Accelerated Quantum Monte Carlo with Mitigated Error on Noisy Quantum Computer

Yongdan Yang⬤, Bing-Nan Lu⬤, and Ying Li⬤*

Graduate School of China Academy of Engineering Physics, Beijing 100193, China

Quantum Monte Carlo and quantum simulation are both important tools for understanding quantum many-body systems. As a classical algorithm, quantum Monte Carlo suffers from the sign problem, preventing its application to most fermion systems and real-time dynamics. In this paper, we introduce a novel nonvariational algorithm using quantum simulation as a subroutine to accelerate quantum Monte Carlo by easing the sign problem. The quantum subroutine can be implemented with shallow circuits and, by incorporating error mitigation, can reduce the Monte Carlo variance by several orders of magnitude even when the circuit noise is significant. As such, the proposed quantum algorithm is applicable to near-term noisy quantum hardware.

I. INTRODUCTION

The simulation of quantum many-body systems is one of the main motivations for quantum computing [1]. A lot of quantum many-body problems are intractable in classical computing. An apparent reason is that the Hilbert-space dimension increases exponentially with the system size and it is impossible to store the wave function of a large system in classical memory. Quantum Monte Carlo (QMC) is a group of classical algorithms designed to bypass this memory issue. By sampling only the most important part of the configuration space, QMC can solve certain many-body problems at a polynomial complexity, at the cost of introducing small statistical errors. Unfortunately, when applied to fermion systems and real-time dynamics, QMC encounters the notorious sign problem, i.e., the target amplitude is a highly oscillating function with alternating sign. This sign problem results in a variance that increases exponentially in the Monte Carlo simulation [2], forming the dominant limitation of QMC. On the other hand, by mapping the target wave function of the simulated system into the wave function of qubits on a fault-tolerant quantum computer [3], we can reproduce the dynamics of quantum systems while the memory and run time scale polynomially [4]. With the development of the fault-tolerant technologies as a long-term goal, exploring

the power of noisy intermediate-scale quantum hardware is of particular importance for near-term applications [5]. In this paper, we establish the framework of quantum-circuit Monte Carlo (QCMC) algorithm, in which quantum computing is a subroutine of QMC. We show that this algorithm has a quantum advantage in solving many-body problems, even on noisy quantum computers.

Since Ulam and Metropolis's pioneering work of using random sampling to simulate real physical systems [6], the Monte Carlo method has grown into a large family of algorithms. Here, we focus on a specific subset of Monte Carlo algorithms, namely, the QMC methods, which are based on real- or imaginary-time evolution. These methods include Green's function Monte Carlo [7], auxiliary field Monte Carlo [8,9], world-line Monte Carlo [10,11], and diagrammatic Monte Carlo [12–15], and their various variants. In what follows, by QMC, we refer to this subset of algorithms. The other QMC algorithms are based on variational methods [16] but while their connection to quantum computing is also an interesting topic, they are not be covered in this work.

In most QMC methods, we sample the configurations according to a quasiprobability amplitude derived from time evolution. For fermion systems, such an amplitude is usually a complex number, which can be positive definite if the system respects certain symmetries. Examples of the latter case include the half-filled Hubbard model with particle-hole exchange symmetry [17,18] and the nuclear system with Wigner-SU(4) symmetry [19,20]. However, a realistic Hamiltonian usually contains terms that break these symmetries and induce oscillating phases in the probability amplitude. As a result, even though QMC methods are very successful in describing certain strongly

*yli@gscaep.ac.cn

correlated systems in chemistry [21], condensed-matter physics [22], and nuclear physics [7], their application is still rather limited due to the sign problem. Although in some important cases the sign problem can be alleviated using complicated techniques [23], e.g., the complex Langevin method [24,25] or the Lefschetz thimble method [26,27], finding a generic solution is unlikely, as it is proven that the sign problem is NP-hard [2].

In quantum computing, the qubit and time costs for simulating the unitary time evolution of a quantum system scale polynomially with the problem parameters, i.e., the system size, evolution time, and accuracy. Such algorithms include the Lie-Trotter-Suzuki decomposition [4,28,29], the truncated Taylor series [30,31], linear combinations of Lie-Trotter-Suzuki products [32,33], and the random compiler [34]. Based on the simulation of unitary time evolution, one can also simulate open-system dynamics [35,36], solve equilibrium-state problems [37,38], and find the ground state for certain Hamiltonians [39–41]. However, implementation of these algorithms at a meaningful scale usually requires a fault-tolerant quantum computer [42,43], on which the logical error rate can be reduced to any level at a polynomial cost in quantum error correction [44]. In recent years, hybrid quantum-classical algorithms have been developed for applications before the era of fault-tolerant technologies [45]. Many such algorithms are based on variational principles for solving the ground-state energy [46,47], real-time simulation [48,49], and imaginary-time simulation [50,51]. A variational quantum algorithm largely depends on the ansatz, i.e., a parametrized quantum circuit. Some ansatz circuits suffer from the "barren-plateaus" problem, which is a vanishing gradient in the parameter landscape, making the algorithm inefficient [52]. So far, a general way to construct a proper ansatz is still lacking. Applied to Hamiltonians with tens to hundreds of qubits, the performance of variational quantum algorithms on a noisy quantum computer remains an open question [53,54].

In this paper, we propose a hybrid nonvariational quantum simulation algorithm, i.e., the QCMC algorithm. Contrary to the QMC methods, there is no sign problem in simulating the time evolution using quantum computing. If we can delegate the calculation of the most oscillating part to quantum computing, the remaining calculations in QMC might have a very mild sign problem, or even be free from it when the entire calculation is delegated to quantum computing. To explore this possibility, we carry out the QCMC simulation by sampling random quantum circuits. Several aspects of this hybrid scheme are discussed, including implementation of the time-evolution operators, the total computational complexity, the optimal sampling distribution in Monte Carlo, and the error-mitigation techniques. We show that our algorithm is polynomial on a fault-tolerant quantum computer and can reduce the variance of the Monte Carlo estimator even

on a noisy quantum computer. As a subroutine of QMC, the circuit depth in quantum computing can be drastically reduced compared with the conventional Lie-Trotter-Suzuki decomposition. Therefore, our algorithm is a suitable candidate for the near-term application of quantum computing.

In the QCMC algorithm, we simulate many-body dynamics by expressing the time-evolution operator in a summation form. Each term in the summation corresponds to a quantum-circuit configuration. The summation formula is chosen to minimize the circuit depth and variance of the Monte Carlo estimator. We introduce two series of summation formulas based on Lie-Trotter-Suzuki product formulas [55,56]: Pauli-operator-expansion (POE) formulas and leading-order-rotation (LOR) formulas. Compared with product formulas, in our formulas the algorithmic error converges faster with the time-step size Δt, at the cost of a moderately increased gate number per time step. For example, the second-order LOR formula converges as $O(\Delta t^6)$, which is even faster than the fourth-order product formula. This algorithmic error in QCMC is only due to the variance of the Monte Carlo estimator and can be reduced by increasing the sample number.

We mitigate errors in QCMC in three ways. First, our summation formulas are exact formulas of the time-evolution operator for any finite time-step size. The product formulas have the decomposition error depending on Δt, which must be sufficiently small to reduce the error. Exact summation formulas allow us to take a large Δt (i.e., a small number of time steps) and use shallow circuits to implement QCMC. We remark that the gate number per time step is only moderately increased to implement the proposed summation formulas. Second, we use quantum error-mitigation techniques to eliminate the impact of machine errors caused by noise in the quantum computer [48,57,58]. We present two types of circuits: forward-backward circuits have larger depths than compact circuits but provide inherent error mitigation. Alternatively, probabilistic error cancellation is a universal way to mitigate machine errors, which enlarges the estimator variance by a factor depending on the circuit depth [57,59]. Considering probabilistic error cancellation applied to compact circuits, we can estimate the overall variance of QCMC due to both QMC and error mitigation. Third, we minimize the variance, i.e., the statistical error, by taking the optimal time-step size. We obtain the minimized variance of QCMC in the form of approximately $e^{4\gamma h_{\mathrm{tot}} t}$, where γ is the increasing rate of the variance, h_{tot} characterizes the magnitude of the Hamiltonian, and t is the evolution time.

QCMC has a variance that depends on the rate of machine errors and achieves a quantum advantage even when the error rate is finite. For the second-order LOR formula, the rate of increase of the variance has the upper bound $\gamma \simeq 2.45\epsilon^{0.82}$, where ϵ is the total gate-error rate of one elementary Lie-Trotter-Suzuki product (i.e., the

first-order product for one time step). QCMC is polynomial on a fault-tolerant quantum computer because we can suppress ϵ to any small value at a polynomial cost in quantum error correction. Suppose that the variance in classical algorithms is in the same exponential form with a finite increasing rate γ_c [2]: the quantum algorithm surpasses the classical algorithms given an error rate of $\epsilon \lesssim (\gamma_c/2.45)^{1/0.82}$. As an example, the rate of increase of the variance in Green's function Monte Carlo taking the computational basis is $\gamma_c = 1$ for a large class of qubit Hamiltonians. Compared with this classical algorithm, QCMC reduces the variance by several orders of magnitude even on a quantum computer with significant noise, e.g., by a factor of approximately 4×10^4 when $h_{\text{tot}}t = 4$ and $\epsilon = 0.1$. As a result, the sample number required in Monte Carlo is reduced by the same factor.

In this paper, we focus on the nonvariational simulation of real-time evolution. With the real-time simulation, we can construct quantum phase-estimation circuits [39] and eigenenergy filtering operators [40] to solve eigenstate and finite-temperature problems. The QCMC algorithm also provides a flexible tool for variational quantum algorithms. Here, we present two such examples. First, the ground state and other eigenstates are stationary and do not evolve with time, which leads to a way of ruling out fallacious solutions from the variational quantum eigensolver: if we find that the state evolves in the real-time simulation, the initial state must not be an eigenstate. Second, the optimizer in the variational algorithm may get stuck in a local minimum; then, real-time evolution can be used to bring the state out of the local minimum without changing the average energy. Note that by using shallow circuits in QCMC, the overall circuit combining the variational ansatz and the time evolution are still within the regime of near-term application.

This paper is organized as follows. In Sec. II, we briefly review Green's function Monte Carlo and auxiliary-field Monte Carlo. In Sec. III, we sketch the QCMC algorithm. Two series of summation formulas are introduced in Sec. IV. Details of the QCMC algorithm are presented in the form of pseudocode in Sec. V. In Sec. VI, we give two types of quantum circuits (i.e., compact circuits and forward-backward circuits) for evaluating transition amplitudes. In Sec. VII, we discuss the optimal distribution for generating samples in Monte Carlo. Two quantum error-mitigation protocols using probabilistic error cancellation and forward-backward circuits, respectively, are discussed in Sec. VIII. The QCMC algorithm and the classical QMC algorithm are compared in Sec. IX. In Sec. X, we summarize the conclusions.

II. QUANTUM MONTE CARLO

Many applications of QMC can be formalized as computing the transition amplitude $\langle \psi_f | e^{iHt^*} O e^{-iHt} | \psi_i \rangle$ given

the initial state $|\psi_i\rangle$, the final state $|\psi_f\rangle$, and the operator O. Here, H is the Hamiltonian, and t is a real or imaginary evolution time. For example, the ground-state energy of an interacting Hamiltonian can be expressed as

$$E_{\text{g.s.}} = \lim_{t \to \infty} \frac{\langle \psi_0 | e^{-Ht/2} H e^{-Ht/2} | \psi_0 \rangle}{\langle \psi_0 | e^{-Ht} | \psi_0 \rangle}, \tag{1}$$

where $|\psi_0\rangle$ is a trial ground state, which has a large overlap with the true ground state.

A canonical approach is Green's function Monte Carlo [7], in which the transition amplitude is expressed in the path-integral form:

$$\langle \psi_f | e^{iHt^*} O e^{-iHt} | \psi_i \rangle$$
$$= \int_{\mathbf{r}_0,\ldots,\mathbf{r}_N,\mathbf{r}'_0,\ldots,\mathbf{r}'_N} d\mathbf{r}_0 \cdots d\mathbf{r}_N d\mathbf{r}'_0 \cdots d\mathbf{r}'_N$$
$$\times \langle \psi_f | \mathbf{r}'_0 \rangle \langle \mathbf{r}'_0 | e^{iH(t^*/N)} | \mathbf{r}'_1 \rangle \cdots \langle \mathbf{r}'_{N-1} | e^{iH(t^*/N)} | \mathbf{r}'_N \rangle$$
$$\times \langle \mathbf{r}'_N | O | \mathbf{r}_N \rangle \langle \mathbf{r}_N | e^{-iH(t/N)} | \mathbf{r}_{N-1} \rangle \cdots$$
$$\times \langle \mathbf{r}_1 | e^{-iH(t/N)} | \mathbf{r}_0 \rangle \langle \mathbf{r}_0 | \psi_i \rangle, \tag{2}$$

where $\{|\mathbf{r}\rangle\}$ is an orthonormal basis of the Hilbert space and N is the number of time steps. The path integral is performed numerically using Monte Carlo methods.

Auxiliary-field Monte Carlo is another important approach of QMC [8,9], which is characterized by the decomposition of particle-particle interactions into interactions of particles with a group of auxiliary fields, i.e.,

$$e^{-iH\Delta t} \simeq \int ds A(s, \Delta t). \tag{3}$$

Here, $A(s, \Delta t)$ is an operator depending on the auxiliary field s. Then, the transition amplitude is expressed as

$$\langle \psi_f | e^{iHt^*} O e^{-iHt} | \psi_i \rangle$$
$$= \int ds_1 \cdots ds_N ds'_1 \cdots ds'_N \langle \psi_f | A(s'_1, -\Delta t^*) \cdots$$
$$\times A(s'_N, -\Delta t^*) O A(s_N, \Delta t) \cdots A(s_1, \Delta t) | \psi_i \rangle. \tag{4}$$

The operator $A(s, \Delta t)$ is chosen such that $\langle \psi_f | \cdots | \psi_i \rangle$ in the integral can be evaluated on a classical computer.

In diagrammatic QMC, the time-evolution amplitudes are expressed as perturbative expansions [12–15]. Suppose that the contribution of an mth-order term is $D(\xi_m, x_1, \ldots, x_m)$: the transition amplitude is a summation

of integrals in the form

$$\langle\psi_f|e^{iHt^*}Oe^{-iHt}|\psi_i\rangle$$
$$= \sum_{m=0}^{\infty}\sum_{\xi_m}\int dx_1\cdots dx_m D(\xi_m, x_1, \ldots, x_m),\quad (5)$$

where ξ_m is the index of the term and the x variables are the temporal and spatial coordinates to be integrated. These terms can be represented by Feynman diagrams. In these models, we can develop similar quantum algorithms, in which both the noninteracting time evolution and the interaction vertices can be implemented as a series of operators that can be evaluated on a quantum computer.

It often occurs that the amplitude $q = \langle\psi_f|\cdots|\psi_i\rangle$ as a function of **r** in Eq. (2) or as a function of s in Eq. (4) is not positive definite. In this case, we have to use the reweighting procedure by splitting q into its modulus and phase, i.e., $q = |q|e^{i\theta_q}$, and sample according to a probability distribution $P \propto |q|$. The expectation value of the remaining phase $\langle e^{i\theta_q}\rangle$ indicates the degree of the sign problem and if it is much smaller than 1 then the sign problem is severe. In many QMC simulations, this phase goes to zero exponentially for a large system volume or particle number, which signifies a very bad sign problem.

