Accelerated quantum Monte Carlo with mitigated error on noisy quantum computer

Yongdan Yang,¹ Bing-Nan Lu,¹ and Ying Li¹,∗

¹Graduate 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 non-variational 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 quasi-probability 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 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 de-
veloped 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 parameterised 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 non-variational 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 minimise 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 $\Delta t$, at the cost of a moderately increased gate number per time step. For example, the second-order LOR formula converges as $O(\Delta t^2)$, 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 $\Delta t$, which must be sufficiently small to reduce the error. Exact summation formulas allow us to take a large $\Delta t$ (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 minimise the variance, i.e. the statistical error, by taking the optimal time step size. We obtain the minimised variance of QCMC in the form of approximately $e^{\gamma t}h_{tot}$, where $\gamma$ is the increasing rate of the variance, $h_{tot}$ characterises 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, rate of increase of variance has the upper bound $\gamma \simeq 2.45\epsilon^{0.82}$, where $\epsilon$ 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 $\epsilon$ 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 $\gamma_c$ [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 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 $4 \times 10^4$ when $h_{tot}t = 4$ 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 non-variational 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 eigen-
state. Second, the optimiser 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 organised 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 summarise the conclusions.

II. QUANTUM MONTE CARLO

Many applications of QMC can be formalised 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

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

(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:

$$\langle \psi_f | e^{iHt} O e^{-iHt} | \psi_i \rangle = \int_{r_0, \ldots, r_N, r_0', \ldots, r_N'} dr_0 \cdots dr_N dr_0' \cdots dr_N'$$

$$\langle \psi_f | \prod_{r} | r \rangle \langle r | e^{iH \Delta t} | r \rangle | r \rangle \langle r | \cdots \langle r_{N-1} | e^{iH \Delta t} | r \rangle | r \rangle \langle r_0 | \psi_i \rangle,$$

(2)

where $\{ | r \rangle \}$ 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.

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

(3)

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

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

$$\cdots \times A(s'_N, -\Delta t^* \cdots) O A(s_N, \Delta t) \cdots A(s_1, \Delta t) | \psi_i \rangle.$$  

(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 $m$-th-order term is $D(\xi_m, x_1, \ldots, x_m)$: the transition amplitude is a summation of integrals of the form

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

(5)

where $\xi_m$ is the index of the term and the $x$ 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 non-interacting 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 $q = | \psi_f | \cdots | \psi_i \rangle$ 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 strong a 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
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 $\Delta t$. 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 minimise the number of time steps $N$ by maximising $\Delta t$. Because of the minimised 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

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

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

$$O_s = U(s_1)^\dagger \cdots U(s_N)^\dagger O U(s_N) \cdots U(s_1). \quad (8)$$

One can realise a summation formula either by using a deterministic circuit [30–32] or sampling random circuits [33, 34]. To minimise 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 $\langle \psi_f | O_s | \psi_i \rangle$ on the quantum computer and then compute the path-integral summation using the Monte Carlo method on the classical computer. 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 quasi-probability decomposition (i.e. probabilistic error cancellation) [57, 59], each unitary circuit for evaluating $\langle \psi_f | O_s | \psi_i \rangle$ 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 quasi-probability decompositions are also exact, the sampling noise in Monte Carlo is the only source of error in our algorithm.

Sampling noise and normalisation 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 $s$ with a probability distribution $P(s)$. According to the importance sampling, the variance is minimised by taking the optimal distribution

$$P(s) \propto \left( \prod_{i=1}^N c(s_i)c(s'_i)^* \right) \langle \psi_f | O_s | \psi_i \rangle. \quad (9)$$
Implementation of the optimal distribution requires knowledge of $|\langle \psi | O_s | \psi \rangle|$.

In this paper, we focus on a practical suboptimal distribution
\begin{equation}
P(s) = \frac{\prod_{i=1}^{N} c(s_i)c(s'_i)^*}{C_A^{2N}}, \tag{10}
\end{equation}
where the normalisation factor $C_A = \sum_s |c(s)|$ determines the variance. Taking the suboptimal distribution, the transition amplitude $\langle \psi | e^{iHt} O e^{-iHt} | \psi \rangle$ is the expected value of $C_A^{2N} e^{i\theta_s} \langle \psi | O_s | \psi \rangle$, where
\begin{equation}
\theta_s = \arg \left( \prod_{i=1}^{N} c(s_i)c(s'_i)^* \right). \tag{11}
\end{equation}

Formally, we have
\begin{equation}
\langle \psi | e^{iHt} O e^{-iHt} | \psi \rangle = E \left[ C_A^{2N} e^{i\theta_s} \langle \psi | O_s | \psi \rangle \right] = \sum_s P(s) C_A^{2N} e^{i\theta_s} \langle \psi | O_s | \psi \rangle. \tag{12}
\end{equation}

Taking the suboptimal distribution, the estimator of $\langle \psi | e^{iHt} O e^{-iHt} | \psi \rangle$ is
\begin{equation}
\hat{A} = C_A^{2N} \langle e^{i\theta_s} \langle \psi | O_s | \psi \rangle \rangle_{N_s}. \tag{13}
\end{equation}
Here, $(\bullet)_{N_s}$ denotes the empirical mean taken over $N_s$ samples of $s$. The variance of the estimator is
\begin{equation}
\text{Var} \left( \hat{A} \right) = \frac{1}{N_s} C_A^{4N} \text{Var} \left( e^{i\theta_s} \langle \psi | O_s | \psi \rangle \right). \tag{14}
\end{equation}

When $O$ is a unitary operator, $|\langle \psi | O_s | \psi \rangle| \leq 1$, and the variance has the upper bound
\begin{equation}
\text{Var} \left( \hat{A} \right) \leq \frac{1}{N_s} C_A^{4N}. \tag{15}
\end{equation}

In our QCMC algorithm, we use the circuits given in Sec. VI to evaluate $\langle \psi | O_s | \psi \rangle$. Each quantum circuit reports a probabilistic binary outcome, the expected value of which is either the real or imaginary part of $e^{i\theta_s} \langle \psi | O_s | \psi \rangle$. We find that the suboptimal distribution (which is suboptimal when we can deterministically evaluate $\langle \psi | O_s | \psi \rangle$) is actually the optimal distribution for the probabilistic evaluation without prior knowledge of $|\langle \psi | O_s | \psi \rangle|$ (see VII). Accordingly, the minimum variance is
\begin{equation}
\text{Var} \left( \hat{A} \right) = \frac{1}{M_{\text{tot}}} \left( 2C_A^{4N} - |\langle \psi | e^{iHt} O e^{-iHt} | \psi \rangle|^2 \right), \tag{16}
\end{equation}
where $2M_{\text{tot}}$ 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 $\Delta t$, $c(1) \simeq 1$, $U(1) \simeq e^{-iH\Delta t}$ 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 minimising $C_A$, in order to reduce the variance.

\section{IV. SUMMATION FORMULAS OF TIME EVOLUTION OPERATORS}

We look for summation formulas satisfying the following criteria:

- The unitary operators $U(s)$ are easy to implement using elementary quantum gates, in order to reduce the gate number.
- The normalisation factor $C_A$ is minimised.
- 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 $l$th-order product formula, the corresponding POE summation formula has the normalisation factor $C_A = 1 + O(\Delta t^{l+1})$. By replacing leading-order Pauli operators with rotation operators, we obtain LOR formulas and the normalisation factor is reduced to $C_A = 1 + O(\Delta t^{l+2})$.

