local minimum of AR, satisfies $\theta^* > 0$, and is closest to 0. In this paper, we use the average of θ^* in various instances as a hyperparameter $\theta_{\max}$.

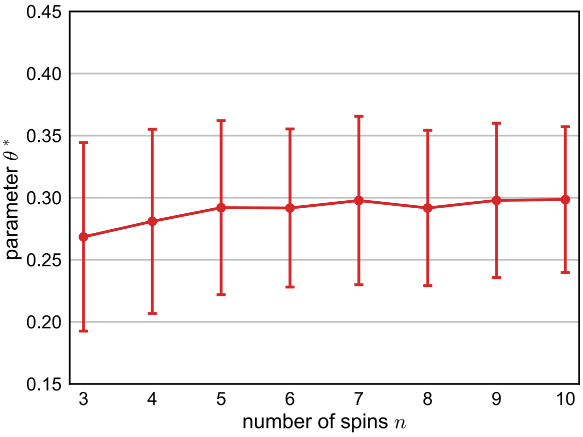


FIG. 6. Relationship between θ^* and n calculated by 500 random instances.

1. Analyzing θ^* vs instance

Here, θ^* varies with specific model instances. To examine the distribution of θ^* among various instances, we generate 500 instances [Eq. (5)] with random $\{J_{jk}\}$ and $\{h_j\}$ for each $3 \leqslant n \leqslant 10$ and determine θ^* for each of them. In this numerical experiment, we use the circuit of Fig. 1(b) with $p = 5$. The target distribution is the Boltzmann distribution with $T = 0.1$ [Eq. (4)]. The result is shown in Fig. 6. The dots denote the average $\langle\theta^*\rangle$ of the 500 instances for each n, while the bands represent the corresponding standard deviations. It can be seen that the average of $\langle\theta^*\rangle$ remains ~ 0.3 (indicated by a dotted line in Fig. 6) irrespective of n. We, therefore, set $\theta_{\max} = 0.3$ in the numerical experiments presented in the main text.

2. Analyzing θ^* vs p

We investigate the relationship between p and θ^*. From Appendix 1, we see that $\theta_{\max}$ can be set at the same value, regardless of model instances. However, this is not the case

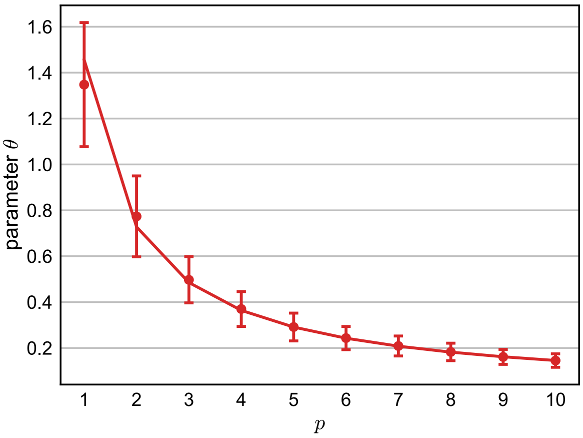


FIG. 7. Relationship between θ^* and p for each of the 50 random instances.

when p varies. We prepare 50 random instances for the spin glass [Eq. (5)] for $n = 5$, vary the p from 1 to 10, and calculate θ^*. The results are shown in Fig. 7. The dots denote the average θ^* of the 50 instances for each p, while the bands represent the corresponding standard deviations. Here, $\langle\theta^*\rangle$ is approximately proportional to $1/p$; the curved line in Fig. 7 represents a/p with $a = 1.455\,58(25)$ which is obtained by using the least squares method. When varying p, it seems appropriate to select $\theta_{\max}$ according to the fitting curve displayed in Fig. 7.

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

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

[3] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins *et al.*, Quantum computational advantage with a programmable photonic processor, Nature (London) **606**, 75 (2022).

[4] 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).

[5] 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).

[6] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028 (2014).

[7] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[8] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002 (2018).

[9] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, Quantum **4**, 263 (2020).

[10] J. F. Gonthier, M. D. Radin, C. Buda, E. J. Doskocil, C. M. Abuan, and J. Romero, Measurements as a roadblock to near-term practical quantum advantage in chemistry: Resource analysis, Phys. Rev. Res. **4**, 033154 (2022).

[11] K. Ito, Latency-aware adaptive shot allocation for run-time efficient variational quantum algorithms, arXiv:2302.04422 (2023).

[12] D. Hangleiter and J. Eisert, Computational advantage of quantum random sampling, Rev. Mod. Phys. **95**, 035001 (2023).

[13] S. Aaronson and L. Chen, Complexity-theoretic foundations of quantum supremacy experiments, in *Proceedings of the 32nd Computational Complexity Conference, Riga, Latvia* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017), Article No. 22.

[14] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Achieving quantum supremacy with sparse and noisy commuting quantum computations, Quantum **1**, 8 (2017).

[15] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2011), pp. 333–342.

[16] D. Layden, G. Mazzola, R. V. Mishmash, M. Motta, P. Wocjan, J.-S. Kim, and S. Sheldon, Quantum-enhanced Markov chain Monte Carlo, Nature (London) **619**, 282 (2023).

[17] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97 (1970).

[18] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[19] D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2021).

[20] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[21] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, An introduction to MCMC for machine learning, Mach. Learn. **50**, 5 (2003).

[22] M. Suzuki, Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Commun. Math. Phys. **51**, 183 (1976).

[23] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[24] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*, 2nd ed. (American Mathematical Society, Providence, 2017).

[25] R. H. Swendsen and J.-S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett. **58**, 86 (1987).

[26] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett. **62**, 361 (1989).

[27] P. H. Peskun, Optimum Monte-Carlo sampling using Markov chains, Biometrika **60**, 607 (1973).

[28] A. Frigessi, C.-R. Hwang, and L. Younes, Optimal spectral structure of reversible stochastic matrices, Monte Carlo methods and the simulation of Markov random fields, Ann. Appl. Probab. **2**, 610 (1992).

[29] A. Gelman, W. R. Gilks, and G. O. Roberts, Weak convergence and optimal scaling of random walk metropolis algorithms, Ann. Appl. Probab. **7**, 110 (1997).

[30] K. Neklyudov, E. Egorov, P. Shvechikov, and D. Vetrov, Metropolis-Hastings view on variational inference and adversarial training, arXiv:1810.07151 (2018).

[31] J. S. Rosenthal, Optimal proposal distributions and adaptive MCMC, in *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (CRC Press, New York, 2011), Chap. 4.

[32] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya *et al.*, Qulacs: A fast and versatile quantum circuit simulator for research purpose, Quantum **5**, 559 (2021).

[33] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. **16**, 1190 (1995).

[34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods **17**, 261 (2020).

[35] R. P. Brent, *Algorithms for Minimization Without Derivatives* (Dover Publications, Inc., Mineola, 2013).

[36] W. Dür, M. Hein, J. I. Cirac, and H.-J. Briegel, Standard forms of noisy quantum operations via depolarization, Phys. Rev. A **72**, 052326 (2005).

[37] E. Magesan, J. M. Gambetta, and J. Emerson, Scalable and robust randomized benchmarking of quantum processes, Phys. Rev. Lett. **106**, 180504 (2011).

[38] M. R. Geller and Z. Zhou, Efficient error models for fault-tolerant architectures and the Pauli twirling approximation, Phys. Rev. A **88**, 012314 (2013).