In some special cases, the sign problem is only induced by part of the integral variables. In other words, the amplitude q is a highly oscillating function of some variables and a smooth function of the others. This usually occurs when the system is protected by an approximate symmetry. For example, for fermion systems with equal numbers of up and down spins, a spin-independent attractive interaction respecting the SU(2) spin symmetry does not induce the sign problem. In more general problems, the realistic interaction might be dominated by such a "good" component, while other "bad" components play a minor role but induce most of the sign problem. A typical example is the nuclear force, which is approximately independent of spin and isospin at low energy [20]. The spin-isospin dependent components and the Coulomb force only contribute a small portion of the total nuclear binding energy but introduce a strong sign problem in the auxiliary field Monte Carlo calculations. Usually, these interactions can be simulated using the coupling-constant extrapolation method [60], perturbation theory [61], or the eigenvector-continuation method [62–64], at the cost of additional uncertainties.

The above problem has an alternative solution in the quantum computing era. As a quantum computer can calculate the amplitude q with the same complexity regardless of the form of the interaction, we can use the quantum computer to simulate interactions causing the sign problem, while leaving the smooth high-dimensional integrals to the classical Monte Carlo solver. For example, in the auxiliary-field Monte Carlo simulation of atomic nuclei [9], we can

simulate the repulsive Coulomb force using quantum computing. In this paper, we introduce such a hybrid simulation scheme and establish a general framework for future work in this direction.

III. QUANTUM-CIRCUIT MONTE CARLO

To implement QMC using a quantum computer, we replace the integral over the auxiliary field with a summation over unitary operators. The time-evolution operator is expressed in the summation form

$$e^{-iH\Delta t} = \sum_s c(s)U(s),\quad (6)$$

where the $U(s)$ are unitary operators and the $c(s)$ are complex coefficients. For real-time evolution, approximate summation formulas have been proposed, including truncated Taylor expansion [30,31] and linear combinations of Lie-Trotter-Suzuki products [32,33]. In this paper, we propose exact summation formulas of the real-time-evolution operator (see Sec. IV). Note that we can also construct the imaginary-time-evolution operator as a summation of unitary operators and construct any operator in the limit that the $U(s)$ form a complete basis of the operator space. By combining quantum circuits and the Monte Carlo method, our exact formulas can be implemented for any finite time-step size Δt. In quantum circuits, the gate number per time step is only moderately increased upon the Lie-Trotter-Suzuki product (see Sec. VI) and we can minimize the number of time steps N by maximizing Δt. Because of the minimized circuit depth, which is proportional to N, our formulas are practical on noisy quantum computers without fault tolerance.

With the summation expression of the time-evolution operator, the transition amplitude in the path-integral form becomes

$$\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle$$
$$= \sum_{s_1,\ldots,s_N,s'_1,\ldots,s'_N}\left[\prod_{i=1}^{N}c(s_i)c(s'_i)^*\right]\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle,\quad (7)$$

where $\mathbf{s} = (s_1, \ldots, s_N, s'_1, \ldots, s'_N)$ and

$$O_{\mathbf{s}} = U(s'_1)^{\dagger}\cdots U(s'_N)^{\dagger}OU(s_N)\cdots U(s_1).\quad (8)$$

One can realize a summation formula either by using a deterministic circuit [30–32] or sampling random circuits [33,34]. To minimize the circuit depth, we compute the transition amplitude using random circuits: we sample random unitary operators (i.e., the parameter **s**) on the classical computer, evaluate $\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle$ on the quantum computer, and then compute the path-integral summation using the Monte Carlo method on the classical computer.
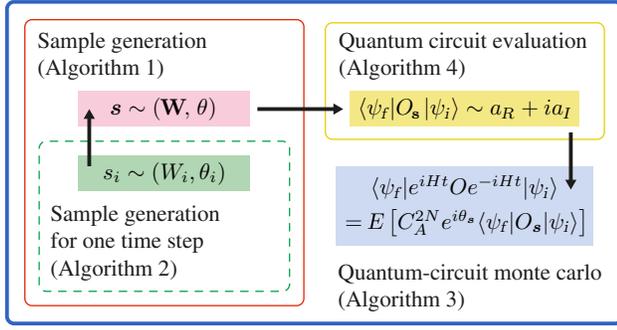
FIG. 1. A schematic diagram of the quantum-circuit Monte Carlo algorithm. The quantum computer evaluates $\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$ according to samples $\mathbf{s}$ generated by the classical computer. The final estimate of the transition amplitude $\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle$ is the empirical mean of results from the quantum computer up to a factor.

See Fig. 1 for a schematic diagram of the QCMC algorithm and see Sec. V for details.

Without fault tolerance, we use error-mitigation techniques to eliminate errors in quantum circuits. In the quantum error mitigation based on quasiprobability decomposition (i.e., probabilistic error cancellation) [57,59], each unitary circuit for evaluating $\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$ is decomposed into a linear combination of noisy circuits. Then, the overall algorithm includes Monte Carlo summations over unitary operators and also noisy circuits. Details of the error mitigation are given in Sec. VIII. Using our exact formulas of the time-evolution operator and assuming that the quasiprobability decompositions are also exact, the sampling noise in Monte Carlo is the only source of error in our algorithm.

A. Sampling noise and normalization factor

The Monte Carlo summation has a finite variance depending on the sampling approach. To compute the transition amplitude in Eq. (7), we randomly generate samples of $\mathbf{s}$ with a probability distribution $P(\mathbf{s})$. According to the importance sampling method, the variance is minimized by taking the optimal distribution

$$P(\mathbf{s}) \propto \left|\left[\prod_{i=1}^{N} c(s_i)c(s_i')^*\right]\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right|. \quad (9)$$

Implementation of the optimal distribution requires knowledge of $\left|\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right|$.

In this paper, we focus on a practical suboptimal distribution

$$P(\mathbf{s}) = \left|\prod_{i=1}^{N} c(s_i)c(s_i')^*\right|/C_A^{2N}, \quad (10)$$

where the normalization factor $C_A = \sum_s |c(s)|$ determines the variance. Taking the suboptimal distribution, the transition amplitude $\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle$ is the expected value of $C_A^{2N}e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$, where

$$\theta_\mathbf{s} = \arg\left[\prod_{i=1}^{N} c(s_i)c(s_i')^*\right]. \quad (11)$$

Formally, we have

$$\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle = \mathrm{E}\left[C_A^{2N}e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right]$$
$$= \sum_\mathbf{s} P(\mathbf{s})C_A^{2N}e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle. \quad (12)$$

Taking the suboptimal distribution, the estimator of $\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle$ is

$$\hat{A} = C_A^{2N}\left\langle e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right\rangle_{N_s}. \quad (13)$$

Here, $\langle\bullet\rangle_{N_s}$ denotes the empirical mean taken over N_s samples of $\mathbf{s}$. The variance of the estimator is

$$\mathrm{Var}\left(\hat{A}\right) = \frac{1}{N_s}C_A^{4N}\mathrm{Var}\left(e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right). \quad (14)$$

When O is a unitary operator, $|\langle\psi_f|O_\mathbf{s}|\psi_i\rangle| \leq 1$ and the variance has the upper bound

$$\mathrm{Var}\left(\hat{A}\right) \leq \frac{1}{N_s}C_A^{4N}. \quad (15)$$

In our QCMC algorithm, we use the circuits given in Sec. VI to evaluate $\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$. Each quantum circuit reports a probabilistic binary outcome, the expected value of which is either the real or imaginary part of $e^{i\theta_\mathbf{s}}\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$. We find that the suboptimal distribution (which is suboptimal when we can deterministically evaluate $\langle\psi_f|O_\mathbf{s}|\psi_i\rangle$) is actually the optimal distribution for the probabilistic evaluation without prior knowledge of $\left|\langle\psi_f|O_\mathbf{s}|\psi_i\rangle\right|$ (see Sec. VII). Accordingly, the minimum variance is

$$\mathrm{Var}\left(\hat{A}\right) = \frac{1}{M_{\mathrm{tot}}}\left(2C_A^{4N} - \left|\langle\psi_f|e^{iHt}Oe^{-iHt}|\psi_i\rangle\right|^2\right), \quad (16)$$

where $2M_{\mathrm{tot}}$ is the total number of quantum-circuit shots, and each shot is an implementation of the circuit that returns one binary measurement outcome.

We find that ideally $C_A = 1$, i.e., the variance does not increase with the number of time steps. This limit can be approached on a fault-tolerant quantum computer: we take a sufficiently small Δt, $c(1) \simeq 1$, $U(1) \simeq e^{-iH\Delta t}$ is a Lie-Trotter-Suzuki product, and terms with $s > 1$ are negligible. On a noisy quantum computer, C_A is always greater than one. A large part of our effort is devoted to minimizing C_A, in order to reduce the variance.

IV. SUMMATION FORMULAS OF TIME-EVOLUTION OPERATORS

We look for summation formulas satisfying the following criteria:

(a) The unitary operators $U(s)$ are easy to implement using elementary quantum gates, in order to reduce the gate number.
(b) The normalization factor C_A is minimized.
(c) Samples of **s** can be efficiently generated on a classical computer according to the distribution in Eq. (10).

We propose two types of summation formulas in this paper as examples of the general approach. By adding Pauli operators to Lie-Trotter-Suzuki products, we obtain POE formulas. For an lth-order product formula, the corresponding POE summation formula has the normalization factor $C_A = 1 + O(\Delta t^{l+1})$. By replacing leading-order Pauli operators with rotation operators, we obtain LOR formulas and the normalization factor is reduced to $C_A = 1 + O(\Delta t^{2l+2})$.

In the following, we first discuss Lie-Trotter-Suzuki product formulas and then introduce our summation formulas.

A. Product formulas

In this section, we review Lie-Trotter-Suzuki product formulas [55,56] and discuss some properties that are important for our discussion. Given the Hamiltonian $H = \sum_{j=1}^{M} H_j$, where the H_j are Hermitian operators, the first-order formula reads

$$S_1(\Delta t) = e^{-iH_M \Delta t} \cdots e^{-iH_1 \Delta t} = e^{-iH\Delta t} + O(\Delta t^2). \quad (17)$$

Higher-order formulas are defined recursively for any positive integer m by

$$S_{2m}(\Delta t) = K_{2m}(-\Delta t)^\dagger K_{2m}(\Delta t)$$
$$= e^{-iH\Delta t} + O(\Delta t^{2m+1}), \quad (18)$$

where $K_2(\Delta t) = S_1(\Delta t/2)$,

$$K_{2m}(\Delta t) = K_{2m-2}\left[(1 - 2rp_{r,m})\Delta t\right] S_{2m-2}\left(p_{r,m}\Delta t\right)^r, \quad (19)$$

when $m > 1$, and $p_{r,m} = \left[2r - (2r)^{1/(2m+1)}\right]^{-1}$. Here, r can be any positive integer. Each S_{2m} is a product of $2r + 1$ S_{2m-2} operators.

For the first-order formula, we define the correction operator

$$V_1(\Delta t) \equiv e^{-iH\Delta t} S_1(\Delta t)^\dagger = e^{-i\sum_{k=2}^{\infty} R_1^{(k)} \Delta t^k}, \quad (20)$$

where $R_1^{(k)}$ are operators that are independent of Δt. Because $V_1(\Delta t)$ is unitary for all real Δt, all $R_1^{(k)}$ are Hermitian operators. Then,

$$V_1(\Delta t) = \mathbb{1} - iL_1(\Delta t) + O(\Delta t^4), \quad (21)$$

where the leading-order operator

$$L_1(\Delta t) = R_1^{(2)}\Delta t^2 + R_1^{(3)}\Delta t^3 \quad (22)$$

is Hermitian. Later, we show that the Hermitian leading-order operator is important for minimizing the normalization factor C_A.

For higher-order formulas, the correction operators are

$$V_{2m}(\Delta t) \equiv K_{2m}(-\Delta t)e^{-iH\Delta t}K_{2m}(\Delta t)^\dagger$$
$$= e^{-i\sum_{k=2m+1}^{\infty} R_{2m}^{(k)} \Delta t^k}, \quad (23)$$

where $R_{2m}^{(k)}$ are Hermitian operators that are independent of Δt. Because of the symmetric form, $V_{2m}(\Delta t) = V_{2m}(-\Delta t)^\dagger$ for all real Δt and $R_{2m}^{(k)} = 0$ for all even k [56]. Then,

$$V_{2m}(\Delta t) = \mathbb{1} - iL_{2m}(\Delta t) + O(\Delta t^{4m+2}), \quad (24)$$

where the leading-order operator

$$L_{2m}(\Delta t) = \sum_{k=m}^{2m} R_{2m}^{(2k+1)}\Delta t^{2k+1} \quad (25)$$

is Hermitian. For the second-order formula,

$$L_2(\Delta t) = R_2^{(3)}\Delta t^3 + R_2^{(5)}\Delta t^5. \quad (26)$$

B. Summation formulas

To simplify the quantum circuits, we work with Pauli operators $\mathbf{P}_n = \{I, X, Y, Z\}^{\otimes n}$ as the basis of matrix space, where n is the number of qubits. Without loss of generality, we assume that each term of the Hamiltonian is a Pauli operator, i.e., $H_j = h_j \sigma_j$, where $\sigma_j \in \mathbf{P}_n$, and h_j is a real coefficient. We define $h_{\text{tot}} \equiv \sum_j |h_j|$, which characterizes the magnitude of the Hamiltonian.

Given the time-evolution operator, there exist many different summation formulas $e^{-iH\Delta t} = \sum_s c(s)U(s)$. Each formula represents a sampling protocol in Monte Carlo. For example,

$$e^{-iH\Delta t} = \sum_{\sigma \in \mathbf{P}_n} 2^{-n}\text{Tr}\left(\sigma e^{-iH\Delta t}\right)\sigma. \quad (27)$$

Such a formula is impractical, because the computing of the coefficients $\text{Tr}\left(\sigma e^{-iH\Delta t}\right)$ on a classical computer is usually difficult when n is large.

TABLE I. The normalization factors of summation formulas. In the table, $x \equiv h_{\text{tot}}\Delta t$.

Pauli-operator-expansion formulas			$C_A = 1 + C_L + C_T$	
Leading-order-rotation formulas			$C_A = \sqrt{1 + C_L^2} + C_T$	
High-order contribution C_T			$e^{\lambda x} - \sum_{k=0}^{2l+1}(1/k!)(\lambda x)^k$	
l (order of formula)	0	1	2	$2m$
λ	1	2	2	$1 + \prod_{k=2}^{m}\left(4r p_{r,k} - 1\right)$
Leading-order contribution C_L	x	$\frac{1}{2}(2x)^2 + \frac{1}{6}(2x)^3$	$\frac{1}{6}(2x)^3 + \frac{1}{120}(2x)^5$	$\sum_{k=m}^{2m}[1/(2k+1)!](\lambda x)^{2k+1}$
Simplified leading-order contribution C_L		$< \frac{1}{2}x^2 + \frac{1}{6}(2x)^3$	$< \frac{1}{18}x^3 + \frac{1}{120}(2x)^5$	

For the practical implementation, we express the time-evolution operator in the form

$$e^{-iH\Delta t} = K_L V K_R, \tag{28}$$

where K_L and K_R are unitary operators in the Lie-Trotter-Suzuki product form and V is the correction operator [see Eqs. (20) and (23)]. We apply the Taylor expansion to the correction operator to obtain the summation formula. We divide the Taylor expansion into three parts, $V = \mathbb{1} - iL + T$, where L is the leading-order operator, and T is the high-order operator. The normalization factor of a POE summation formula is $C_A = 1 + C_L + C_T$, where C_L and C_T are contributions of L and T, respectively. The normalization factor of a LOR summation formula is $C_A = \sqrt{1 + C_L^2} + C_T$. The normalization factors of all the formulas are summarized in Table I.