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

\subsection{A. Product formulas}

In this section, we review Lie-Trotter-Suzuki product formulas \cite{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
\begin{equation}
S_1(\Delta t) = e^{-iH_{\text{tot}} \Delta t} \cdots e^{-iH_1 \Delta t} = e^{-iH \Delta t} + O(\Delta t^2). \tag{17}
\end{equation}

Higher-order formulas are defined recursively for any positive integer $m$ by
\begin{equation}
S_{2m}(\Delta t) = K_{2m}(-\Delta t)^{\dagger} K_{2m}(\Delta t) = e^{-iH \Delta t} + O(\Delta t^{2m+1}), \tag{18}
\end{equation}
where $K_2(\Delta t) = S_1(\frac{\Delta t}{2})$, $K_{2m}(\Delta t) = K_{2m-2} \left( (1 - 2r_{p,m} \Delta t) S_{2m-2} (p_{r,m} \Delta t) \right)^\dagger$ (19) when $m > 1$, and $p_{r,m} = \left[ 2r - (2r)^{-1} \right]^{-1}$. Here, $r$ can be any positive integer. $S_{2m}$ is a product of $2r + 1$ $S_{2m-2}$ operators.

For the first-order formula, we define the correction operator
\begin{equation}
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, \tag{20}
\end{equation}
where $R_1^{(k)}$ are operators that are independent of $\Delta t$. Because $V_1(\Delta t)$ is unitary for all real $\Delta t$, all $R_1^{(k)}$ are Hermitian operators. Then,
\begin{equation}
V_1(\Delta t) = 1 - iL_1(\Delta t) + O(\Delta t^4), \tag{21}
\end{equation}
where
The leading-order operator

\[ L_1(\Delta t) = R_1^{(2)} \Delta t^2 + R_1^{(3)} \Delta t^3 \]  

is Hermitian. Later, we show that the Hermitian leading-order operator is important for minimising the normalisation factor \( C_A \).

For higher-order formulas, the correction operators are

\[ V_{2m}(\Delta t) = K_{2m}(-\Delta t) e^{-iH\Delta t} K_{2m}(\Delta t) \]

\[ = e^{-i \sum_{k=2m+1}^{\infty} \rho^{(k)}_{2m} \Delta t^k}, \]

(23)

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

\[ V_{2m}(\Delta t) = 1 - iL_{2m}(\Delta t) + O(\Delta t^{4m+2}), \]

(24)

where the leading-order operator

\[ L_{2m}(\Delta t) = \sum_{k=m}^{2m} R_{2m}^{(2k+1)} \Delta t^{2k+1} \]

(25)

is Hermitian. For the second-order formula,

\[ L_2(\Delta t) = R_2^{(3)} \Delta t^3 + R_2^{(5)} \Delta t^5. \]

(26)

### B. Summation formulas

To simplify the quantum circuits, we work with Pauli operators \( P_n = \{I, X, Y, Z\} \otimes^n \) 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 \( \sigma_j \in P_n \), and \( h_j \) is a real coefficient. We define \( h_{\text{tot}} \equiv \sum_j |h_j| \), which characterises the magnitude of the Hamiltonian.

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

\[ e^{-iH\Delta t} = \sum_{\sigma \in P_n} 2^{-n} \text{Tr} \left( \sigma e^{-iH\Delta t} \right) \sigma. \]

(27)

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

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

\[ e^{-iH\Delta t} = K_L V K_R, \]

(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, \( V = I - iL + T \), where \( L \) is the leading-order operator, and \( T \) is the high-order operator. The normalisation 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 normalisation factor of a LOR summation formula is \( C_A = \sqrt{1 + C_L^2 + C_T} \). The normalisation factors of all the formulas are summarised 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

\[ V_0(\Delta t) = e^{-iH\Delta t} = 1 - iL_0(\Delta t) + T_0(\Delta t), \]

(29)

where the Hermitian leading-order operator is

\[ L_0(\Delta t) = \sum_j h_j \Delta t \sigma_j, \]

(30)

and the high-order operator is

\[ T_0(\Delta t) = \sum_{k=2}^{\infty} \sum_{j_1, \ldots, j_k=1}^M \prod_{k=1}^k \frac{(-ih_{j_1} \Delta t)}{k!} \sigma_{j_1} \cdots \sigma_{j_k}. \]

(31)

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

| Formula Type | \( C_A = 1 + C_L + C_T \) |
|--------------|--------------------------|
| Zeroth-order | \( C_A = \sqrt{1 + C_L^2 + C_T} \) |

| Formula Type | \( l \) (order of formula) | \( \lambda \) | Leading-order contribution \( C_L \) | Simplified leading-order contribution \( C_L \) |
|--------------|----------------|--------|---------------------------------|---------------------------------|
| POE          | 0              | 1      | \( x \)                          | \( \frac{1}{2} x^2 \)            |
|              | 1              | 2      | \( \frac{1}{6} (2x)^3 + \frac{1}{120} (2x)^5 \) | \( < \frac{1}{5} x^2 + \frac{1}{6} (2x)^3 \) |

| Table I. Normalisation factors of summation formulas. In the table, \( x \equiv h_{\text{tot}} \Delta t \). |
2. First-order Pauli-operator-expansion formula

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

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

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

\[ V_1(\Delta t) = e^{-iH\Delta t}e^{ih_1\sigma_1\Delta t}...e^{ih_M\sigma_M\Delta t} = I - iL_1(\Delta t) + T_1(\Delta t), \]  
(33)

where

\[ L_1(\Delta t) = if^{(2)}_1(\Delta t) + if^{(3)}_1(\Delta t), \]  
(34)

\[ T_1(\Delta t) = \sum_{k=4}^\infty F^{(k)}_1(\Delta t), \]  
(35)

and

\[ F^{(k)}_1(\Delta t) = \sum_{k,k_1,...,k_M=0,j_1,...,j_k=1}^\infty \sum_{j=1}^M \delta_{k,k+j} \prod_{a=1}^k (-ih_{j_a}\Delta t) \left[ \prod_{j=1}^M (ih_j\Delta t)^{j_a} \right] \]
\[ \times \sigma_{j_1}...\sigma_{j_{a_1}}\sigma_1^{k_1}...\sigma_{j_{a}}^{k_{a}}, \]  
(36)

Note that the first term in L1 is \( O(\Delta t^2) \) according to discussions on product formulas. For the first-order formula, the normalisation factor is given by \( C_L = \frac{1}{2} (2h_{\text{tot}}\Delta t)^2 + \frac{1}{12} (2h_{\text{tot}}\Delta t)^3 \) and \( C_T = e^{2h_{\text{tot}}\Delta t} - \sum_{k=0}^5 \frac{1}{k!} (2h_{\text{tot}}\Delta t)^k \).