1. Zeroth-order Pauli-operator-expansion formula

The direct Taylor expansion of the time-evolution operator gives the zeroth-order summation formula

$$V_0(\Delta t) = e^{-iH\Delta t} = \mathbb{1} - iL_0(\Delta t) + T_0(\Delta t), \tag{29}$$

where the Hermitian leading-order operator is

$$L_0(\Delta t) = \sum_j h_j \Delta t \sigma_j \tag{30}$$

and the high-order operator is

$$T_0(\Delta t) = \sum_{k=2}^{\infty} \sum_{j_1,\ldots j_k=1}^{M} \frac{\prod_{a=1}^{k}\left(-ih_{j_a}\Delta t\right)}{k!} \sigma_{j_k}\cdots\sigma_{j_1}. \tag{31}$$

The normalization factor is given by $C_L = h_{\text{tot}}\Delta t$ and $C_T = e^{h_{\text{tot}}\Delta t} - (1 + h_{\text{tot}}\Delta t)$.

2. First-order Pauli-operator-expansion formula

According to the first-order product formula, we express the time-evolution operator as

$$e^{-iH\Delta t} = V_1(\Delta t)S_1(\Delta t). \tag{32}$$

We obtain the summation formula by applying the Taylor expansion to each exponential in the correction operator,

$$\begin{aligned} V_1(\Delta t) &= e^{-iH\Delta t}e^{ih_1\sigma_1\Delta t}\cdots e^{ih_M\sigma_M\Delta t} \\ &= \mathbb{1} - iL_1(\Delta t) + T_1(\Delta t), \end{aligned} \tag{33}$$

where

$$L_1(\Delta t) = iF_1^{(2)}(\Delta t) + iF_1^{(3)}(\Delta t), \tag{34}$$

$$T_1(\Delta t) = \sum_{k=4}^{\infty} F_1^{(k)}(\Delta t), \tag{35}$$

and

$$\begin{aligned} F_1^{(k')}(\Delta t) = &\sum_{k,k_1,\ldots,k_M=0}^{\infty} \sum_{j_1,\ldots j_k=1}^{M} \delta_{k',k+\sum_{j=1}^{M}k_j} \frac{\prod_{a=1}^{k}\left(-ih_{j_a}\Delta t\right)}{k!} \\ &\times \left[\prod_{j=1}^{M}\frac{(ih_j\Delta t)^{k_j}}{k_j!}\right]\sigma_{j_k}\cdots\sigma_{j_1}\sigma_1^{k_1}\cdots\sigma_M^{k_M}. \end{aligned} \tag{36}$$

Note that the first term in L_1 is $O(\Delta t^2)$ according to discussions on product formulas. For the first-order formula, the normalization factor is given by $C_L = \frac{1}{2}(2h_{\text{tot}}\Delta t)^2 + \frac{1}{6}(2h_{\text{tot}}\Delta t)^3$ and $C_T = e^{2h_{\text{tot}}\Delta t} - \sum_{k=0}^{3}(1/k!)(2h_{\text{tot}}\Delta t)^k$.

3. Second-order Pauli-operator-expansion formula

Similar to the first-order formula, according to the second-order product formula, we express the time-evolution operator as

$$e^{-iH\Delta t} = S_1\left(-\frac{\Delta t}{2}\right)^{\dagger} V_2(\Delta t)S_1\left(\frac{\Delta t}{2}\right). \tag{37}$$

The Taylor expansion of the correction operator reads

$$V_2(\Delta t) = \mathbb{1} - iL_2(\Delta t) + T_2(\Delta t), \quad (38)$$

where

$$L_2(\Delta t) = iF_2^{(3)}(\Delta t) + iF_2^{(5)}(\Delta t), \quad (39)$$

$$T_2(\Delta t) = \sum_{k=6}^{\infty} F_2^{(k)}(\Delta t), \quad (40)$$

and

$$F_2^{(k')}(\Delta t) = \sum_{k,k_1,\dots,k_1',\dots=0}^{\infty} \sum_{j_1,\dots,j_k=1}^{M} \delta_{k',k+\sum_{j=1}^{M}(k_j+k_j')}$$

$$\times \frac{\prod_{a=1}^{k}\left(-ih_{j_a}\Delta t\right)}{k!} \left[\prod_{j=1}^{M} \frac{(ih_j\,\Delta t/2)^{k_j+k_j'}}{k_j!\,k_j'!}\right]$$

$$\times \sigma_M^{k_M'} \cdots \sigma_1^{k_1'} \sigma_{j_k} \cdots \sigma_{j_1} \sigma_1^{k_1} \cdots \sigma_M^{k_M}. \quad (41)$$

According to discussions on product formulas, L_2 only contain Δt^3 and Δt^5 terms. For the second-order formula, the normalization factor is given by $C_L = \frac{1}{6}(2h_{\text{tot}}\Delta t)^3 + \frac{1}{120}(2h_{\text{tot}}\Delta t)^5$ and $C_T = e^{2h_{\text{tot}}\Delta t} - \sum_{k=0}^{5}(1/k!)(2h_{\text{tot}}\Delta t)^k$.

4. Higher-order Pauli-operator-expansion formulas

For the $2m$th-order formula, we express the time-evolution operator as

$$e^{-iH\Delta t} = K_{2m}(-\Delta t)^\dagger V_{2m}(\Delta t) K_{2m}(\Delta t). \quad (42)$$

Then, we can obtain the POE summation formula by applying a Taylor expansion to each exponential in the correction operator V_{2m}, similar to the first- and second-order formulas. The normalization factor of the $2m$th-order formula is given by $C_L = \sum_{k=m}^{2m}[1/(2k+1)!](\lambda h_{\text{tot}}\Delta t)^{2k+1}$ and $C_T = e^{\lambda h_{\text{tot}}\Delta t} - \sum_{k=0}^{4m+1}(1/k!)(\lambda h_{\text{tot}}\Delta t)^k$. Here, the factor $\lambda = 1 + \prod_{k=2}^{m}(4rp_{r,k}-1)$ is due to the backward evolution with the time $(1 - 2rp_{r,m})\Delta t$ in the product formula.

5. Simplified leading-order operators

By combining terms with the same Pauli operator in the summation formula, we can reduce the normalization factor. For example, if both $\alpha\sigma$ and $-\alpha\sigma$ exist in the summation formula, the contribution to the normalization factor is $2|\alpha|$, which is reduced to zero after combining like terms. We apply this approach to $F_1^{(2)}$ and $F_2^{(3)}$ in L_1 and L_2, respectively, to minimize the dominant contribution to the normalization factor. See Appendix A for the simplified expressions of $F_1^{(2)}$ and $F_2^{(3)}$. As a result, the leading-order contributions are reduced to

$C_L < \frac{1}{2}(h_{\text{tot}}\Delta t)^2 + \frac{1}{6}(2h_{\text{tot}}\Delta t)^3$ in the first-order formula and $C_L < \frac{1}{18}(h_{\text{tot}}\Delta t)^2 + \frac{1}{120}(2h_{\text{tot}}\Delta t)^5$ in the second-order formula.

6. Leading-order-rotation formulas

The leading-order operator L_l is Hermitian, which allows us to reduce its contribution to the normalization factor C_A from $O(\Delta t^{l+1})$ to $O(\Delta t^{2l+2})$. We suppose that the Pauli-operator summation form of L_l is

$$L_l = \sum_u \alpha_u \tau_u, \quad (43)$$

where the $\tau_u \in \mathbf{P}_n$ are Pauli operators. Here, all α_u are real because L_l is Hermitian, which is the key to LOR formulas. To minimize the normalization factor, we express the leading-order terms as a summation of rotation operators,

$$\mathbb{1} - iL_l = \sum_u \beta_u e^{-i\mathrm{sgn}(\alpha_u)\phi\tau_u}, \quad (44)$$

where $\phi = \arctan(C_L)$, $\beta_u = |\alpha_u|/\sin\phi$ and $C_L = \sum_u |\alpha_u|$.

The normalization factor contributed by $\mathbb{1} - iL_l$ is $1 + C_L$ in POE formulas, which is reduced to $\sum_i |\beta_u| = C_L/\sin\phi = \sqrt{1+C_L^2} \simeq 1 + C_L^2/2$ in LOR formulas. Note that $C_L = O(\Delta t^{l+1})$ and $C_T = O(\Delta t^{2l+2})$. By using LOR formulas, we reduce the normalization factor C_A from $1 + O(\Delta t^{l+1})$ to $1 + O(\Delta t^{2l+2})$.

We have introduced all of our summation formulas. We remark that our summation formulas are used for sampling random $U(s)$ rather than sampling quantum operations [34], which corresponds to a summation of completely positive maps instead of operators.

C. Comparison between formulas

Now, we compare different formulas of the time-evolution operator in the fault-tolerance limit, i.e., gate errors are negligible. In this case, we can use deep quantum circuits to implement the formulas and take a sufficiently small time-step size Δt. We leave the discussions on noisy quantum computing to Secs. VIII and IX.

When gate errors are negligible, sampling noise is the only source of error for our exact summation formulas. The error due to sampling noise is approximately $(1/\sqrt{N_s})C_A^{2N}$. Therefore, the error for the lth-order POE formula is approximately $1/\sqrt{N_s} + (2N/\sqrt{N_s})O(\Delta t^{l+1})$ and the error for the lth-order LOR formula is approximately $1/\sqrt{N_s} + (2N/\sqrt{N_s})O(\Delta t^{2l+2})$.

For Lie-Trotter-Suzuki product formulas, there are two sources of error: the error due to finite Δt, i.e., the formulas are approximate, and the error due to sampling noise. The error due to finite Δt is systematic and cannot be reduced by increasing the number of samples. For

the lth-order product formula, the error is approximately $1/\sqrt{N_s} + NO(\Delta t^{l+1})$, where the first term is due to the sampling noise and the second term is due to the finite Δt. We note that on a fault-tolerant quantum computer, we can use amplitude amplification to accelerate the evaluation of an amplitude of the wave function [65]. Amplitude amplification can be applied to product formulas; how to apply it to our summation formulas is an open question.

We find that for the same order of formulas, our summation formulas have a smaller error than product formulas, due to the factor $1/\sqrt{N_s}$ in the Δt term and the increased exponent of Δt (for LOR formulas). The reduced error is at the cost of an increased gate number per time step: to implement our formulas, we need to add a correction operator to the Lie-Trotter-Suzuki product for each time step. A correction operator is either a Pauli operator σ or a rotation operator in the form $e^{-i\phi\sigma}$. In Sec. C, we show that implementation of the correction operator for POE and LOR formulas requires at most n and $4n$ controlled-NOT gates, respectively, on an all-to-all qubit network ($4n - 3$ and $8n - 4$ gates, respectively, on a linear qubit network). Here, n is the qubit number. Unless the Hamiltonian has the simplest structure, such as the one-dimensional quantum Ising model, it is reasonable to assume that the gate number for the first-order Lie-Trotter-Suzuki product S_1 is more than $2n$. Therefore, the gate-number increment in each time step is moderate.

The linear combination of Lie-Trotter-Suzuki products can efficiently reduce the error due to finite Δt [32,33]. The simplest example is $e^{-iH\Delta t} = \frac{4}{3}S_2(\Delta t/2)^2 - \frac{1}{3}S_2(\Delta t) + O(\Delta t^5)$. We find that the error for our second-order LOR formula converges faster as $O(\Delta t^6)$ and the gate number is smaller compared with S_2^2 (assuming that the gate number for one S_2 is larger than a correction operator).

V. ALGORITHM

The algorithm consists of three phases. First, the classical computer generates samples of **s** according to the distribution given by Eq. (10) and composes corresponding quantum circuits. Second, the quantum computer implements circuits to evaluate $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$. Finally, with results from the quantum computer, the classical computer calculates the expected value of $e^{i\theta_\mathbf{s}} \langle \psi_f | O_\mathbf{s} | \psi_i \rangle$ and returns the final estimate of the transition amplitude $\langle \psi_f | e^{iHt} O e^{-iHt} | \psi_i \rangle$.

In this section, we present the first and final phases of the algorithm, which are implemented on the classical computer. We leave details of the second phase, i.e., the quantum computing, to Sec. VI. We focus on second-order summation formulas and the algorithms for the other summation formulas are similar.

Our algorithm has some implicit connections to the diagrammatic Monte Carlo, in which the Feynman diagrams represent the perturbative expansions for interacting amplitudes. Similarly, the summation formulas in our algorithm are perturbativelike expansions around Lie-Trotter-Suzuki products. In our case, each term represents a path in the Hilbert space defined by the unitary operator $U(s)$ instead of ξ_m and x variables and these paths constitute the time evolution, which resembles the path-integral picture. This connection may be further explored to design new quantum algorithms.

A. Sampling algorithm

The Hamiltonian is specified by a vector of real numbers $\mathbf{h} = (h_1, \ldots, h_M)$ and a vector of Pauli operators $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M)$. Given the evolution time t, we need to choose a number of time steps N; then, the corresponding time-step size is $\Delta t = t/N$. These parameters, $\mathbf{h}, \boldsymbol{\sigma}, N$, and Δt, are inputs to the sampling algorithm. To present the algorithm in a way that works for both POE and LOR formulas, we introduce an additional input parameter $F = P, R$ to denote POE and LOR formulas, respectively.

In the second-order summation formulas, each term is in the form $S_1[-(\Delta t/2)]^\dagger W S_1(\Delta t/2)$: In the POE formula, W is always a Pauli operator; in the LOR formula, W is either a rotation operator or a Pauli operator. Taking $U(s_i) = S_1[-(\Delta t/2)]^\dagger W_i S_1(\Delta t/2)$ and $U(s_i') = S_1[-(\Delta t/2)]^\dagger W_i' S_1(\Delta t/2)$, we have

$$O_\mathbf{s} = S_1^\dagger W_1'^\dagger S_1'^\dagger \cdots S_1^\dagger W_N'^\dagger S_1'^\dagger O S_1' W_N S_1 \cdots S_1' W_1 S_1, \quad (45)$$

Here, we use the notations $S_1 = S_1(\Delta t/2)$ and $S_1' = S_1[-(\Delta t/2)]^\dagger$ for simplicity. Given the vector of correction operators

$$\mathbf{W} = (W_1, \ldots, W_N, W_1', \ldots, W_N'), \quad (46)$$

the quantum computer can evaluate $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$.

In the final phase, the classical computer estimates the transition amplitude by computing the expected value of $e^{i\theta_\mathbf{s}} \langle \psi_f | O_\mathbf{s} | \psi_i \rangle$. Therefore, the sampling algorithm also needs to output $\theta_\mathbf{s}$.

Overall, the outputs of the sampling algorithm are $\mathbf{W}$ and θ. The procedure for generating $\mathbf{W}$ and θ is given in Algorithm 1. Algorithm 2 is a subroutine for processing one time step.

1: **function** SAMGEN(F, $\boldsymbol{h}, \boldsymbol{\sigma}, N, \Delta t$)
2:     **for** $i = 1$ to N **do**
3:         $(W_i, \theta_i) \leftarrow$ SAMGENONESTEP(F, $\boldsymbol{h}, \boldsymbol{\sigma}, \Delta t$)
4:         $(W_i', \theta_i') \leftarrow$ SAMGENONESTEP(F, $\boldsymbol{h}, \boldsymbol{\sigma}, \Delta t$)
5:     $\boldsymbol{W} \leftarrow (W_1, \ldots, W_N, W_1', \ldots, W_N')$
6:     $\theta \leftarrow \sum_{i=1}^{N} (\theta_i - \theta_i')$
7:     Output $(\boldsymbol{W}, \theta)$.

Algorithm 1. Sample generation.