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

\[ e^{-iH\Delta t} = S_1(-\Delta t/2)^1V_2(\Delta t)S_1(\Delta t/2). \]  
(37)

The Taylor expansion of the correction operator reads

\[ V_2(\Delta t) = I - iL_2(\Delta t) + T_2(\Delta t), \]  
(38)

where

\[ L_2(\Delta t) = if^{(3)}_1(\Delta t) + if^{(5)}_1(\Delta t), \]  
(39)

\[ T_2(\Delta t) = \sum_{k=6}^\infty F^{(k)}_2(\Delta t), \]  
(40)

and

\[ F^{(k)}_2(\Delta t) = \sum_{k,k_1,...,k_M=0,j_1,...,j_k=1}^\infty \sum_{j=1}^M \delta_{k,k+j} \prod_{a=1}^k (-ih_{j_a}\Delta t) \left[ \prod_{j=1}^M (ih_j\Delta t)^{j_a} \right] \]
\[ \times \sigma_{j_1}...\sigma_{j_{a_1}}\sigma_1^{k_1}...\sigma_{j_{a}}^{k_{a}}. \]  
(41)

According to discussions on product formulas, \( L_2 \) only contain \( \Delta t^4 \) and \( \Delta t^5 \) terms. For the second-order formula, the normalisation 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 \( C_T = e^{2h_{\text{tot}}\Delta t} - \sum_{k=0}^5 \frac{1}{k!} (2h_{\text{tot}}\Delta t)^k \).

4. Higher-order Pauli-operator-expansion formulas

For the mth-order formula, we express the time evolution operator as

\[ e^{-iH\Delta t} = K_{2m}(-\Delta t)^1V_{2m}(\Delta t)K_{2m}(\Delta t). \]  
(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 normalisation factor of the 2nth-order formula is given by \( C_L = \sum_{k=m}^{2m} \frac{1}{2^{k+1}} (\lambda h_{\text{tot}}\Delta t)^{2k+1} \) and \( C_T = e^{\lambda h_{\text{tot}}\Delta t} - \sum_{k=0}^{4m+1} \frac{1}{k!} (\lambda h_{\text{tot}}\Delta t)^k \). Here, the factor \( \lambda = 1 + \prod_{i=2}^{2m} (4r_{p_i}-1) \) is due to the backward evolution with the time \((1-2r_{p_{r,m}})\Delta t \) 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 normalisation factor. For example, if both \( \sigma \) and \(-\sigma \) exist in the summation formula, the contribution to the normalisation factor is \( 2|\alpha| \), which is reduced to zero after combining like terms. We apply this approach to \( F^{(2)}_1 \) and \( F^{(3)}_2 \) in \( L_1 \) and \( L_2 \), respectively, to minimise the dominant contribution to the normalisation factor. See Appendix A for the simplified expressions of \( F^{(2)}_2 \) and \( F^{(3)}_2 \).

As a result, the leading-order contributions are reduced to \( C_L \leq \frac{1}{2} (h_{\text{tot}}\Delta t)^2 + \frac{1}{12} (2h_{\text{tot}}\Delta t)^3 \) in the first-order formula and \( C_L \leq \frac{1}{15} (h_{\text{tot}}\Delta t)^2 + \frac{1}{120} (2h_{\text{tot}}\Delta t)^5 \) in the second-order formula.

6. Leading-order-rotation formulas

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

\[ L_i = \sum_u \alpha_u \tau_u, \]  
(43)

where the \( \tau_u \in P_n \) are Pauli operators. Here, all \( \alpha_u \) are real because \( L_i \) is Hermitian, which is the key to LOR formulas. To minimise the normalisation factor, we express the leading-order terms as a summation of rotation operators,

\[ I - iL_i = \sum_u \beta_u e^{-i\text{sgn}(\alpha_u)} \phi_{\tau_u}, \]  
(44)
where $\phi = \arctan(C_L)$, $\beta_u = |\alpha_u|/\sin \phi$ and $C_L = \sum_u |\alpha_u|$. The normalisation factor contributed by $1-iL_t$ is $1+C_L$ in POE formulas, which is reduced to $\sum_u |\beta_u| = C_L/\sin \phi = \sqrt{1+C_L^2} \approx 1+C_L^2/2$ 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 normalisation 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 $\Delta t$. 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 of approximately $1/\sqrt{N}C_A^2N$. Therefore, the error for the $l$th-order POE formula is of approximately $1/\sqrt{N}+2N^2O(\Delta t^{l+1})$ and the error for the $l$th-order LOR formula is of approximately $1/\sqrt{N}+2N^2O(\Delta t^{2l+2})$.

For Lie-Trotter-Suzuki product formulas, there are two sources of error: the error due to finite $\Delta t$, i.e., the formulas are approximate, and the error due to sampling noise. The error due to finite $\Delta t$ is systematic and cannot be reduced by increasing the number of samples. For the $l$th order product formula, the error is of approximately $1/\sqrt{N}+NO(\Delta t^{l+1})$, where the first term is due to the sampling noise and the second term is due to the finite $\Delta t$. 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 $1/\sqrt{N}$ in the $\Delta t$ term and the increased exponent of $\Delta t$ (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 $\sigma$ or a rotation operator in the form $e^{-i\sigma}$. In Sec. VII 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 $\Delta t$ [32, 33]. The simplest example is $e^{-iH\Delta t} = 1/2S_2(\Delta t/2)^2 - 1/2S_2(\Delta t) + O(\Delta t^3)$. We can 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 $\langle \psi_s|O_s|\psi_s\rangle$. Finally, with results from the quantum computer, the classical computer calculates the expected value of $e^{i\alpha}|\psi_s|O_s|\psi_s\rangle$ and returns the final estimate of the transition amplitude $\langle \psi_f|e^{iHt}Oen\psi_s\rangle$.

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 perturbative-like 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 $\xi_0$ and $x$ 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 $h = (h_1, \ldots, h_M)$ and a vector of Pauli operators $\sigma = (\sigma_1, \ldots, \sigma_M)$. 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, $h$, $\sigma$, $N$, and $\Delta t$, 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.

**Algorithm 1** Sample generation.

1: function SAMGEN(F, h, σ, N, Δt)
2: for i = 1 to N do
3: (W_i, θ_i) ← SAMGENONESTEP(F, h, σ, Δt)
4: \(W_i', θ_i'\) ← SAMGENONESTEP(F, h, σ, Δt)
5: \(W ← (W_1, \ldots, W_N, W_1', \ldots, W_N')\)
6: \(θ ← \sum_{i=1}^{N} (θ_i - θ_i')\)
7: Output \((W, θ)\)

**Algorithm 2** Sample generation for one time step.

1: function SAMGENONESTEP(F, h, σ, Δt)
2: Compute \(L_2\) according to Eq. (39), simplify \(L_2\) by combining like terms, obtain the final expression \(L_2 = \sum_{σ} a_σ τ_σ \Rightarrow τ_σ \in \mathcal{P}_n\)
3: \(C_T ← \sum_{a_σ} [a_σ]\)
4: \(C_T ← e^{2ih_σ \text{total} \Delta t} - \sum_{a_σ} \frac{1}{2} (2h_σ \text{total} \Delta t)^k \Rightarrow h_σ \text{total} = \sum_{j} |h_j|\)
5: if \(F = P\) then \(C_A ← 1 + C_T + C_T\)
6: else if \(F = R\) then \(C_A ← \sqrt{1 + C_T^2 + C_T}\)
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 \(O = P\) then \(\triangleright\) Sample from leading-order terms
10: \(\{(\tau_a, \arg(-iσ_a))\}\) with probabilities \(1/(1 + C_L)\) and \(\{(a_σ)/C_L, \text{respectively}\), respectively.
11: \(\triangleright\) Sample from high-order terms
12: \(k, k_j, k_j' ← 0\)
13: while \(k + \sum_{j=1}^{N} (k_j + k_j') < 6\) do
14: \(k ← \text{POISSON}(h_σ \text{total} \Delta t) \Rightarrow \text{POISSON}(x)\) returns \(k \in \{0, 1, \ldots\}\) with the probability \(e^{-x}x^k/k!\).
15: \(\text{for } j = 1 \text{ to } M \text{ do}\)
16: \(k_j ← \text{POISSON}(|h_j|/h_σ \text{total} \Delta t/2)\)
17: \(k_j' ← \text{POISSON}(|h_j'|/h_σ \text{total} \Delta t/2)\)
18: \(\text{for } a = 1 \text{ to } k \text{ do}\) Choose \(j_a\) from \(\{1, \ldots, M\}\) with probabilities \(\{(|h_{j_a}|/h_σ \text{total} \Delta t)\}\).
19: \(W ← \zeta s_M^{k_M} \cdots s_1^{k_1} \sigma_{j_1} \cdots \sigma_{j_a} \cdots \sigma_{j_M}^{k_M} \Rightarrow \text{Take } \zeta = ±1, ±i \text{ to meet } W \in \mathcal{P}_n\).
20: \(θ ← \arg\left(\zeta \prod_{a=1}^{k} (-i\hbar_{j_a}) \times \prod_{j=1}^{M} (i\hbar_j)^{k_j + k_j'}\right)\)
21: Output \((W, θ)\).

In the second-order summation formulas, each term is in the form \(S_1(-\frac{\Delta t}{2}) W S_1(\frac{\Delta t}{2})\): 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 \(U(s_i) = S_1(-\frac{\Delta t}{2}) W S_1(\frac{\Delta t}{2})\) and \(U(s_i') = S_1(\frac{\Delta t}{2}) W S_1(-\frac{\Delta t}{2})\), we have
\[
O_S = S_1 W_1 S_1' \cdots S_N W_N S_1' \times OS_1 W_N S_1' \cdots S'_1 W_1 S_1, \tag{45}
\]
Here, we use the notations \(S_1 = S_1(\Delta t)\) and \(S'_1 = S_1(-\frac{\Delta t}{2})\) for simplicity. Given the vector of correction operators \(W = (W_1, \ldots, W_N, W_1', \ldots, W_N')\), the quantum computer can evaluate \(\langle \psi | O_S | \psi \rangle\).

In the final phase, the classical computer estimates the transition amplitude by computing the expected value of \(e^{iθ} \langle \psi | O_S | \psi \rangle\). Therefore, the sampling algorithm also needs to output \(θ_s\).

Overall, the outputs of the sampling algorithm are \(W\) and \(θ\). The procedure for generating \(W\) and \(θ\) is given in Algorithm 1. Algorithm 2 is a subroutine for processing 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 \(\langle \psi | O_S | \psi \rangle\), respectively. Because the transition amplitude is a complex number in general, the quantum computing returns two real numbers \(a_{R,s}\) and \(a_{I,s}\), which are estimates of the real and imaginary parts of \(e^{iθ} \langle \psi | O_S | \psi \rangle\), respectively. By computing expected values of \(a_{R,s}\) and \(a_{I,s}\), we obtain the transition amplitude \(\langle \psi | e^{iHt_0} O_S e^{-iHt_0} | \psi \rangle\) up to the factor \(C_A^N\). QCMC is summarised in Algorithm 3.

**Algorithm 3** Quantum-circuit Monte Carlo.

1: Input \(F, h, σ, N, Δt, N_s, M_s\).
2: for \(v = 1\) to \(N_s\) do \(\triangleright v\) is the label of \(s\) samples.
3: \((W, θ) ← \text{SAMGEN}(F, h, σ, N, Δt)\)
4: \((a_{R,v}, a_{I,v}) ← \text{QUANTUMCIRCUITS}(\ldots, W, θ, M_s)\)
5: \(A ← C_A N_s \sum_{v=1}^{N_s} (a_{R,v} + ia_{I,v})\)
6: Output \(A\) as the estimate of \(\langle \psi | e^{iHt_0} O_S e^{-iHt_0} | \psi \rangle\).

**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 \(e^{-iH\Delta t} = \sum c(s)σ_s\), where the \(σ_s \in \mathcal{P}_n\) are Pauli operators. We can express a Pauli operator as
\[
σ = i^{x_1 z_1} X^{x_1} Z^{z_1} \cdots i^{x_n z_n} X^{x_n} Z^{z_n}, \tag{47}
\]
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 \(\bigotimes_{a=1}^{n} |μ_a\rangle\), where \(μ_a = 0, 1\). A Pauli operator acting on a basis state always results in a basis state, i.e. \(σ \bigotimes_{a=1}^{n} |μ_a\rangle = \bigotimes_{a=1}^{n} i^{x_{a} z_{a}} (-1)^{z_a μ_a} |μ_a + x_a\rangle\), 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 $\bigotimes_{\alpha=1}^{n} |\psi_{\alpha}\rangle$ and stabiliser states \textsuperscript{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

$$\langle \psi_{1} | O_{\alpha} | \psi_{1} \rangle = \langle \psi_{1} | \sigma_{1} \cdots \sigma_{N} O \sigma_{1} \cdots \sigma_{N} | \psi_{1} \rangle$$  \hspace{1cm} (48)$$
onumber

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 $\langle \psi_{1} | O_{\alpha} | \psi_{1} \rangle$ 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 $\sigma$ and $\tau$ are different, we have $\langle \psi | \sigma \tau \psi \rangle = 0$ for all computational basis states $|\psi\rangle = \bigotimes_{\alpha=1}^{n} |\mu_{\alpha}\rangle$. The short-time evolution operator $e^{-iH\Delta t} \approx \mathbb{I} - iH\Delta t$ acting on a basis state results in

$$e^{-iH\Delta t} |\psi\rangle \approx |\psi\rangle - i\Delta t \sum_{\alpha} h_{\alpha} |\psi_{\alpha}\rangle.$$  \hspace{1cm} (49)$$

We find that there is no interference between the terms if and only if $\langle \psi | \sigma \tau \psi \rangle = 0$ for all $\sigma, \tau \in \{I\} \cup \{\sigma_{j}\}$: 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 $|\psi\rangle$; in QCMC, we sample Pauli operators. Substituting the computational basis for $\{|\psi\rangle\}$, the transition amplitude of each time step reads $\langle \psi' | e^{-iH\Delta t} | \psi \rangle$, where $|\psi'\rangle = \bigotimes_{\alpha=1}^{n} |\mu'_{\alpha}\rangle$. 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 $\{I\} \cup \{\sigma_{j}\}$ have one-to-one correspondence in the limit of small $\Delta t$. 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.

\section{VI. QUANTUM CIRCUITS}

We propose quantum circuits for evaluating the transition amplitude of the operator $O_{\alpha}$, 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 $\sigma$ or rotation operators $e^{-i\sigma\cdot a}$.

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.