1: **function** SAMGENONESTEP(F, $\boldsymbol{h}, \boldsymbol{\sigma}, \Delta t$)

2:     Compute L_2 according to Eq. (39), simplify L_2 by combining like terms, obtain the final expression $L_2 = \sum_u \alpha_u \tau_u$. $\triangleright \tau_u \in \boldsymbol{P}_n$

3:     $C_L \leftarrow \sum_u |\alpha_u|$

4:     $C_T \leftarrow e^{2h_{\text{tot}}\Delta t} - \sum_{k=0}^{5} \frac{1}{k!}(2h_{\text{tot}}\Delta t)^k$ $\triangleright h_{\text{tot}} = \sum_j |h_j|$

5:     **if** F = P **then** $C_A \leftarrow 1 + C_L + C_T$

6:     **else if** F = R **then** $C_A \leftarrow \sqrt{1 + C_L^2} + C_T$

7:     Choose O from L and T with probabilities $(C_A - C_T)/C_A$ and C_T/C_A, respectively.

8:     **if** O = L **then** $\triangleright$ Sample from leading-order terms

9:         **if** F = P **then** Choose (W, θ) from $(\mathbb{1}, 0)$ and $\{(\tau_u, \arg(-i\alpha_u))\}$ with probabilities $1/(1 + C_L)$ and $\{|\alpha_u|/(1 + C_L)\}$, respectively.

10:         **else if** F = R **then** Choose (W, θ) from $\{(e^{-i\text{sgn}(\alpha_u)\phi\tau_u}, 0)\}$ with probabilities $\{|\alpha_u|/C_L\}$. $\triangleright \phi = \arctan(C_L)$

11:     **else if** O = T **then** $\triangleright$ Sample from high-order terms

12:         $k, k_j, k_j' \leftarrow 0$

13:         **while** $k + \sum_{j=1}^{M}(k_j + k_j') < 6$ **do**

14:             $k \leftarrow \text{POISSON}(h_{\text{tot}}\Delta t)$ $\triangleright$ POISSON(x) returns $k \in \{0, 1, \ldots\}$ with the probability $e^{-x}x^k/k!$.

15:             **for** $j = 1$ to M **do**

16:                 $k_j \leftarrow \text{POISSON}(|h_j|\Delta t/2)$

17:                 $k_j' \leftarrow \text{POISSON}(|h_j'|\Delta t/2)$

18:             **for** $a = 1$ to k **do** Choose j_a from $\{1, \ldots, M\}$ with probabilities $\{|h_{j_a}|/h_{\text{tot}}\}$.

19:             $W \leftarrow \zeta^* \sigma_M^{k_M'} \cdots \sigma_1^{k_1'} \sigma_{j_k} \cdots \sigma_{j_1} \sigma_1^{k_1} \cdots \sigma_M^{k_M}$ $\triangleright$ Take $\zeta = \pm 1, \pm i$ to meet $W \in \boldsymbol{P}_n$.

20:             $\theta \leftarrow \arg\left(\zeta \prod_{a=1}^{k}(-ih_{j_a}) \times \prod_{j=1}^{M}(ih_j)^{k_j + k_j'}\right)$

21:     Output (W, θ).

Algorithm 2. Sample generation for one time step.

B. Quantum-circuit Monte Carlo algorithm

Using the Monte Carlo summation to compute the path-integral formula in Eq. (7), we need to choose two parameters N_s and M_s, which are the number of **s** samples and the number of shots per quantum circuit for evaluating $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$, respectively. Because the transition amplitude is a complex number in general, the quantum computing returns two real numbers $a_{R,\mathbf{s}}$ and $a_{I,\mathbf{s}}$, which are estimates of the real and imaginary parts of $e^{i\theta_\mathbf{s}}\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$, respectively. By computing expected values of $a_{R,\mathbf{s}}$ and $a_{I,\mathbf{s}}$, we obtain the transition amplitude $\langle \psi_f | e^{iHt}Oe^{-iHt} | \psi_i \rangle$ up to the factor C_A^{2N}. QCMC is summarized in Algorithm 3.

C. Quantum-circuit Monte Carlo on classical computer

In this section, we show that QCMC with the zeroth-order POE formula is equivalent to QMC on a classical computer. In the zeroth-order POE formula, the time-evolution operator is expanded into the form $e^{-iH\Delta t} = \sum_s c(s)\sigma_s$, where the $\sigma_s \in \mathbf{P}_n$ are Pauli operators. We can

---

1: Input F, $\boldsymbol{h}, \boldsymbol{\sigma}, N, \Delta t, N_s, M_s$.

2: **for** $v = 1$ to N_s **do** $\triangleright v$ is the label of $\boldsymbol{s}$ samples.

3:     $(\boldsymbol{W}, \theta) \leftarrow \text{SAMGEN}(F, \boldsymbol{h}, \boldsymbol{\sigma}, N, \Delta t)$

4:     $(a_{R,v}, a_{I,v}) \leftarrow \text{QUANTUMCIRCUITS}(\ldots, \boldsymbol{W}, \theta, M_s)$

5: $\hat{A} \leftarrow C_A^{2N} \frac{1}{N_s} \sum_{v=1}^{N_s}(a_{R,v} + ia_{I,v})$

6: Output $\hat{A}$ as the estimate of $\langle \psi_f | e^{iHt}Oe^{-iHt} | \psi_i \rangle$.

Algorithm 3. Quantum-circuit Monte Carlo.

express a Pauli operator as

$$\sigma = i^{x_1 z_1} X^{x_1} Z^{z_1} \otimes \cdots \otimes i^{x_n z_n} X^{x_n} Z^{z_n}, \quad (47)$$

where $x_a, z_a = 0, 1$, and $i^{x_a z_a} X^{x_a} Z^{z_a} = I, X, Y, Z$ is a single-qubit Pauli operator of qubit a. We consider computational basis states in the form $\bigotimes_{a=1}^{n} |\mu_a\rangle$, where $\mu_a = 0, 1$. A Pauli operator acting on a basis state always results in a basis state, i.e., $\sigma \bigotimes_{a=1}^{n} |\mu_a\rangle = \bigotimes_{a=1}^{n} i^{x_a z_a}(-1)^{z_a \mu_a} |\mu_a \oplus x_a\rangle$, where $\oplus$ denotes the modulo-2 addition. Therefore, Pauli operators acting on basis states can be efficiently calculated on a classical computer. Similarly, Pauli operators acting on product states in the form $\bigotimes_{j=1}^{n} |\psi_j\rangle$ and stabilizer states [66] can also be efficiently calculated on a classical computer. In the following, we focus on computational basis states.

The zeroth-order POE formula is auxiliary-field Monte Carlo, which takes the space of Pauli operators as the auxiliary field. Suppose that the initial and final states are computational basis states and O is a Pauli operator. We can evaluate

$$\langle \psi_f | O_\mathbf{s} | \psi_i \rangle = \langle \psi_f | \sigma_{s_1'} \cdots \sigma_{s_N'} O \sigma_{s_N} \cdots \sigma_{s_1} | \psi_i \rangle, \quad (48)$$

on a classical computer. By expressing the initial and final states as linear combinations of basis states and the operator O as a linear combination of Pauli operators, we can evaluate $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$ on a classical computer for the general states and the operator. Therefore, we can implement QCMC with the zeroth-order POE formula without using a quantum computer.

Now, we consider a class of Hamiltonians without short-time interference between Pauli operators. Each Pauli operator corresponds to two binary strings $(x_1, \ldots, x_n)$ and $(z_1, \ldots, z_n)$. If the x strings of two Pauli operators σ and τ are different, we have $\langle \psi | \tau\sigma | \psi \rangle = 0$ for all computational basis states $|\psi\rangle = \bigotimes_{a=1}^{n} |\mu_a\rangle$. The short-time-evolution operator $e^{-iH\Delta t} \simeq \mathbb{1} - iH\Delta t$ acting on a basis state results in

$$e^{-iH\Delta t}|\psi\rangle \simeq |\psi\rangle - i\Delta t \sum_j h_j \sigma_j |\psi\rangle. \quad (49)$$

We find that there is no interference between the terms if and only if $\langle \psi | \tau\sigma | \psi \rangle = 0$ for all $\sigma, \tau \in \{\mathbb{1}\} \cup \{\sigma_j\}$: i.e.,

the Pauli operators in the Hamiltonian have different x strings.

For Hamiltonians without short-time interference, the zeroth-order POE formula is equivalent to Green's function Monte Carlo, which takes the computational basis. In Green's function Monte Carlo, we sample states $|\mathbf{r}\rangle$; in QCMC, we sample Pauli operators. Substituting the computational basis for $\{|\mathbf{r}\rangle\}$, the transition amplitude of each time step reads $\langle\psi'|e^{-iH\Delta t}|\psi\rangle$, where $|\psi'\rangle = \bigotimes_{a=1}^{n}|\mu_a'\rangle$. For a Hamiltonian without short-time interference, basis states $|\psi'\rangle$ with nonzero $\langle\psi'|e^{-iH\Delta t}|\psi\rangle$ and Pauli operators in $\{\mathbb{1}\} \cup \{\sigma_i\}$ have one-to-one correspondence in the limit of small Δt. Therefore, sampling Pauli operators is equivalent to sampling basis states $|\psi'\rangle$.

The class of Hamiltonians without short-time interference includes those are hard for simulation in classical computing. In Appendix B, we show that the Fermi-Hubbard model on any bipartite lattice (e.g., the square lattice) can be encoded into a qubit Hamiltonian without short-time interference, using the Jordan-Wigner transformation.

VI. QUANTUM CIRCUITS

We propose quantum circuits for evaluating the transition amplitude of the operator $O_{\mathbf{s}}$ and the gate number per time step is moderately increased upon the Lie-Trotter-Suzuki product. To measure the transition amplitude, we need to introduce an ancillary qubit, which controls the evolution of n qubits representing the system. In Eq. (45), the evolution is driven by Lie-Trotter-Suzuki products R_1 and correction operators W. Our circuits are simplified in two ways: first, we avoid controlled Lie-Trotter-Suzuki products and only use controlled corrections; and, second, the correction operators are either Pauli operators σ or rotation operators $e^{-i\phi\sigma}$.

We propose two types of circuits. For compact circuits, the circuit depth is the same as the Lie-Trotter-Suzuki decomposition with additional controlled-correction gates. For forward-backward circuits, the circuit depth is doubled but they provide inherent quantum error mitigation. In this section, we also show how to efficiently decompose a controlled-correction gate into elementary gates. We assume that O is a unitary operator and we can evaluate a general operator by decomposing it into a linear combination of unitary operators.

A. Compact circuit

The compact circuit for second-order formulas is shown in Fig. 2(a). The circuits for the other summation formulas are similar. For the first-order formulas, we remove the S_1' products from the circuit; for the zeroth-order formulas, we remove both the S_1 and the S_1' products; and by adding more S_1 and S_1' products, the circuit can be used for higher-order formulas. If we ignore controlled-correction gates,

```
 1: function QUANTUMCIRCUITS(h, σ, N, Δt, W, θ, M_s)
 2:     a_R, a_I ← 0
 3:     for i = 1 to M_s do
 4:         Implement the circuit, measure cos θ X_a + sin θ Y_a
        and collect the outcome μ_R = ±1.
 5:         a_R ← a_R + μ_R
 6:         Implement the circuit, measure sin θ X_a − cos θ Y_a
        and collect the outcome μ_I = ±1.
 7:         a_I ← a_I + μ_I
 8:     a_R ← a_R/M_s
 9:     a_I ← a_I/M_s
10:     Output (a_R, a_I).
```

Algorithm 4.   Quantum-circuit evaluation.

the compact circuit for the lth-order summation formula is the same as the circuit for the lth-order Lie-Trotter-Suzuki product formula.

Now, we focus on second-order formulas, and the analysis for the other formulas is similar. The final state of the compact circuit (before the basis-adjusting gate B) is

$$|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle_a \otimes OS_1'W_NS_1\cdots S_1'W_1S_1|\psi_i\rangle + |1\rangle_a \right.$$
$$\left. \otimes \; S_1'W_N'S_1\cdots S_1'W_1'S_1|\psi_f\rangle\right). \tag{50}$$

Measuring the ancillary qubit, we obtain

$$\langle\Psi|X_a|\Psi\rangle = \mathrm{Re}\left(\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle\right), \tag{51a}$$

$$\langle\Psi|Y_a|\Psi\rangle = -\mathrm{Im}\left(\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle\right), \tag{51b}$$

where X_a and Y_a are Pauli operators of the ancillary qubit. Here, we use Eq. (45). The procedure for evaluating $\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle$ using compact circuits is given in Algorithm 4.

B. Forward-backward circuit

The forward-backward circuit for second-order formulas is shown in Fig. 2(b). Compared with the compact circuit, the number of S_1 and S_1' products is doubled. The final state of the circuit is

$$|\Phi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle_a \otimes |0\rangle^{\otimes n} + |1\rangle_a \otimes U_f^\dagger O_{\mathbf{s}}|\psi_i\rangle\right). \tag{52}$$

Here, we use Eq. (45). Measuring the ancillary qubit, we obtain

$$\langle\Phi|X_a|\Phi\rangle = \mathrm{Re}\left(\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle\right), \tag{53a}$$

$$\langle\Phi|Y_a|\Phi\rangle = \mathrm{Im}\left(\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle\right). \tag{53b}$$

The procedure for evaluating $\langle\psi_f|O_{\mathbf{s}}|\psi_i\rangle$ is similar to Algorithm 4. Note that evaluating the transition amplitude in this way does not provide inherent error mitigation. We discuss the inherent error mitigation using postselection in Sec. VIII B.
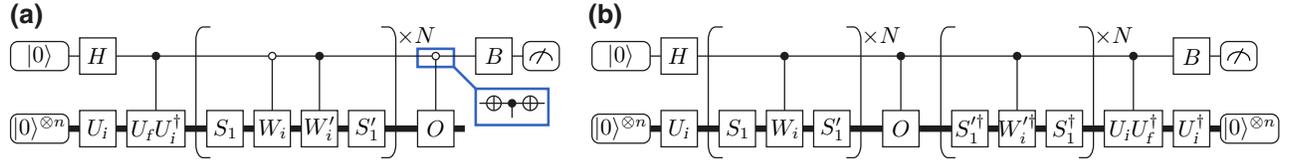
FIG. 2. Quantum circuits for evaluating $e^{i\theta_s}\langle\psi_f|O_s|\psi_i\rangle$. The qubit on the top is the ancillary qubit. The empty circle in the blue box denotes a controlled-U gate that U acts on the n qubits when the ancillary qubit is in $|0\rangle$. Unitary operators U_i and U_f prepare the initial and final states, respectively, i.e., $|\psi_i\rangle = U_i|0\rangle^{\otimes n}$ and $|\psi_f\rangle = U_f|0\rangle^{\otimes n}$. The gate B is for adjusting the measurement basis. For simplicity, we use the notations $S_1 = S_1(\Delta t/2)$ and $S_1' = S_1[-(\Delta t/2)]^\dagger$.

C. Controlled-correction gates

We consider two types of qubit networks. On the all-to-all network, controlled-NOT gates on all pairs of qubits are available. On the linear network, only controlled-NOT gates on nearest-neighboring qubits are allowed. We use $\Lambda_{a,b}$ to denote the controlled-NOT gate that a and b are the control and target qubits, respectively. Because the error rate of controlled-NOT gates is usually much higher than that of single-qubit gates, we only count controlled-NOT gates and we minimize their number.