\subsection{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 $S_{1}^{r}$ products from the circuit; for the zeroth-order formulas, we remove both the $S_{1}$ and the $S_{1}^{r}$ products; and by adding more $S_{1}$ and $S_{1}^{r}$ products, the circuit can be used for higher-order formulas. If we ignore controlled-correction gates, the compact circuit for the $l$th-order summation formula is the same as the circuit for the $l$th-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

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left( |0\rangle_{a} \otimes OS_{1}^{r} W_{N} S_{1} \cdots S_{1}^{r} W_{1} S_{1} |\psi_{1}\rangle + |1\rangle_{a} \otimes S_{1}^{r} W_{N} S_{1} \cdots S_{1}^{r} W_{1} S_{1} |\psi_{1}\rangle \right).$$  \hspace{1cm} (50)$$

Measuring the ancillary qubit, we obtain

$$\langle \Psi | X_{a} | \Psi \rangle = \text{Re} \left( \langle \psi_{1} | O_{a} | \psi_{1} \rangle \right),$$  \hspace{1cm} (51a)$$

$$\langle \Psi | Y_{a} | \Psi \rangle = -\text{Im} \left( \langle \psi_{1} | O_{a} | \psi_{1} \rangle \right),$$  \hspace{1cm} (51b)$$

where $X_{a}$ and $Y_{a}$ are Pauli operators of the ancillary qubit. Here, we use Eq. (45). The procedure for evaluating $\langle \psi_{1} | O_{a} | \psi_{1} \rangle$ using compact circuits is given in Algorithm 4.
Algorithm 4 Quantum circuit evaluation.

1: function QUANTUMCIRCUITS($h, \sigma, N, \Delta t, W, \theta, M_a$)
2:     $a_R, a_t \leftarrow 0$
3:     for $i = 1$ to $M_a$
4:         implement the circuit, measure $\cos \theta X_a + \sin \theta Y_a$
5:         and collect the outcome $\mu_R = \pm 1$
6:     end for
7:     $a_t \leftarrow a_t + \mu_1$
8:     $a_R \leftarrow a_R/M_a$
9:     $a_t \leftarrow a_t/M_a$
10:    Output ($\mu_R, a_t$).

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

$$\Phi = \frac{1}{\sqrt{2}} \left( |0\rangle \otimes |0\rangle_{\otimes n} + |1\rangle \otimes U_1 \otimes \ldots \otimes U_n \otimes |\psi_1\rangle \right). \quad (52)$$

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

$$\langle \Phi|X_a|\Phi\rangle = \text{Re} (\langle \psi_1|O_{\sigma}|\psi_1\rangle), \quad (53a)$$
$$\langle \Phi|Y_a|\Phi\rangle = \text{Im} (\langle \psi_1|O_{\sigma}|\psi_1\rangle). \quad (53b)$$

The procedure for evaluating $\langle \psi_1|O_{\sigma}|\psi_1\rangle$ 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.

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 minimise their number.

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

$$\tilde{\sigma} = \tilde{R}\sigma\tilde{R} = X^{z_1x_1} \otimes \ldots \otimes X^{z_nx_n},$$
where $x_a \lor z_a = 1 - (1 - x_a)(1 - z_a)$.

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

In the compact circuit shown in Fig. 2(a), there are two controlled-correction gates in each 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-$\tilde{\sigma}$ gate in the following way. Note that $\tau'\tau = \zeta\sigma$, where $\zeta$ is a phase factor. We apply $\tau$ first, then a controlled-$\sigma$ gate, and finally a phase gate 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 $N_\Lambda$.

Now, we present another protocol for the controlled-$\sigma$
 gate. The circuit is shown in Fig. 3(a), which is formed of three parts: gates transforming a general Pauli operator $\sigma$ 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 $x_1 \lor z_1 = 1$. In this protocol, there is only one instead of $N_\Lambda$ 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 $\Lambda$ with $\tilde{\Lambda}$ in the expansion is the identity. (54)

To transform $\tilde{\sigma}$ into $X_1$, we look for a transformation $\Lambda$ that satisfies $\tilde{\sigma} = \Lambda X_1 \Lambda$. On the all-to-all network, we take $\Lambda = \prod_{n=2}^{\infty} \Lambda_{n,n-1}$ and the number of controlled-NOT gates for each $\Lambda$ is $N_\Lambda - 1$. On the linear network, we take $\Lambda = \Lambda_{1,2} \Lambda_{2,3} \Lambda_{3,4} \cdots \Lambda_{n-1,n} \Lambda_{n,n-1}$, (54)

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

The maximum number of controlled-NOT gates for implementing the two controlled-correction gates in each time step is summarised 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 $2[2(n - 1) + 1] = 4n$ for LOR formulas. On the linear network, the maximum gate number is $2(2n - 2) + 1 = 4n - 3$ for POE formulas and $2[2(2n - 2) + 1] = 8n - 4$ 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.

VII. OPTIMAL DISTRIBUTION

In this section, we derive the optimal distribution of $s$ that minimises the variance in Monte Carlo. Using the protocols in Sec. VI to evaluate $\langle \psi | O_s | \psi \rangle$, 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 $a_{R,s}$ and $a_{I,s}$ are random variables. We suppose that $a_{\nu,s}$ ($\nu = R, I$) takes the value $a_i$ with the probability $P_{\nu,s,i}$ in the quantum computing; then, its expected value is $E[a_{\nu,s}]_{QC} = \sum_i P_{\nu,s,i} a_i$. Here, $E[\bullet]_{QC}$ denotes the mean taken over both $s$ and quantum computing runs. Using the protocols in Sec. VI, $a_{R,s}$ and $a_{I,s}$ are unbiased estimators of $\langle \psi | O_s | \psi \rangle$, i.e.

$$e^{i\theta_s} \langle \psi | O_s | \psi \rangle = E[a_{R,s}]_{QC} + i E[a_{I,s}]_{QC}. \quad (55)$$

Let $A_R$ and $A_I$ be the real and imaginary parts of $\langle \psi | e^{iH_{QC}} O_s e^{-iH_{QC}} | \psi \rangle$, respectively. In the QMC algorithm, we evaluate the summation $A_s = \sum_s c_s E[a_{\nu,s}]_{QC}$ using the Monte Carlo method, where $c_s = |\prod_{i=1}^{N} c(s_i) c(s_i')|^2$. Given any probability distribution $P(s)$, we have

$$A_s = \sum_s P(s) c_s E[a_{\nu,s}]_{QC} = E \left[ c_s a_{\nu,s} \right]. \quad (56)$$

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

$$\text{Var} \left( \hat{A}_s \right) \propto \frac{1}{N_s} \text{Var} \left( \frac{c_s a_{\nu,s}}{P(s)} \right) = \frac{1}{N_s} \sum_s c_s^2 a_{\nu,s}^2 \frac{1}{P(s)} - \frac{A_s^2}{N_s}. \quad (57)$$

where $\alpha_{\nu,s} = \sqrt{\text{E} [ a_{\nu,s}^2 ]_{QC}}$. The optimal distribution that minimises the variance is $P(s) \propto |c_s| \alpha_{\nu,s}$, and the minimum variance is

$$\text{Var} \left( \hat{A}_s \right) \propto \frac{1}{N_s} \sum_s c_s^2 a_{\nu,s}^2 \frac{1}{P(s)} - \frac{A_s^2}{N_s}. \quad (58)$$

Now, we consider that $a_{\nu,s}$ is obtained by taking the empirical mean of $M_s$ binary numbers. Each binary number takes $\pm 1$ corresponding to the measurement outcome of the ancillary qubit (see Algorithm 4). Then, $a_{\nu,s}$ follows the binomial distribution and

$$\alpha_{\nu,s} = \sqrt{1 + (M_s - 1) \text{E} [ a_{\nu,s}^2 ]_{QC}}. \quad (59)$$

Let $M_{tot}$ be the total number of circuit shots; we have $N_s = M_{tot}/M_s$. Substituting $\alpha_{\nu,s}$ 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 minimised at $M_s = 1$. When $M_s = 1$, we have $\alpha_{\nu,s} = 1$, and the optimal distribution is $P(s) \propto |c_s|$. Accordingly, the minimum variance is

$$\text{Var} \left( \hat{A}_s \right) = \frac{1}{M_{tot}} \left( \sum_s |c_s|^2 \right) - \frac{A_s^2}{M_{tot}}. \quad (60)$$

With $\text{Var} \left( \hat{A} \right) = \text{Var} \left( \hat{A}_R \right) + \text{Var} \left( \hat{A}_I \right)$, 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 $a_{\nu,s}$. If we have prior knowledge of the $a_{\nu,s}$ distribution, we can use other estimators such as the Bayes estimator to reduce the variance. In the extreme case, suppose that $E[a_{\nu,s}]_{QC}$ is known, the optimal distribution is $P(s) \propto |c_s E[a_{\nu,s}]_{QC}|$ instead of $P(s) \propto |c_s a_{\nu,s}|$ (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. quasi-probability 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 quasi-probability decomposition in QC\(MC\), 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 $\Delta t$ to minimise the variance. Eventually, the variance is determined by the error rate, which is discussed in Sec. IX.

A. Quasi-probability decomposition

In the quasi-probability decomposition, an error-free quantum operation is expressed as a linear combination of noisy operations. Let $G^d = [U]$ and $G_i$ be the error-free operation and noisy operations, respectively. The quasi-probability decomposition in the form

$$ G^{ef} = \sum_i q_i G_i, $$  \hspace{1cm} (61)

where $q_i$ are real coefficients, i.e. quasi-probabilities. Here, $U$ is a unitary quantum gate, $[U](\bullet) = U \bullet U^\dagger$ is the trace-preserving completely positive map of the gate and $G_i$ 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 using Pauli twirling \[72\]. In the Pauli error model, the noisy operation of a two-qubit gate $U$ reads $G = N[U]$, where the noise map is

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

$p_\sigma \ll 1$ is the rate of Pauli error $\sigma$, and $p = \sum_{\sigma \not= I \otimes I} p_\sigma$ is the total error rate. The inverse map of $N$ is also in the Pauli-operation summation form, i.e.

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

and we can solve coefficients $q_\sigma$ numerically. Without a general analytically expression of $q_\sigma$, 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
\begin{align}
q_{\sigma \otimes I} &= 1 + p + O(p^2), \\
q_{\sigma \neq I \otimes I} &= -p_{\sigma} + O(p^2).
\end{align}

Given the inverse map, the quasi-probability decomposition of gate $U$ is
\begin{equation}
G^\text{ef} = \sum_{\sigma \in \{I, X, Y, Z\}^\otimes} q_{\sigma} [\sigma] G.
\end{equation}

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

Now, we apply the quasi-probability 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
\begin{equation}
\langle Z_a \rangle = \text{Tr} \left[ Z_a G_{N_G} \cdots G_1(\rho) \right],
\end{equation}
where $\rho = |0\rangle\langle 0|^n$ is the initial state of the quantum circuit, and $N_G$ is the number of gates. Suppose that the quasi-probability decomposition of each gate is $G^\text{ef}_j = \sum_i q_{j,i} G_{j,i}$. The error-free mean is
\begin{equation}
\langle Z_a \rangle^\text{ef} = \sum_{i_1, \ldots, i_{N_G}} \left( \prod_{j=1}^{N_G} q_{j,i_j} \right) \times \text{Tr} \left[ Z_a G_{N_G,i_{N_G}} \cdots G_{1,i_1}(\rho) \right].
\end{equation}
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 $C_E = \prod_{j=1}^{N_G} (\sum_i |q_{j,i}|^2)$. 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 $j$th 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 quasi-probability decomposition,
\begin{equation}
\langle Z_a \rangle^\text{ef}_{C_0} = \sum_k q_k \langle Z_a \rangle_{C_k},
\end{equation}
where $\langle Z_a \rangle_{C_k}$ is the mean of $Z_a$ in the circuit $C_k$, $C_0$ is the original circuit, and the $C_{k \neq 0}$ 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 quasi-probabilities using gate set tomography [59]; with correlations, we can determine quasi-probabilities using data of Clifford circuits, i.e., Clifford sampling [73, 74]. Given the quasi-probability decomposition formulas, we can evaluate $\langle \psi \rangle\langle O_s | \psi \rangle$ with error mitigation following the procedure in Algorithm 5.

**Algorithm 5** Quantum circuit evaluation with error mitigation.
\begin{itemize}
  \item[1.] Compose the circuit $C_0$ according to input parameters, in which $\nu = \mathbb{R}, I$ (real or imaginary) determines the basis adjusting gate $B$ in Fig. 2.
  \item[2.] Work out the decomposition formula in Eq. (69).
  \item[3.] $C_E \leftarrow \sum_k |q_k|$
  \item[4.] $a_\nu \leftarrow 0$
  \item[5.] for $i = 1$ to $M$, do
  \item[6.] Choose $k$ with the probability $|q_k|/C_E$
  \item[7.] Implement the circuit $C_k$, measure $Z_a$ and collect the outcome $\mu = \pm 1$.
  \item[8.] $a_\nu \leftarrow a_\nu + C_E \mu$
  \item[9.] $a_\nu \leftarrow a_\nu / M_a$
  \item[10.] Output $a_\nu$.\end{itemize}

**B. Inherent error mitigation by postselection**

As shown in verified quantum phase estimation [75] and dual-state purification [76], a quantum circuit with the 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
\begin{equation}
|\Phi'\rangle = \frac{|0\rangle_a + \left< \psi \right|O_s|\psi\rangle |1\rangle_a}{\sqrt{1 + |\left< \psi \right|O_s|\psi\rangle|^2}} \otimes |0\rangle^\otimes n).
\end{equation}

Measuring the ancillary qubit in the state after postselection, we have
\begin{align}
\langle X_a \rangle_0 &= \frac{2\text{Re} \left( \left< \psi \right|O_s|\psi\rangle \right)}{1 + |\left< \psi \right|O_s|\psi\rangle|^2}, \\
\langle Y_a \rangle_0 &= \frac{2\text{Im} \left( \left< \psi \right|O_s|\psi\rangle \right)}{1 + |\left< \psi \right|O_s|\psi\rangle|^2}, \tag{71b}
\langle Z_a \rangle_0 &= \frac{1 - |\left< \psi \right|O_s|\psi\rangle|^2}{1 + |\left< \psi \right|O_s|\psi\rangle|^2}. \tag{71c}
\end{align}
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 $|\psi_i\rangle = |\psi_i\rangle$. The observable is $O = 2c_{3,\uparrow}^\dagger c_{3,\uparrow} - \mathbb{1}$ and the simulation is to compute $\langle O \rangle = \langle \psi | e^{iHt} O e^{-iHt} |\psi\rangle$. To minimise the variance of QCMC, we first expand the correction operator using Pauli operators, i.e. $V_1 = \sum_{\sigma \in P_n} (a_\sigma - ib_\sigma)\sigma$, (75)
where $a_\sigma$ and $b_\sigma$ are real, and

$$a_\sigma - ib_\sigma = 2^{-n} \text{Tr} \left[ \sigma e^{-iHt} S_1 (\Delta t)^f \right].$$