A general Pauli operator σ is equivalent to an X-product Pauli operator (i.e., a tensor product of I and X) up to a unitary transformation. For the σ in Eq. (47), the transformation is $\tilde{R} = H^{z_1(1-x_1)}S^{z_1x_1} \otimes \cdots \otimes H^{z_n(1-x_n)}S^{z_nx_n}$, where H is the Hadamard gate, and S is the $\pi/4$ phase gate. This transformation leads to $\tilde{\sigma} = \tilde{R}^\dagger\sigma\tilde{R} = X^{x_1\vee z_1} \otimes \cdots \otimes X^{x_n\vee z_n}$, where $x_a \vee z_a = 1 - (1-x_a)(1-z_a)$.

Implementation of the controlled-$\tilde{\sigma}$ gate on the all-to-all network is straightforward. For each qubit with $x_a \vee z_a = 1$, we apply the controlled-NOT gate $\Lambda_{0,a}$, where qubit 0 is the ancillary qubit. The controlled-$\tilde{\sigma}$ gate is $\prod_{a=1}^n \Lambda_{0,a}^{x_a\vee z_a}$ and the number of controlled-NOT gates is $N_\Lambda = \sum_{a=1}^n x_a \vee z_a \leq n$.

In the compact circuit shown in Fig. 2(a), there are two controlled-correction gates in each time step, corresponding to W_i and W_i', respectively. When $W_i = \tau$ and $W_i' = \tau'$ are Pauli operators, we can combine the two controlled-correction gates into one controlled-σ gate in the following way. Note that $\tau'\tau = \zeta\sigma$, where ζ is a phase factor. We apply τ first, then a controlled-σ gate, and finally a phase gate $\text{diag}(1,\zeta)$ on the ancillary qubit. The overall transformation is equivalent to the two controlled-correction gates. The total number of controlled-NOT gates is N_Λ.

Now, we present another protocol for the controlled-σ gate. The circuit is shown in Fig. 3(a), which is formed of three parts: gates transforming a general Pauli operator σ into an X-product Pauli operator $\tilde{\sigma}$, gates transforming $\tilde{\sigma}$ into the single-qubit Pauli operator X_1 on qubit 1, and the controlled-NOT gate $\Lambda_{0,1}$ on the ancillary qubit and qubit 1. On the linear network, we assume that qubit 1 is next to the ancillary qubit. On the all-to-all network, we can label any qubit as qubit 1; without loss of generality, we assume that $x_1 \vee z_1 = 1$. In this protocol, there is only one instead

of N_Λ gates on the ancillary qubit. Because the outcome is obtained by measuring the ancillary qubit, applying fewer gates on the ancillary qubit potentially reduces the impact of errors. Replacing $\Lambda_{0,1}$ with the circuit in Fig. 3(b), we can realize the controlled-$e^{-i\phi\sigma}$ gate.

To transform $\tilde{\sigma}$ into X_1, we look for a transformation $\tilde{\Lambda}$ that satisfies $\sigma = \tilde{\Lambda}^\dagger X_1\tilde{\Lambda}$. On the all-to-all network, we take $\tilde{\Lambda} = \prod_{a=2}^n \Lambda_{1,a}^{x_a\vee z_a}$ and the number of controlled-NOT gates for each $\tilde{\Lambda}$ is $N_\Lambda - 1$. On the linear network, we take

$$\tilde{\Lambda} = \Lambda_{1,2}\Lambda_{2,1}^{x_1\downarrow z_1}\Lambda_{2,3}\Lambda_{3,2}^{x_2\downarrow z_2}\cdots\Lambda_{n'-1,n'}\Lambda_{n',n'-1}^{x_{n'-1}\downarrow z_{n'-1}}, \quad (54)$$

where $x_a \downarrow z_a = (1-x_a)(1-z_a)$ and $n' = \max\{a\,|\,x_a = 1\}$. On the linear network, the number of controlled-NOT gates for each $\tilde{\Lambda}$ is $(n'-1) + \sum_{a=1}^{n'-1} x_a \downarrow z_a \leq 2n - 2$.

The maximum number of controlled-NOT gates for implementing the two controlled-correction gates in each time step is summarized as follows. On the all-to-all network, the maximum gate number is n for POE formulas, which becomes $2(n-1) + 1 = 2n - 1$ to reduce gates on the ancillary qubit and $2[2(n-1)+2] = 4n$ for LOR formulas. On the linear network, the maximum gate number is $2(2n-2) + 1 = 4n - 3$ for POE formulas and $2[2(2n-2)+2] = 8n - 4$ for LOR formulas. In the MCQC algorithm, the controlled-correction gates are randomly selected and the gate number could be much smaller than its maximum value. For example, for the POE formula, the Pauli operator of the zeroth-order term in the expansion is the identity.
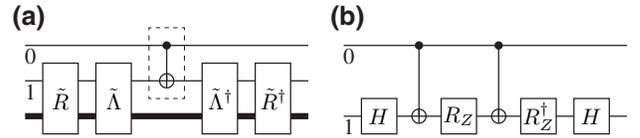
FIG. 3. (a) The circuit of the controlled-σ gate. Qubit 0 is the ancillary qubit. (b) The circuit of the controlled-$e^{-i\phi X_1}$ gate. Replacing the controlled-NOT gate in the dashed box with the controlled-$e^{-i\phi X_1}$ gate, we have the circuit of the controlled-$e^{-i\phi\sigma}$ gate. The single-qubit phase gate $R_Z = e^{i(\phi/2)Z}$.

VII. OPTIMAL DISTRIBUTION

In this section, we derive the optimal distribution of **s** that minimizes the variance in Monte Carlo. Using the protocols in Sec. VI to evaluate $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$, we prove that taking the distribution in Eq. (10) and $M_s = 1$ is optimal and that the minimum variance is given by Eq. (16).

A quantum circuit usually has random measurement outcomes; therefore, the outputs of quantum computing $a_{R,\mathbf{s}}$ and $a_{I,\mathbf{s}}$ are random variables. We suppose that $a_{\nu,\mathbf{s}}$ ($\nu = R, I$) takes the value a_i with the probability $P_{\nu,\mathbf{s},i}$ in the quantum computing; then, its expected value is $E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}} = \sum_i P_{\nu,\mathbf{s},i} a_i$. Here, $E\left[\bullet\right]_{\mathrm{QC}}$ denotes the mean taken over quantum computing runs for the specific **s** (each run returns an output evaluated using M_s shots) and $E\left[\bullet\right]$ without the subscript "QC" denotes the mean taken over both **s** and quantum computing runs. Using the protocols in Sec. VI, $a_{R,\mathbf{s}}$ and $a_{I,\mathbf{s}}$ are unbiased estimators of $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$, i.e.,

$$e^{i\theta_\mathbf{s}} \langle \psi_f | O_\mathbf{s} | \psi_i \rangle = E\left[a_{R,\mathbf{s}}\right]_{\mathrm{QC}} + i E\left[a_{I,\mathbf{s}}\right]_{\mathrm{QC}}. \qquad (55)$$

Let A_R and A_I be the real and imaginary parts of $\langle \psi_f | e^{iHt} O e^{-iHt} | \psi_i \rangle$, respectively. In the QCMC algorithm, we evaluate the summation $A_\nu = \sum_\mathbf{s} c_\mathbf{s} E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}}$ using the Monte Carlo method, where $c_\mathbf{s} = \left| \prod_{i=1}^N c(s_i) c(s_i')^* \right|$. Given any probability distribution $P(\mathbf{s})$, we have

$$A_\nu = \sum_\mathbf{s} P(\mathbf{s}) \frac{c_\mathbf{s} E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}}}{P(\mathbf{s})} = E\left[\frac{c_\mathbf{s} a_{\nu,\mathbf{s}}}{P(\mathbf{s})}\right]. \qquad (56)$$

Therefore, we can estimate A_ν by sampling **s** according to the distribution $P(\mathbf{s})$ and compute the empirical mean of $c_\mathbf{s} a_{\nu,\mathbf{s}} / P(\mathbf{s})$. The variance of the estimator $\hat{A}_\nu$ with N_s samples is

$$\begin{aligned} \mathrm{Var}\left(\hat{A}_\nu\right) &= \frac{1}{N_s} \mathrm{Var}\left[\frac{c_\mathbf{s} a_{\nu,\mathbf{s}}}{P(\mathbf{s})}\right] \\ &= \frac{1}{N_s} \sum_\mathbf{s} \frac{c_\mathbf{s}^2 \alpha_{\nu,\mathbf{s}}^2}{P(\mathbf{s})} - \frac{A_\nu^2}{N_s}, \end{aligned} \qquad (57)$$

where $\alpha_{\nu,\mathbf{s}} = \sqrt{E\left[a_{\nu,\mathbf{s}}^2\right]_{\mathrm{QC}}}$. The optimal distribution that minimizes the variance is $P(\mathbf{s}) \propto |c_\mathbf{s}| \alpha_{\nu,\mathbf{s}}$ and the minimum variance is

$$\mathrm{Var}\left(\hat{A}_\nu\right) = \frac{1}{N_s}\left(\sum_\mathbf{s} |c_\mathbf{s}| \alpha_{\nu,\mathbf{s}}\right)^2 - \frac{A_\nu^2}{N_s}. \qquad (58)$$

Now, we consider that $a_{\nu,\mathbf{s}}$ is obtained by taking the empirical mean of M_s binary numbers. Each binary number takes ± 1 corresponding to the measurement outcome of

the ancillary qubit (see Algorithm 4). Then, $a_{\nu,\mathbf{s}}$ follows the binomial distribution and

$$\alpha_{\nu,\mathbf{s}} = \sqrt{\frac{1 + (M_s - 1) E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}}^2}{M_s}}. \qquad (59)$$

Let M_{tot} be the total number of circuit shots; we have $N_s = M_{\mathrm{tot}}/M_s$. Substituting $\alpha_{\nu,\mathbf{s}}$ and N_s into Eq. (58), we obtain the variance as a function of M_s. Taking M_s as a continuous variable, we find that the derivative of the variance with respect to M_s is always positive when $M_s \geq 1$. Therefore, the variance is minimized at $M_s = 1$. When $M_s = 1$, we have $\alpha_{\nu,\mathbf{s}} = 1$ and the optimal distribution is $P(\mathbf{s}) \propto |c_\mathbf{s}|$. Accordingly, the minimum variance is

$$\mathrm{Var}\left(\hat{A}_\nu\right) = \frac{1}{M_{\mathrm{tot}}}\left(\sum_\mathbf{s} |c_\mathbf{s}|\right)^2 - \frac{A_\nu^2}{M_{\mathrm{tot}}}. \qquad (60)$$

With $\mathrm{Var}\left(\hat{A}\right) = \mathrm{Var}\left(\hat{A}_R\right) + \mathrm{Var}\left(\hat{A}_I\right)$, we obtain the minimum total variance in Eq. (16). Here, we assume that the total number of shots for each of the real and imaginary parts is M_{tot}.

We remark that the optimal distribution is obtained by assuming the empirical mean estimator for $a_{\nu,\mathbf{s}}$. If we have prior knowledge of the $a_{\nu,\mathbf{s}}$ distribution, we can use other estimators such as the Bayes estimator to reduce the variance. In the extreme case, suppose that $E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}}$ is known: the optimal distribution is $P(\mathbf{s}) \propto |c_\mathbf{s} E\left[a_{\nu,\mathbf{s}}\right]_{\mathrm{QC}}|$ instead of $P(\mathbf{s}) \propto |c_\mathbf{s}| \alpha_{\nu,\mathbf{s}}$ (note that in this case, we do not even need the quantum computer).

VIII. QUANTUM ERROR MITIGATION

Many quantum error-mitigation protocols can be classified into three categories. In the first category, with knowledge of the error model, we compensate the effect of errors by using approaches such as error extrapolation and probabilistic error cancellation (i.e., quasiprobability decomposition) [48,57,59]. In the second category, data from quantum circuits are processed according to constraints on the quantum state. The protocols in this category include, for example, symmetry-based postselection [67,68] and purification [69–71]. There are also protocols, e.g., subspace expansion [58], introduced for specific algorithms, which belong to the third category.

In this section, we first discuss the application of quasiprobability decomposition in QCMC and then we show that the forward-backward circuit in Fig. 2(b) provides inherent error mitigation based on constraints on the state. The error mitigation increases the variance in Monte Carlo. On a noisy quantum computer, we need to choose an optimal time-step size Δt to minimize the variance. Eventually, the variance is determined by the error rate, which is discussed in Sec. IX.

A. Quasiprobability decomposition

In the quasiprobability decomposition, an error-free quantum operation is expressed as a linear combination of noisy operations. Let $\mathcal{G}^{\text{ef}} = [U]$ and $\mathcal{G}_i$ be the error-free operation and noisy operations, respectively. The quasiprobability decomposition is in the form

$$\mathcal{G}^{\text{ef}} = \sum_i q_i \mathcal{G}_i, \qquad (61)$$

where q_i are real coefficients, i.e., quasiprobabilities. Here, U is a unitary quantum gate, $[U](\bullet) = U \bullet U^{\dagger}$ is the trace-preserving completely positive map of the gate and $\mathcal{G}_i$ are operations that can actually be implemented on the noisy quantum computer. Similar decompositions can be applied to the initial state and measurement.

We take the Pauli-error model as an example. Note that a general error model can be converted into the Pauli-error model by using Pauli twirling [72]. In the Pauli-error model, the noisy operation of a two-qubit gate U reads $\mathcal{G} = \mathcal{N}[U]$, where the noise map is

$$\mathcal{N} = (1-p)[I \otimes I] + \sum_{\sigma \in \{I,X,Y,Z\}^{\otimes 2} \setminus \{I \otimes I\}} p_{\sigma}[\sigma], \qquad (62)$$

$p_{\sigma} \ll 1$ is the rate of Pauli error σ, and $p = \sum_{\sigma \neq I \otimes I} p_{\sigma}$ is the total error rate. The inverse map of $\mathcal{N}$ is also in the Pauli-operation summation form, i.e.,

$$\mathcal{N}^{-1} = \sum_{\sigma \in \{I,X,Y,Z\}^{\otimes 2}} q_{\sigma}[\sigma], \qquad (63)$$

and we can solve coefficients q_{σ} numerically. Without a general analytically expression of q_{σ}, it is sufficient for us to consider the first-order expansion in order to discuss the impact on variance. To the first order, we have

$$q_{I \otimes I} = 1 + p + O(p^2), \qquad (64)$$

$$q_{\sigma \neq I \otimes I} = -p_{\sigma} + O(p^2). \qquad (65)$$

Given the inverse map, the quasiprobability decomposition of gate U is

$$\mathcal{G}^{\text{ef}} = \sum_{\sigma \in \{I,X,Y,Z\}^{\otimes 2}} q_{\sigma}[\sigma]\mathcal{G}. \qquad (66)$$

Assuming that errors in single-qubit gates are negligible, the composite operation $[\sigma]\mathcal{G}$ can be implemented on the noisy quantum computer by adding a Pauli gate σ after the noisy two-qubit gate $\mathcal{G}$. We note that the assumption of negligible errors in single-qubit gates is not necessary for the quasiprobability decomposition.

Now, we apply the quasiprobability decomposition to a quantum circuit. In QCMC, using protocols in Sec. VI,

only the ancillary qubit is measured. We can adjust the measurement basis using the gate B in Fig. 2; therefore, without loss of generality, we focus on the observable Z_a in the error mitigation. Given a quantum circuit formed of many elementary gates, the mean of Z_a reads

$$\langle Z_a \rangle = \text{Tr}\left[Z_a \mathcal{G}_{N_G} \cdots \mathcal{G}_1(\rho) \right], \qquad (67)$$

where $\rho = |0\rangle\langle 0|^{\otimes n}$ is the initial state of the quantum circuit and N_G is the number of gates. Suppose that the quasiprobability decomposition of each gate is $\mathcal{G}_j^{\text{ef}} = \sum_i q_{j,i} \mathcal{G}_{j,i}$. The error-free mean is

$$\langle Z_a \rangle^{\text{ef}} = \sum_{i_1,\ldots,i_{N_G}} \left(\prod_{j=1}^{N_G} q_{j,i} \right) \text{Tr}\left[Z_a \mathcal{G}_{N_G, i_{N_G}} \cdots \mathcal{G}_{1, i_1}(\rho) \right]. \qquad (68)$$

Each term in the summation is the mean of Z_a in a circuit modified from the original one. We remark that errors in the initial state and the final measurement can be corrected in a similar way.