(66)

We have $a_\| > 0$ and $b_\| = 0$. Then, we take the summation formula

$$e^{-iHt} = \sum_{\sigma \in \mathbb{Z}_n \setminus \{\|\}} \left( a_\sigma \sigma + \beta_\sigma e^{-i\text{sgn}(b_\sigma)\phi_\sigma} \right) S_1,$$

(67)

where $\phi = \arctan(a_\|^2 \sum_\sigma |b_\sigma|^2)$ 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 significantly suppressed, as shown in Fig. 4. We model the noise in quantum computing using the depolarising error model. For a controlled-NOT gate, the noise model is given by Eq. (62) with parameters $p_\sigma = p/15$. We neglect errors in the initialisation, 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 86 time steps, i.e. the total number of controlled-NOT gates is above 23.8. 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 $N_s = 10000$ 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 $1/N_C^4 = 1/N_C^4 \Delta t^{-1} \ln C_A$. As summarised in Table I, $C_A = 1 + \xi \Delta t^k + O(\Delta t^{k+1})$, where $\xi$ is a constant depending on the Hamiltonian. When $k > 1$, by taking $\Delta t = (\delta/4\pi)^{1/(k-1)}$, we can reduce the factor to $C_A^4 = e^{\delta+O(\delta^{k/(k-1)})}$ for any small $\delta$. We note that $k > 1$ in $l$th-order POE formulas with $l > 0$ and all LOR formulas. For the zeroth-order POE formula, because $C_A = e^{\text{h_{tot}t}}$ (i.e. $k = 1$), we have $C_A^4 = e^{\text{h_{tot}t}}$ for all $\Delta t$.

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. V C). In this classical algorithm, the variance increases exponentially with the evolution time and system size, i.e. approximately $1/N_C^4 e^{\text{h_{tot}t}}$, whatever $\Delta t$ we choose. Here, $t$ is the evolution time and $\text{h_{tot}}$ increases with the system size. The variance is up to minimization, e.g. changing the Hilbert space basis [23] and optimising 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 $\Delta t$, we can reduce the factor $C_A^4$ to a satisfactory level and $\Delta t$ scales polynomially with $t$ and $\text{h_{tot}}$. Therefore, the number of times steps $N = t/\Delta t$, i.e. the circuit depth, scales polynomially with $t$ and $\text{h_{tot}}$.

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 \( \langle e^{i\theta} \rangle \) denotes the phase of \( e^{i\theta} \langle \psi_f | O_s | \psi_i \rangle \). We find that the sign problem is significant in the zeroth-order POE formula, i.e. \( \langle e^{i\theta} \rangle \) 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. \( \langle e^{i\theta} \rangle \) 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 quasi-probability decomposition to mitigate errors, the error mitigation enlarges the variance. The variance taking into account quantum error mitigation is approximately

\[
\frac{1}{N_s} C^4 A^N C^2 \tilde{E}^2
\]

We consider the compact circuit in Fig. 2(a). Let \( \epsilon \) 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 \( \epsilon' \) and the factor due to error mitigation is \( C_E \approx (1 + 2\epsilon')(1 + 2(g + \eta)\epsilon)^N \), according to the Pauli error model. Then, we can express the variance in the form

\[
\frac{1}{N_s} C^4 A^N C^2 \approx \frac{1}{N_s} (1 + 2\epsilon'\epsilon)^2 e^{\gamma h_{\text{tot}}},
\]

where

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

We find that the rate \( \gamma \) decreases with the error rate.

Given the error rate of the noisy quantum computer, we choose the time step size \( \Delta t \) to minimise the rate \( \gamma \). Taking \( C_A \approx 1 + \xi \Delta t^k \), we find that the optimal step size is

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

and the corresponding minimum rate is

\[
\gamma_{\text{min}} \approx \frac{k^k e^{1/k}}{h_{\text{tot}} \Delta t} \left[ \frac{(g + \eta)\epsilon}{k - 1} \right]^{k-1}. \]

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. 1. For the first- and second-order formulas, we take the upper bound of the simplified leading-order contribution. We find that the second-order LOR formula outperforms other formulas. Taking higher-order formulas does not further reduce \( \gamma \) for the given error rates.

Although the cost of the QCMC algorithm on a noisy quantum computer increases 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 \( \delta \), i.e. to reduce the variance to \( \delta^2 \), we take

\[
N_s \sim e^{\gamma h_{\text{tot}} / \delta^2}. \]

In the classical algorithm, \( \gamma = 1 \). In the quantum algorithm, taking the minimum value for \( \eta = 1 \) [see Fig. 7(b)], we have \( \gamma \approx 0.34 \) when the error rate per elementary product is \( \epsilon = 0.1 \) and \( \gamma \approx 0.058 \) when \( \epsilon = 0.01 \). For \( h_{\text{tot}} t = 4 \), the quantum algorithm reduces the sample size \( N_s \) by a factor of approximately \( 4 \times 10^4 \) when \( \epsilon = 0.1 \) and approximately \( 4 \times 10^6 \) 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 \( \gamma \approx 2.45 \epsilon^{0.82} \).

We can optimise 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

\[
\begin{align*}
\text{Fig. 7. The minimum rate of increase of variance } \gamma_{\text{min}}, \epsilon & \text{ is the error rate of one elementary Lie-Trotter-Suzuki product } S_1, \eta \epsilon \text{ is the average error rate of the controlled-correction gates in one time step. POE } l \text{ denotes the } l\text{th-order Pauli-operator-expansion formula and LOR } l \text{ denotes the } l\text{th-order leading-order-rotation formula.}
\end{align*}
\]
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 $\langle \psi_i | O_x | \psi_i \rangle$, we can optimise 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 $\gamma$ variance due to quantum error mitigation can be reduced. of generating samples to reduce the variance. Third, the

\[ \langle f \rangle = \frac{1}{Z} \sum_i \frac{1}{Z_i} \prod_{k=1}^{L} \sum_{\phi_k} e^{-\beta H_k(\phi_k)} \psi_i | O_x | \psi_i \]

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 $\langle \psi_i | O_x | \psi_i \rangle$, we can optimise 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 $\gamma$ variance due to quantum error mitigation can be reduced.

\[ \langle f \rangle = \frac{1}{Z} \sum_i \frac{1}{Z_i} \prod_{k=1}^{L} \sum_{\phi_k} e^{-\beta H_k(\phi_k)} \psi_i | O_x | \psi_i \]

In the error mitigation protocol used to estimate $\gamma$ variance due to quantum error mitigation can be reduced. of generating samples to reduce the variance. Third, the variance due to quantum error mitigation can be reduced.

In this paper, we propose a QMC algorithm that uses quantum computing as a subroutine, which allows the non-variational quantum simulation to be implemented with noisy intermediate-scale quantum hardware. In our algorithm, we use exact summation formulas to express the time evolution operator. We optimise these summation formulas and quantum circuits to minimise 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 $\Delta t$, we can maximise the quantum speedup 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 non-variational simulation of quantum many-body systems. We theoretically analyse 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.

ACKNOWLEDGMENTS

We acknowledge the use of simulation toolkit QuESTLink [81] for this work. We acknowledge the support of the National Natural Science Foundation of China (Grant No. 11875050 and 12088101) and NSAF (Grant No. U1930403).