We evaluate the decomposition formula in Eq. (68) using the Monte Carlo summation method by sampling random noisy circuits; therefore, such an error-mitigation protocol is called probabilistic error cancellation. Similar to QCMC, the sampling of random circuits increases the variance by a factor of C_E^2, where $C_E = \prod_{j=1}^{N_G} \left(\sum_i |q_{j,i}| \right)$. According to the Pauli-error model, we have $C_E = \prod_{j=1}^{N_G} [1 + 2p_j + O(p_j^2)]$, where p_j is the error rate of the jth gate. We find that the factor C_E increases with the number of noisy gates. Therefore, the circuit with fewer gates, i.e., the compact circuit in Fig. 2(a), is preferred.

In previous discussions, we have assumed that errors in different gates are not correlated. To deal with correlations, we need to introduce a general form of the quasiprobability decomposition,

$$\langle Z_a \rangle_{\mathbf{C}_0}^{\text{ef}} = \sum_k q_k \langle Z_a \rangle_{\mathbf{C}_k}, \qquad (69)$$

where $\langle Z_a \rangle_{\mathbf{C}_k}$ is the mean of Z_a in the circuit $\mathbf{C}_k$, $\mathbf{C}_0$ is the original circuit, and the $\mathbf{C}_{k \neq 0}$ are modified circuits. Modified circuits generated by adding single-qubit operations to the original circuit are usually sufficient for the existence of the decomposition formula. Without correlations, we can work out quasiprobabilities using gate-set tomography [59]; with correlations, we can determine quasiprobabilities using data of Clifford circuits, i.e., Clifford sampling [73,74]. Given the quasiprobability decomposition formulas, we can evaluate $\langle \psi_f | O_{\mathbf{s}} | \psi_i \rangle$ with error mitigation following the procedure in Algorithm 5.

B. Inherent error mitigation by postselection

As shown in verified quantum phase estimation [75] and dual-state purification [76], a quantum circuit with the

---

Input $\boldsymbol{h}, \boldsymbol{\sigma}, N, \Delta t, \boldsymbol{W}, \theta, M_s, \nu$.

1: Compose the circuit $\boldsymbol{C}_0$ according to input parameters, in which $\nu = \text{R}, \text{I}$ (real or imaginary) determines the basis adjusting gate B in Fig. 2.

2: Work out the decomposition formula in Eq. (69).

3: $C_E \leftarrow \sum_k |q_k|$

4: $a_\nu \leftarrow 0$

5: **for** $i = 1$ to M_s **do**

6:      Choose k with the probability $|q_k|/C_E$.

7:      Implement the circuit $\boldsymbol{C}_k$, measure Z_a and collect the outcome $\mu = \pm 1$.

8:      $a_\nu \leftarrow a_\nu + C_E \mu$

9: $a_\nu \leftarrow a_\nu / M_s$

10: Output a_ν.

Algorithm 5. Quantum-circuit evaluation with error mitigation.

forward-backward structure incorporating postselection is robust to errors. For the postselection, we measure the n qubits representing the system in addition to the ancillary qubit [see Fig. 2(b)]. We only select the state when the measurement outcome is $|0\rangle^{\otimes n}$, which transforms the final state in Eq. (52) into

$$|\Phi'\rangle = \frac{|0\rangle_a + \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle |1\rangle_a}{\sqrt{1 + \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2}} \otimes |0\rangle^{\otimes n}. \qquad (70)$$

Measuring the ancillary qubit in the state after postselection, we have

$$\langle X_a \rangle_0 = \frac{2\text{Re}\left(\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right)}{1 + \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2}, \qquad (71\text{a})$$

$$\langle Y_a \rangle_0 = \frac{2\text{Im}\left(\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right)}{1 + \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2}, \qquad (71\text{b})$$

$$\langle Z_a \rangle_0 = \frac{1 - \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2}{1 + \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2}, \qquad (71\text{c})$$

where $\langle \bullet \rangle_0$ denotes the mean conditioned on the outcome $|0\rangle^{\otimes n}$. Solving the equations, we obtain

$$\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle = \frac{\langle X_a \rangle_0 + i\langle Y_a \rangle_0}{1 + \langle Z_a \rangle_0}. \qquad (72)$$

The postselection forces most of qubits into a pure state, which eliminates errors that transform $|0\rangle^{\otimes n}$ into orthogonal states. In addition to postselection, we can purify the ancillary qubit as follows. According to Eq. (70), the state of the ancillary qubit is a pure state when the quantum circuit is error free. In the tomography purification, we implement the state tomography to the ancillary qubit and compute the eigenstate with the largest eigenvalue of the reconstructed reduced density matrix [76]. Using the

eigenstate to compute the three means $\langle \bullet \rangle_0$, we can make sure that the final result is obtained from a pure state. In Sec. 1, we demonstrate that the inherent error mitigation can significantly reduce the error in QCMC.

Now, we have two protocols using the circuit in Fig. 2(b) to evaluate $\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle$. In the protocol without postselection (see Sec. VI B), the estimator of $\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle$ is unbiased and it is optimal to take $M_s = 1$. In the protocol with postselection, the estimator is biased due to the denominator in Eq. (72); i.e., the mean of estimates is not exactly $\langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle$ when M_s is finite. Therefore, for the postselection protocol, it is necessary to choose a large M_s to evaluate each $\langle \bullet \rangle_0$ (such that the bias is small) in order to obtain an accurate final result of the transition amplitude.

The inherent error mitigation increases the variance of QCMC. When the circuit is error free, the postselection succeeds with the probability

$$P_S = \frac{1}{2}\left(1 + \left| \langle \psi_f |O_{\mathbf{s}}|\psi_i\rangle \right|^2 \right) \geq \frac{1}{2}. \qquad (73)$$

If the circuit is implemented for M_s shots, only $P_S M_s$ shots generate effective data on average. When the circuit is noisy, errors transform $|0\rangle^{\otimes n}$ into orthogonal states, which reduces the success rate. Therefore, the number of effective shots decreases with the error rate and the gate number, which causes an enlarged variance.

1. Numerical demonstration

To demonstrate the inherent error mitigation, we consider the one-dimensional Fermi-Hubbard model and numerically simulate the noisy quantum computing on a classical computer. The Hamiltonian reads

$$H = -J \sum_{i=1}^{N_L-1} \sum_{s=\uparrow,\downarrow} \left(c_{i,s}^\dagger c_{i+1,s} + c_{i+1,s}^\dagger c_{i,s} \right)$$
$$+ U \sum_i c_{i,\uparrow}^\dagger c_{i,\uparrow} c_{i,\downarrow}^\dagger c_{i,\downarrow}, \qquad (74)$$

where $N_L = 3$ is the number of sites and $c_{i,s}$ is the annihilation operator for the fermion with spin s on the ith site. This model can be encoded into $2N_L$ qubits using the Jordan-Wigner transformation.

We use the first-order Lie-Trotter-Suzuki product formula and a corresponding summation formula to simulate the real-time evolution. The initial state is $|\psi_i\rangle = c_{1,\uparrow}^\dagger c_{2,\downarrow}^\dagger c_{3,\uparrow}^\dagger |\text{Vac}\rangle$, where $|\text{Vac}\rangle$ is the vacuum state and we take $|\psi_f\rangle = |\psi_i\rangle$. The observable is $O = 2c_{3,\uparrow}^\dagger c_{3,\uparrow} - \mathbb{1}$ and the simulation is to compute $\langle O \rangle = \langle \psi_i |e^{iHt} O e^{-iHt}|\psi_i\rangle$. To minimize the variance of QCMC, we first expand the

correction operator using Pauli operators, i.e.,

$$V_1 = \sum_{\sigma \in \mathbf{P}_n} (a_\sigma - ib_\sigma)\sigma, \tag{75}$$

where a_σ and b_σ are real and

$$a_\sigma - ib_\sigma = 2^{-n}\mathrm{Tr}\left[\sigma e^{-iH\Delta t}S_1(\Delta t)^\dagger\right]. \tag{76}$$

We have $a_\mathbb{1} > 0$ and $b_\mathbb{1} = 0$. Then, we take the summation formula

$$e^{-iH\Delta t} = \sum_{\sigma \in \mathbf{P}_n \backslash \{\mathbb{1}\}} \left[a_\sigma \sigma + \beta_\sigma e^{-i\mathrm{sgn}(b_\sigma)\phi\sigma}\right]S_1, \tag{77}$$

where $\phi = \arctan(a_\mathbb{1}^{-1}\sum_\sigma |b_\sigma|)$ and $\beta_\sigma = |b_\sigma|/\sin\phi$.

Using the forward-backward circuit for error mitigation, we find that the impact of machine errors can be
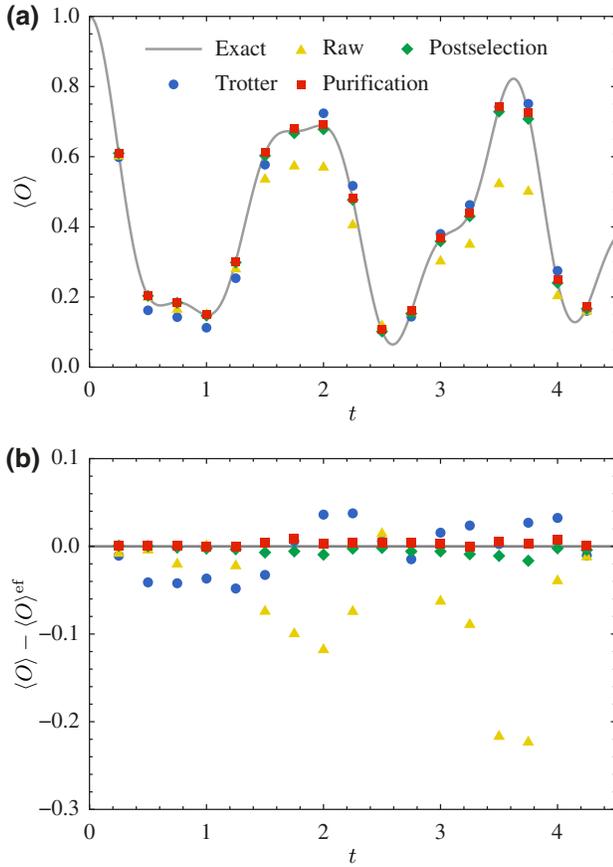
FIG. 4. (a) The observable $\langle O \rangle$ as a function of the time t. In the simulation, we take $J = 2$, $U = 4$ and $\Delta t = 0.05$. The raw data are obtained using the forward-backward circuit without error mitigation, according to Eq. (53). With postselection, the observable is computed according to Eq. (72). Tomography purification is used to further reduce the error. As a comparison, the Lie-Trotter-Suzuki product formula is evaluated without machine errors. (b) The error $\langle O \rangle - \langle O \rangle^{\mathrm{ef}}$ in the observable. Here, $\langle O \rangle^{\mathrm{ef}}$ denotes the exact value.

significantly suppressed, as shown in Fig. 4. We model the noise in quantum computing using the depolarizing error model. For a controlled-NOT gate, the noise map is given by Eq. (62) with parameters $p_\sigma = p/15$. We neglect errors in the initialization, single-qubit gates, and measurement. In the numerical simulation, we take the error rate per gate $p = 0.03\%$. The number of controlled-NOT gates for each S_1 is 14 and the simulation involves at most 85 time steps, i.e., the total number of controlled-NOT gates is above 2380. Therefore, the maximum total error rate is above 71.4%. After the error mitigation, we find that the overall accuracy of the summation formula taking $N_s = 10\,000$ samples is higher than the product formula without machine errors.

IX. QUANTUM COMPUTING VERSUS CLASSICAL COMPUTING

Sampling noise is the main source of error in the QCMC algorithm. The Monte Carlo variance increases exponentially with the evolution time as approximately $(1/N_s)C_A^{4N} = (1/N_s)e^{4t\Delta t^{-1}\ln C_A}$. As summarized in Table I, $C_A = 1 + \xi\Delta t^k + O(\Delta t^{k+1})$, where ξ is a constant depending on the Hamiltonian. When $k > 1$, by taking $\Delta t = (\delta/4t\xi)^{1/(k-1)}$, we can reduce the factor to $C_A^{4N} = e^{\delta + O\left(\delta^{k/(k-1)}\right)}$ for any small δ. We note that $k > 1$ in lth-order POE formulas with $l > 0$ and all LOR formulas. For the zeroth-order POE formula, because $C_A = e^{h_{\mathrm{tot}}\Delta t}$ (i.e., $k = 1$), we have $C_A^{4N} = e^{4h_{\mathrm{tot}}t}$ for all Δt.

It is widely believed that a classical computer cannot simulate the time evolution of general quantum many-body systems at a polynomial cost, which is one of main motivations for quantum computing [1]. The QCMC algorithm with the zeroth-order POE formula is equivalent to a classical algorithm, i.e., Green's function Monte Carlo taking the computational basis, for a large class of Hamiltonians (see Sec. C). In this classical algorithm, the variance increases exponentially with the evolution time and system size, i.e., approximately $(1/N_s)e^{4h_{\mathrm{tot}}t}$, whatever Δt we choose. Here, t is the evolution time and h_{tot} increases with the system size. The variance is up to minimization, e.g., changing the Hilbert-space basis [23] and optimizing the method for generating samples. Nevertheless, the existence of a generic approach that reduces the exponential scaling to polynomial one is unlikely [2].

On a fault-tolerant quantum computer, we can simulate the time evolution of quantum many-body systems at a polynomial cost. By taking a small Δt, we can reduce the factor C_A^{4N} to a satisfactory level and Δt scales polynomially with t and h_{tot}. Therefore, the number of times steps $N = t/\Delta t$, i.e., the circuit depth, scales polynomially with t and h_{tot}.

To demonstrate the impact on the sign problem in QMC incorporating quantum computing, we simulate the

real-time evolution of two models, the Fermi-Hubbard model and the Heisenberg model. We use two formulas in the simulation of each model, the zeroth-order POE and first-order LOR formulas. The zeroth-order POE formula corresponds to a classical QMC algorithm. Because the zeroth-order POE formula only includes products of Pauli operators, we can efficiently evaluate it on a classical computer even when the system size is large. The first-order LOR formula includes products of non-Pauli unitary operators, e.g., the product in Eq. (17). With a quantum computer, we can efficiently evaluate these non-Pauli products when the system is large. For the purpose of comparing the sign problem in two formulas, we evaluate both formulas on a classical computer, given that the system size is up to six qubits. The phase average $\langle e^{i\theta_q}\rangle$ is used to indicate the sign problem, where $e^{i\theta_q}$ denotes the phase of $e^{i\theta_s}\langle\Psi_f|O_s|\Psi_i\rangle$. We find that the sign problem is significant in the zeroth-order POE formula, i.e., $\langle e^{i\theta_q}\rangle$ converges to zero rapidly with the evolution time (see Figs. 5 and 6). As a result, the estimation of an observable has a large variance. In comparison, the sign problem is mild in the first-order LOR formula, i.e., $\langle e^{i\theta_q}\rangle$ is finite. With the sign problem mitigated, the simulation using the first-order LOR formula is accurate for a much longer evolution time compared with the zeroth-order POE formula.

On a noisy quantum computer, the cost of simulating quantum many-body systems increases exponentially with the evolution time and system size and the rate of increase decreases with the error rate. Using the quasiprobability decomposition to mitigate errors, the error mitigation enlarges the variance. The variance taking into account quantum error mitigation is approximately $(1/N_s)C_A^{4N}C_E^2$. We consider the compact circuit in Fig. 2(a). Let ϵ be the error rate of one elementary product S_1 (we assume that S_1' has the same error rate), let g be the number of S_1 and S_1' products per time step, and let $\eta\epsilon$ be the average error rate of the controlled-correction gates in one time step. The total error rate of one time step is approximately $(g+\eta)\epsilon$. Here, $g = 0, 1, 2$ for zeroth-, first-, and second-order formulas, respectively. Suppose that the total error rate of other operations [which are out of the bracket in Fig. 2(a)] is ϵ' and the factor due to error mitigation is $C_E \simeq (1 + 2\epsilon')[1 + 2(g+\eta)\epsilon]^N$, according to the Pauli-error model. Then, we can express the variance in the form

$$\frac{1}{N_s}C_A^{4N}C_E^2 \simeq \frac{1}{N_s}(1+2\epsilon')^2 e^{4\gamma h_{\text{tot}}t}, \qquad (78)$$

where

$$\gamma = \frac{1}{h_{\text{tot}}\Delta t}\ln\left[C_A\sqrt{1+2(g+\eta)\epsilon}\right]. \qquad (79)$$

We find that the rate γ decreases with the error rate.

Given the error rate of the noisy quantum computer, we choose the time-step size Δt to minimize the rate γ. Taking
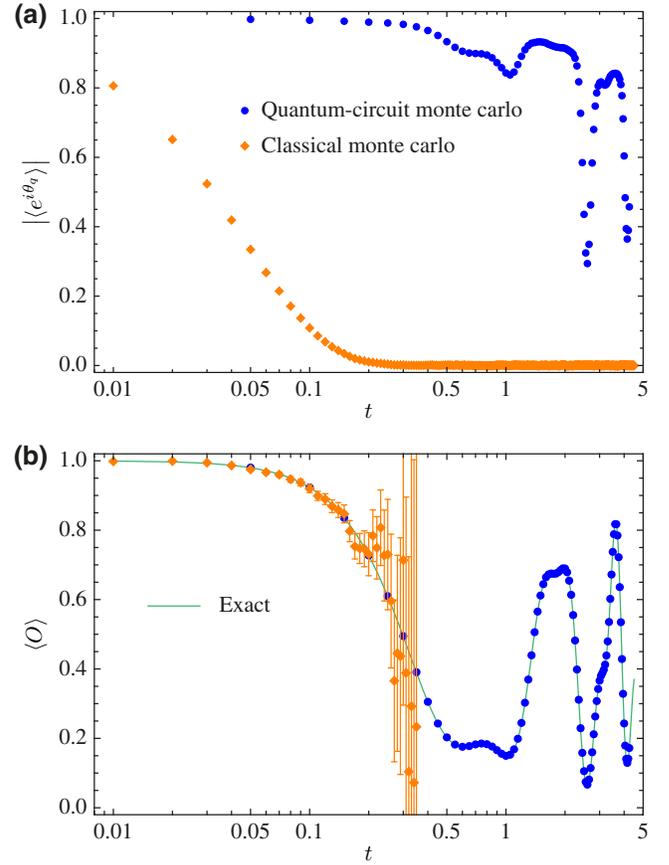
FIG. 5. (a) The phase average and (b) the expected value of the observable $O = 2c_{3,\uparrow}^\dagger c_{3,\uparrow} - \mathbb{1}$ in the Monte Carlo simulation of the Fermi-Hubbard model. The Hamiltonian is given in Eq. (74), in which we take $J = 2$ and $U = 4$. The simulation is to compute $\langle O\rangle = \langle\psi_i|e^{iHt}Oe^{-iHt}|\psi_i\rangle$, where $|\psi_i\rangle = c_{1,\uparrow}^\dagger c_{2,\downarrow}^\dagger c_{3,\uparrow}^\dagger|\text{Vac}\rangle$. In the quantum-circuit Monte Carlo (first-order leading-order-rotation formula), we take $\Delta t = 0.05$ and $N_s = 10\,000$. In the classical Monte Carlo (zeroth-order Pauli-operator-expansion formula), we take $\Delta t = 0.01$ and $N_s = 100\,000$ [77]. The samples are generated according to the distribution in Eq. (10).

$C_A \simeq 1 + \xi\Delta t^k$, we find that the optimal step size is

$$\Delta t_{\text{opt}} \simeq \left[\frac{(g+\eta)\epsilon}{(k-1)\xi}\right]^{1/k}, \qquad (80)$$

and the corresponding minimum rate is

$$\gamma_{\min} \simeq \frac{k\xi^{1/k}}{h_{\text{tot}}}\left[\frac{(g+\eta)\epsilon}{k-1}\right]^{(k-1)/k}. \qquad (81)$$

For the second-order LOR formula, $k = 6$, $g = 2$, and $\xi < h_{\text{tot}}^6/(2\times 18^2)$. In Fig. 7, we plot the minimum rate computed numerically using formulas of C_A in Table I. For the first- and second-order formulas, we take the upper bound of the simplified leading-order contribution. We find that

(a)

(b)

FIG. 7. The minimum rate of increase of the variance γ_{min}. ϵ is the error rate of one elementary Lie-Trotter-Suzuki product S_1. $\eta\epsilon$ is the average error rate of the controlled-correction gates in one time step. POE l denotes the lth-order Pauli-operator-expansion formula and LOR l denotes the lth-order leading-order-rotation formula.

sample size N_s by a factor of approximately 4×10^4 when $\epsilon = 0.1$ and approximately 4×10^6 when $\epsilon = 0.01$. The advantage of the quantum algorithm grows when the error rate decreases: fitting to the second-order LOR curve in Fig. 7(b), we have $\gamma \simeq 2.45\epsilon^{0.82}$.

We can optimize the quantum algorithm to reduce the variance in various ways. First, we can significantly reduce C_A for a Hamiltonian with only local interactions. C_A is greater than one because of the correction operator, which is used to compensate the difference between the exact time-evolution operator and the Lie-Trotter-Suzuki product. According to the Baker-Campbell-Hausdorff formula, this difference is a series of commutators. For local interactions, most of the commutators in low-order terms are zero. In this case, expanding the correction operator according to the Baker-Campbell-Hausdorff formula (instead of the direct Taylor expansion) can reduce C_A. Second, similar to the classical algorithm, with some knowledge of $\langle\psi_f|O_s|\psi_i\rangle$, we can optimize the distribution of generating samples to reduce the variance. Third, the variance due to quantum error mitigation can be reduced. In the error-mitigation protocol used to estimate γ, we correct all Pauli errors in the circuit, which is unnecessary. Because the ancillary qubit is measured to evaluate the transition amplitude, we only need to correct errors that affect the ancillary qubit. These errors can be identified and corrected by utilizing the learning-based approach of error mitigation [73].

X. CONCLUSIONS

In this paper, we propose a QMC algorithm that uses quantum computing as a subroutine, which allows the nonvariational quantum simulation to be implemented

(a)

(b)

FIG. 6. (a) The phase average and (b) the expected value of the observable $O = Z_3$ in the Monte Carlo simulation of the Heisenberg model. The Hamiltonian is $H = -J\sum_{i=1}^{N_S-1}(X_iX_{i+1} + Y_iY_{i+1} + Z_iZ_{i+1}) - h\sum_{i=1}^{N_S}Z_i$, where the number of spins is $N_S = 6$, and $J = h = 1$. The simulation is to compute $\langle O \rangle = \langle\psi_i|e^{iHt}Oe^{-iHt}|\psi_i\rangle$, where $|\psi_i\rangle = |010101\rangle$. In the quantum-circuit Monte Carlo (first-order leading-order-rotation formula), we take $\Delta t = 0.05$ and $N_s = 10\,000$. In the classical Monte Carlo (zeroth-order Pauli-operator-expansion formula), we take $\Delta t = 0.01$ and $N_s = 200\,000$ [77]. The samples are generated according to the distribution in Eq. (10).

the second-order LOR formula outperforms other formulas. Taking higher-order formulas does not further reduce γ for the given error rates.

Although the cost of the QCMC algorithm on a noisy quantum computer scales exponentially with the evolution time and system size in the same way as the classical algorithm, the quantum computing can accelerate QMC by reducing the variance. To achieve the computation accuracy δ, i.e., to reduce the variance to δ^2, we take $N_s \sim e^{4\gamma h_{tot}t}/\delta^2$. In the classical algorithm, $\gamma = 1$. In the quantum algorithm, taking the minimum value for $\eta = 1$ [see Fig. 7(b)], we have $\gamma \simeq 0.34$ when the error rate per elementary product is $\epsilon = 0.1$ and $\gamma \simeq 0.058$ when $\epsilon = 0.01$. For $h_{tot}t = 4$, the quantum algorithm reduces the

with noisy intermediate-scale quantum hardware. In our algorithm, we use exact summation formulas to express the time-evolution operator. We optimize these summation formulas and quantum circuits to minimize the Monte Carlo variance and circuit depth. The optimal distribution of generating samples in Monte Carlo is derived in the circumstances of probabilistic evaluation using quantum computing. On a noisy quantum computer, we can use probabilistic error cancellation or inherent error mitigation to eliminate machine errors. By choosing the parameter Δt, we can maximize the quantum speed-up given a finite error rate. This scheme illustrates a way of designing quantum algorithms with reduced circuit depth by using Monte Carlo techniques [78].

Our algorithm shows that a quantum computer without fault tolerance can speed up solving practical problems. In terms of algorithmic complexity, quantum computing has an advantage over classical computing in many computational tasks, in the fault-tolerance regime achieved with quantum error correction [79]. Even without error correction, a quantum device can perform tasks that are intractable for classical computers, such as sampling the output of a quantum circuit [80]. Our algorithm is to solve a practical problem, i.e., the nonvariational simulation of quantum many-body systems. We theoretically analyze the complexity of our algorithm, i.e., the circuit depth and sampling cost. The complexity is polynomial on a fault-tolerant quantum computer. On a noisy quantum computer, although the complexity is exponential due to the finite error rate, our algorithm can still outperform classical algorithms and speed up Monte Carlo calculations by substantially reducing the sign problem.

Note added.—Shortly after this work (the first version) posted on arXiv, a relevant paper was also made public [82], which reports a quantum algorithm for imaginary-time dynamics based on QMC.

APPENDIX A: LEADING-ORDER TERMS

Let $A_j \equiv -i h_j \sigma_j \Delta t$ and $A \equiv \sum_i A_i = -iH\Delta t$ for simplicity. The Taylor expansion of the time-evolution operator reads

$$e^{-iH\Delta t} = e^A = \mathbb{1} + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + O\left(\Delta t^4\right). \quad \text{(A1)}$$

We have

$$A^2 = \sum_{i<j}\left(A_iA_j + A_jA_i\right) + \sum_i A_i^2 \quad \text{(A2)}$$

and

$$A^3 = \sum_{i<j<k}\left(A_iA_jA_k + A_kA_jA_i + A_jA_iA_k + A_kA_iA_j\right.$$
$$\left. + A_iA_kA_j + A_jA_kA_i\right) + \sum_{i<j}\left(A_i^2A_j + A_jA_i^2 + A_iA_jA_i\right.$$
$$\left. + A_iA_j^2 + A_j^2A_i + A_jA_iA_j\right) + \sum_i A_i^3. \quad \text{(A3)}$$

1. First-order formula

According to the first-order formula, we have

$$S_1(\Delta t)^\dagger = e^{-A_1}\cdots e^{-A_M}$$

$$= \prod_{i=1}^{M}\left[\mathbb{1} - A_i + \frac{1}{2}A_i^2 - \frac{1}{6}A_i^3 + O\left(\Delta t^4\right)\right]$$

$$= \mathbb{1} - A + \frac{1}{2}A^{(2)} - \frac{1}{6}A^{(3)} + O\left(\Delta t^4\right), \quad \text{(A4)}$$

where

$$A^{(2)} = 2\sum_{i<j}A_iA_j + \sum_i A_i^2, \quad \text{(A5)}$$

$$A^{(3)} = 6\sum_{i<j<k}A_iA_jA_k + 3\sum_{i<j}\left(A_i^2A_j + A_iA_j^2\right) + \sum_i A_i^3. \quad \text{(A6)}$$

The correction operator is

$$V_1(\Delta t) = \mathbb{1} - A^2 + \frac{1}{2}\left(A^2 + A^{(2)}\right) + \frac{1}{2}A\left(A^{(2)} - A^2\right)$$
$$+ \frac{1}{6}\left(A^3 - A^{(3)}\right) + O\left(\Delta t^4\right). \quad \text{(A7)}$$

Then, we have

$$F_1^{(2)}(\Delta t) = -A^2 + \frac{1}{2}\left(A^2 + A^{(2)}\right) = \frac{1}{2}\left(A^{(2)} - A^2\right)$$

$$= \frac{1}{2}\sum_{i<j}\left(A_iA_j - A_jA_i\right) \quad \text{(A8)}$$

and

$$F_1^{(3)}(\Delta t) = \frac{1}{2}A\left(A^{(2)} - A^2\right) + \frac{1}{6}\left(A^3 - A^{(3)}\right)$$

$$= \frac{1}{2}\sum_{i<j<k}\left(A_iA_jA_k - A_kA_jA_i + A_jA_iA_k + A_kA_iA_j - A_iA_kA_j - A_jA_kA_i\right)$$

$$+ \frac{1}{2}\sum_{i<j}\left(A_i^2A_j - A_iA_jA_i - A_j^2A_i + A_jA_iA_j\right) + \frac{1}{6}\left(A^3 - A^{(3)}\right)$$

$$= \frac{1}{6}\sum_{i<j<k}\left(-2A_iA_jA_k - 2A_kA_jA_i + 4A_jA_iA_k + 4A_kA_iA_j - 2A_iA_kA_j - 2A_jA_kA_i\right)$$

$$+ \frac{1}{6}\sum_{i<j}\left(A_i^2A_j + A_jA_i^2 - 2A_iA_jA_i - 2A_iA_j^2 - 2A_j^2A_i + 4A_jA_iA_j\right). \tag{A9}$$

According to Eq. (A8), the contribution of $F_1^{(2)}(\Delta t)$ to the normalization factor is

$$\sum_{i<j}|h_ih_j|\Delta t^2 < \frac{1}{2}\left(\sum_i|h_i|\right)^2\Delta t^2. \tag{A10}$$

2. Second-order formula

We can write the second-order correction operator as

$$V_2(\Delta t) = V_1\left(-\frac{\Delta t}{2}\right)^\dagger V_1\left(\frac{\Delta t}{2}\right)$$

$$= \mathbb{1} + F_1^{(3)}\left(\frac{\Delta t}{2}\right) + F_1^{(3)}\left(-\frac{\Delta t}{2}\right)^\dagger + O(\Delta t^5). \tag{A11}$$

Then, we have

$$F_2^{(3)}(\Delta t) = F_1^{(3)}\left(\frac{\Delta t}{2}\right) + F_1^{(3)}\left(-\frac{\Delta t}{2}\right)^\dagger. \tag{A12}$$

Accordingly, the contribution of $F_2^{(3)}(\Delta t)$ to the normalization factor is

$$\frac{1}{3}\sum_{i<j<k}|h_ih_jh_k|\Delta t^3 + \frac{1}{12}\sum_{i<j}\left(|h_i^2h_j| + 2|h_ih_j^2|\right)\Delta t^3$$

$$< \frac{1}{18}\left(\sum_i|h_i|\right)^3\Delta t^3. \tag{A13}$$

APPENDIX B: FERMI-HUBBARD MODEL

The Hamiltonian of Fermi-Hubbard model reads

$$H_{\text{FH}} = -\sum_{i<j}J_{i,j}\sum_{s=\uparrow,\downarrow}\left(c_{i,s}^\dagger c_{j,s} + c_{j,s}^\dagger c_{i,s}\right)$$

$$+ U\sum_i\left(c_{i,\uparrow}^\dagger c_{i,\uparrow} - \frac{\mathbb{1}}{2}\right)\left(c_{i,\downarrow}^\dagger c_{i,\downarrow} - \frac{\mathbb{1}}{2}\right), \tag{B1}$$

where $c_{i,s}$ is the annihilation operator for the fermion with spin s on the ith site. Operators of fermions satisfy $\{c_{i,s}, c_{i',s'}\} = 0$ and $\{c_{i,s}, c_{i',s'}^\dagger\} = \delta_{i,i'}\delta_{s,s'}\mathbb{1}$. Here, we modify the original Fermi-Hubbard model by adding a uniform on-site potential $-(U/2)N$, which does not affect the time evolution if the initial state is an eigenstate of the total particle number operator $N = \sum_{i,s}c_{i,s}^\dagger c_{i,s}$. For a bipartite lattice, $J_{i,j} = 0$ for all $i + j \in$ Even, i.e., two sites are not coupled if their labels have the same parity.

To encode the Fermi-Hubbard model into qubits, we take the Jordan-Wigner transformation

$$c_{i,\uparrow} = \frac{Y_{2i-1} - iZ_{2i-1}}{2}\prod_{l<2i-1}X_l,$$

$$c_{i,\downarrow} = \frac{Y_{2i} - iZ_{2i}}{2}\prod_{l<2i}X_l, \tag{B2}$$

where X_a, Y_a, and Z_a are the Pauli operators of qubit a. The spin-$\uparrow$ and spin-$\downarrow$ on the ith site are encoded on qubits $(2i - 1)$ and $2i$, respectively. According to the Jordan-Wigner transformation, the qubit Hamiltonian of Fermi-Hubbard model is

$$H_{\text{FH}} = -\sum_{i<j}\frac{J_{i,j}}{2}\left(Y_{2i-1,2j-1} + Z_{2i-1,2j-1} + Y_{2i,2j} + Z_{2i,2j}\right)$$

$$+ \frac{U}{4}\sum_i X_{2i-1}X_{2i}, \tag{B3}$$

where

$$Y_{a,b} = Y_aY_b\prod_{a<l<b}X_l,$$

$$Z_{a,b} = Z_aZ_b\prod_{a<l<b}X_l. \tag{B4}$$

Each Pauli operator σ corresponds to an x binary string according to Eq. (47), and we define $\mathbf{x}(\sigma) \equiv (x_1, \ldots, x_n)$ as

the x binary string of the Pauli operator σ. A Hamiltonian does not have short-time interference if $\mathbf{x}(\sigma_1) \neq \mathbf{x}(\sigma_2)$ for any pair of Pauli-operator terms σ_1 and σ_2 in the Hamiltonian. We use $\mathbf{x}_{a,b}$ to denote the binary string for which $x_k = 0$ if $k < a$ or $k > b$ and $x_k = 1$ if $a \leq k \leq b$. Then,

$$\mathbf{x}\left(Y_{a,b}\right) = \mathbf{x}_{a,b},$$
$$\mathbf{x}\left(Z_{a,b}\right) = \mathbf{x}_{a+1,b-1}, \qquad \text{(B5)}$$
$$\mathbf{x}\left(X_{2i-1}X_{2i}\right) = \mathbf{x}_{2i-1,2i}.$$

We find that x strings of $Y_{a,b}$ and $Z_{a,b}$ terms in Eq. (B3) are all different from $X_{2i-1}X_{2i}$ terms: note that a and b have the same parity. The only question is whether $Y_{a,b}$ and $Z_{a',b'}$ have the same x string. If their x strings are the same, we must have $a = a' + 1$ and $b = b' - 1$. For a bipartite lattice, $(b - a)/2$ and $(b' - a')/2$ are both odd; however, $(b' - 1 - a' - 1)/2$ is even if $(b' - a')/2$ is odd. Therefore, x strings of $Y_{a,b}$ and $Z_{a',b'}$ are always different.

---

[1] R. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[2] M. Troyer and U.-J. Wiese, Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations, Phys. Rev. Lett. **94**, 170201 (2005).

[3] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation, Science **279**, 342 (1998).

[4] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[5] J. Preskill, Quantum computing in the NISQ era and beyond, ArXiv:1801.00862.

[6] N. Metropolis and S. Ulam, The Monte Carlo method, J. Am. Stat. Assoc. **44**, 335 (1949).

[7] J. Carlson, S. Gandolfi, F. Pederiva, S. C. Pieper, R. Schiavilla, K. E. Schmidt, and R. B. Wiringa, Quantum Monte Carlo methods for nuclear physics, Rev. Mod. Phys. **87**, 1067 (2015).

[8] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, Monte Carlo calculations of coupled boson-fermion systems. I, Phys. Rev. D **24**, 2278 (1981).

[9] D. Lee, Lattice simulations for few- and many-body systems, Prog. Part. Nucl. Phys. **63**, 117 (2009).

[10] H. G. Evertz, G. Lana, and M. Marcu, Cluster Algorithm for Vertex Models, Phys. Rev. Lett. **70**, 875 (1993).

[11] S. Bour, D. Lee, H.-W. Hammer, and U.-G. Meißner, *Ab initio* Lattice Results for Fermi Polarons in Two Dimensions, Phys. Rev. Lett. **115**, 185301 (2015).

[12] K. Van Houcke, E. Kozik, N. Prokof'ev, and B. Svistunov, Diagrammatic Monte Carlo, Phys. Proc. **6**, 95 (2010).

[13] K. Van Houcke, F. Werner, E. Kozik, N. Prokof'ev, B. Svistunov, M. J. H. Ku, A. T. Sommer, L. W. Cheuk, A. Schirotzek, and M. W. Zwierlein, Feynman diagrams versus Fermi-gas Feynman emulator, Nat. Phys. **8**, 366 (2012).

[14] G. Cohen, E. Gull, D. R. Reichman, and A. J. Millis, Taming the Dynamical Sign Problem in Real-Time Evolution of Quantum Many-Body Problems, Phys. Rev. Lett. **115**, 266802 (2015).

[15] C. Bertrand, S. Florens, O. Parcollet, and X. Waintal, Reconstructing Nonequilibrium Regimes of Quantum Many-Body Systems from the Analytical Structure of Perturbative Expansions, Phys. Rev. X **9**, 041008 (2019).

[16] J. Lomnitz-Adler, V. Pandharipande, and R. Smith, Monte Carlo calculations of triton and ⁴He nuclei with the Reid potential, Nucl. Phys. A **361**, 399 (1981).

[17] J. Hubbard, Electron correlations in narrow energy bands, Proc. R. Soc. London, A **276**, 238257 (1963).

[18] M. Takahashi, Half-filled Hubbard model at low temperature, J. Phys. C: Solid State Phys. **10**, 1289 (1977).

[19] B.-N. Lu, N. Li, S. Elhatisari, D. Lee, E. Epelbaum, and U.-G. Meißner, Essential elements for nuclear binding, Phys. Lett. B **797**, 134863 (2019).

[20] D. Lee, S. Bogner, B. A. Brown, S. Elhatisari, E. Epelbaum, H. Hergert, M. Hjorth-Jensen, H. Krebs, N. Li, B.-N. Lu, and U.-G. Meißner, Hidden spin-isospin exchange symmetry, ArXiv:2010.09420.

[21] B. J. Hammond, W. A. Lester, and P. J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry* (World Scientific, Singapore, 1994).

[22] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Quantum Monte Carlo simulations of solids, Rev. Mod. Phys. **73**, 33 (2001).

[23] D. Hangleiter, I. Roth, D. Nagaj, and J. Eisert, Easing the Monte Carlo sign problem, Sci. Adv. **6**, eabb8341 (2020).

[24] G. Parisi, On complex probabilities, Phys. Lett. B **131**, 393 (1983).

[25] J. R. Klauder, Stochastic quantization, Acta Phys. Austriaca Suppl. **25**, 251 (1983).

[26] M. Cristoforetti, F. DiRenzo, and L. Scorzato, New approach to the sign problem in quantum field theories: High density QCD on a Lefschetz thimble, Phys. Rev. D **86**, 074506 (2012).

[27] J.-L. Wynen, E. Berkowitz, S. Krieg, T. Luu, and J. Ostmeyer, Machine learning to alleviate Hubbard-model sign problems, Phys. Rev. B **103**, 125153 (2021).

[28] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. **270**, 359 (2007).

[29] N. Wiebe, D. Berry, P. Høyer, and B. C. Sanders, Higher order decompositions of ordered operator exponentials, J. Phys. A: Math. Theor. **43**, 065203 (2010).

[30] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, Phys. Rev. Lett. **114**, 090502 (2015).

[31] R. Meister, S. C. Benjamin, and E. T. Campbell, Tailoring term truncations for electronic structure calculations using a linear combination of unitaries, ArXiv:2007.11624.

[32] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[33] P. K. Faehrmann, M. Steudtner, R. Kueng, M. Kieferova, and J. Eisert, Randomizing multi-product formulas for improved Hamiltonian simulation, ArXiv:2101.07808.

[34] E. Campbell, Random Compiler for Fast Hamiltonian Simulation, Phys. Rev. Lett. **123**, 070503 (2019).

[35] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert, Dissipative Quantum Church-Turing Theorem, Phys. Rev. Lett. **107**, 120501 (2011).

[36] H. Wang, S. Ashhab, and F. Nori, Quantum algorithm for simulating the dynamics of an open quantum system, Phys. Rev. A **83**, 062317 (2011).

[37] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Quantum metropolis sampling, Nature **471**, 87 (2011).

[38] A. Riera, C. Gogolin, and J. Eisert, Thermalization in Nature and on a Quantum Computer, Phys. Rev. Lett. **108**, 080402 (2012).

[39] T. E. O'Brien, B. Tarasinski, and B. M. Terhal, Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments, New J. Phys. **21**, 023022 (2019).

[40] S. Lu, M. C. Bañuls, and J. I. Cirac, Algorithms for quantum simulation at finite energies, PRX Quantum **2**, 020321 (2021).

[41] F. Turro, V. Amitrano, P. Luchi, K. A. Wendt, J. L. DuBois, S. Quaglioni, and F. Pederiva, Imaginary time propagation on a quantum chip, ArXiv:2102.12260.

[42] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, PNAS **114**, 7555 (2017).

[43] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Phys. Rev. X **8**, 041015 (2018).

[44] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[45] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Hybrid Quantum-Classical Approach to Correlated Materials, Phys. Rev. X **6**, 031045 (2016).

[46] 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).

[47] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[48] Y. Li and S. C. Benjamin, Efficient Variational Quantum Simulator Incorporating Active Error Minimisation, Phys. Rev. X **7**, 021050 (2017).

[49] J. W. Z. Lau, T. Haug, L. C. Kwek, and K. Bharti, NISQ algorithm for Hamiltonian simulation via truncated Taylor series, ArXiv:2103.05500.

[50] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational ansatz-based quantum simulation of imaginary time evolution, npj Quantum Inf. **5**, 75 (2019).

[51] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. Kin-Lic Chan, Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution, Nat. Phys. **16**, 205 (2020).

[52] 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).

[53] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chem. Rev. **119**, 10856 (2019).

[54] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Rev. Mod. Phys. **92**, 015003 (2020).

[55] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Phys. Lett. A **146**, 319 (1990).

[56] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A **150**, 262 (1990).

[57] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[58] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states, Phys. Rev. A **95**, 042308 (2017).

[59] S. Endo, S. C. Benjamin, and Y. Li, Practical Quantum Error Mitigation for Near-Future Applications, Phys. Rev. X **8**, 031027 (2018).

[60] T. A. Lähde, T. Luu, D. Lee, U.-G. Meißner, E. Epelbaum, H. Krebs, and G. Rupak, Nuclear lattice simulations using symmetry-sign extrapolation, Eur. Phys. J. A **51**, 92 (2015).

[61] E. Epelbaum, H. Krebs, T. A. Lähde, D. Lee, U.-G. Meißner, and G. Rupak, *Ab initio* Calculation of the Spectrum and Structure of 16°, Phys. Rev. Lett. **112**, 102501 (2014).

[62] D. Frame, R. He, I. Ipsen, D. Lee, D. Lee, and E. Rrapaj, Eigenvector Continuation with Subspace Learning, Phys. Rev. Lett. **121**, 032501 (2018).

[63] S. König, A. Ekström, K. Hebeler, D. Lee, and A. Schwenk, Eigenvector continuation as an efficient and accurate emulator for uncertainty quantification, Phys. Lett. B **810**, 135814 (2020).

[64] A. Sarkar and D. Lee, Convergence of Eigenvector Continuation, Phys. Rev. Lett. **126**, 032501 (2021).

[65] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Quantum Comput. Quantum Inf. **305**, 53 (2002).

[66] D. Gottesman, The Heisenberg representation of quantum computers, ArXiv:quant-ph/9807006.

[67] S. McArdle, X. Yuan, and S. Benjamin, Error-Mitigated Digital Quantum Simulation, Phys. Rev. Lett. **122**, 180501 (2019).

[68] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, Low-cost error mitigation by symmetry verification, Phys. Rev. A **98**, 062339 (2018).

[69] B. Koczor, Exponential error suppression for near-term quantum devices, ArXiv:2011.05942.

[70] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, Virtual distillation for quantum error mitigation, ArXiv:2011.07064.

[71] P. Czarnik, A. Arrasmith, L. Cincio, and P. J. Coles, Qubit-efficient exponential suppression of errors, ArXiv:2102.06056.

[72] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Phys. Rev. A **94**, 052325 (2016).

[73] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, Learning-based quantum error mitigation, ArXiv:2005.07601.

[74] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, Error mitigation with Clifford quantum-circuit data, ArXiv:2005.10189.

[75] T. E. O'Brien, S. Polla, N. C. Rubin, W. J. Huggins, S. McArdle, S. Boixo, J. R. McClean, and R. Babbush, Error mitigation via verified phase estimation, ArXiv:2010.02538.

[76] M. Huo and Y. Li, Dual-state purification for practical quantum error mitigation, ArXiv:2105.01239.

[77] The variance of the zeroth-order POE formula is independent of Δt. Amplitudes $\langle \psi_f | O_\mathbf{s} | \psi_i \rangle$ in the classical Monte Carlo simulation take three values, 0 and ± 1. We choose sample numbers N_s in the classical Monte Carlo such that the number of nonzero amplitudes is larger than 10 000. In other words, even if we generate 10 000 samples according to the optimal distribution in Eq. (9), the variance is not smaller.

[78] P. A. M. Casares, R. Campos, and M. A. Martin-Delgado, QFold: Quantum walks and deep learning to solve protein folding, ArXiv:2101.10279.

[79] S. Bravyi, D. Gosset, R. König, and M. Tomamichel, Quantum advantage with noisy shallow circuits, Nat. Phys. **16**, 1040 (2020).

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

[81] T. Jones and S. Benjamin, QuESTlink—MATHEMATICA embiggened by a hardware-optimised quantum emulator, Quantum Sci. Technol. **5**, 034012 (2020).

[82] W. J. Huggins, B. A. O'Gorman, C. Neil, N. C. Rubin, P. Roushan, D. R. Reichman, R. Babbush, and J. Lee, Unbiasing fermionic quantum Monte Carlo with a quantum computer, ArXiv:2106.16235.