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.

[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. Pedevilla, Steven 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 Ulf-G. Meißner, Ab initio Lattice Results for Fermi Polaron in Two Dimensions, Phys. Rev. Lett. 115, 185301 (2015).
[12] K. Van Houcke, E. Kozik, N. Proko’ev, and B. Svistunov, Diagrammatic Monte Carlo, Phys. Procedia 6, 95 (2010).
[13] K. Van Houcke, F. Werner, E. Kozik, N. Proko’ev, B. Svistunov, M. J. H. Ku, A. T. Sommer, L. W. Cheuk, A. Schirozbek, 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 $^4$He nuclei with the Reid potential, Nucl. Phys. A 361, 399 (1981).

[17] J. Hubbard, Electron correlations in narrow energy bands, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences. 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 Ulf-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, Ulf-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, Essaying the Monte Carlo sign problem, Sci. Adv. 6, eabb341 (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, B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. 270, 359 (2007).

[29] N. Wiebe, D. Berry, P. Hoyer, 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. Kieferová, 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. 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. Pedervi, 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, 042305 (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. McArdrle, 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. McArdrle, 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, Ulf-G. Meißner, and G. Rupak, Nuclear lattice simulations using symmetry-sign extrapolation, Eur. Phys. J. A 51, 92 (2015).
[61] E. Meißner, H. Krebs, T. A. Lähde, D. Lee, Ulf-G. Meißner, and G. Rupak, Ab Initio Calculation of the Spectrum and Structure of 60O, Phys. Rev. Lett. 112, 102501 (2014).
[62] D. Frame, R. He, I. Ipsen, Daniel Lee, Dean Lee, and E. Rrapaj, Eigenvector continuation with subspace learning, Phys. Rev. Lett. 121, 032501 (2018).
[63] S. König, A. Ekström, K. Hebelev, 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 Computation and Quantum Information 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, 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 $\Delta t$. Amplitudes $\langle \psi_i | O_x | \psi_i \rangle$ in the classical Monte Carlo simulation take three values 0 and $\pm 1$. We choose sample numbers $N_x$ in the classical Monte Carlo such that the number of non-zero amplitudes is larger than 10000. In other words, even if we generate 10000 samples according to the optimal distribution in Eq. (9), the variance is not smaller.
[78] P. A. M. Casares, R. Campos, 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 et al., Quantum supremacy using a programmable superconducting processor, Nature 574, 505 (2019).
[81] T. Jones and S. Benjamin, QuESTlink—Mathematica embiggend 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.

Appendix A: Leading-order terms

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

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

We have

$$A^2 = \sum_{i<j} (A_i A_j + A_j A_i) + \sum_i A_i^2 \quad (A2)$$

and

$$A^3 = \sum_{i<j<k} (A_i A_j A_k + A_k A_j A_i + A_j A_i A_k) + \sum_{i<j} (A_i^2 A_j + A_j A_i^2 + A_i A_j A_i) + \sum_i A_i A_j + A_j A_i A_j + \sum_i A_i^3. \quad (A3)$$
1. First-order formula

According to the first-order formula, we have
\[
S_1(\Delta t) = e^{-A_1} \ldots e^{-A_M}
\]
\[
= \prod_{i=1}^{M} \left( \mathbb{1} - A_i + \frac{1}{2} A_i^2 - \frac{1}{6} A_i^3 + O(\Delta t^4) \right)
\]
\[
= \mathbb{1} - A + \frac{1}{2} A(2) - \frac{1}{6} A(3) + O(\Delta t^4), \quad (A4)
\]
where
\[
A(2) = 2 \sum_{i<j} A_i A_j + \sum_i A_i^2, \quad (A5)
\]
\[
A(3) = 6 \sum_{i<j<k} A_i A_j A_k + 3 \sum_{i<j} (A_i^2 A_j + A_i A_j^2) + \sum_i A_i^3. \quad (A6)
\]
The correction operator is
\[
V_1(\Delta t) = \mathbb{1} - A^2 + \frac{1}{2} \left( A^2 + A(2) \right) + \frac{1}{6} \left( A^3 - A(3) \right) + O(\Delta t^4).
\]
Then, we have
\[
F_1^{(2)}(\Delta t) = -A^2 + \frac{1}{2} \left( A^2 + A(2) \right) = \frac{1}{2} \left( A(2) - A^2 \right)
\]
\[
= \frac{1}{2} \sum_{i<j} (A_i A_j - A_j A_i) \quad (A8)
\]
and
\[
F_1^{(3)}(\Delta t) = \frac{1}{2} A \left( A(2) - A^2 \right) + \frac{1}{6} \left( A^3 - A(3) \right)
\]
\[
= \frac{1}{2} \sum_{i<j<k} (A_i A_j A_k - A_k A_j A_i + A_j A_k A_i)
+ \frac{1}{2} \sum_{i<j} (A_i^2 A_j - A_i A_j^2 - A_j A_i^2 + A_j A_i A_j)
+ \frac{1}{6} \left( A^3 - A(3) \right)
\]
\[
= \frac{1}{6} \sum_{i<j<k} (-2A_i A_j A_k - 2A_k A_j A_i + 4A_j A_i A_k
+ 4A_k A_i A_j - 2A_i A_k A_j - 2A_j A_k A_i)
+ \frac{1}{6} \sum_{i<j} (A_i^2 A_j + A_j^2 A_i - 2A_i A_j A_i
- 2A_i A_j^2 - 2A_j A_i A_j).
\]
According to Eq. (A8), the contribution of $F_1^{(2)}(\Delta t)$ to the normalisation factor is
\[
\sum_{i<j} |h_i h_j| \Delta t^2 < \frac{1}{2} \left( \sum_i |h_i| \right)^2 \Delta t^2. \quad (A10)
\]

2. Second-order formula

We can write the second-order correction operator as
\[
V_2(\Delta t) = V_1 \left( -\frac{\Delta t}{2} \right) V_1 \left( \frac{\Delta t}{2} \right)
\]
\[
= \mathbb{1} + F_1^{(3)} \left( \frac{\Delta t}{2} \right) + F_1^{(3)} \left( -\frac{\Delta t}{2} \right) + O(\Delta t^5). \quad (A11)
\]
Then, we have
\[
F_2^{(3)}(\Delta t) = F_1^{(3)} \left( \frac{\Delta t}{2} \right) + F_1^{(3)} \left( -\frac{\Delta t}{2} \right). \quad (A12)
\]
Accordingly, the contribution of $F_2^{(3)}(\Delta t)$ to the normalisation factor is
\[
\frac{1}{3} \sum_{i<j<k} |h_i h_j h_k| \Delta t^3 + \frac{1}{12} \sum_{i<j} (|h_i^2 h_j| + 2|h_i h_j^2|) \Delta t^3
\]
\[
< \frac{1}{18} \left( \sum_i |h_i| \right)^3 \Delta t^3. \quad (A13)
\]

Appendix B: Fermi-Hubbard model

The Hamiltonian of Fermi-Hubbard model reads
\[
H_{FH} = - \sum_{i<j} J_{i,j} \sum_{a=\uparrow, \downarrow} (c_{i,a}^\dagger c_{j,a} + c_{j,a}^\dagger c_{i,a})

+ 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), \quad (B1)
\]
where $c_{i,a}$ is the annihilation operator for the fermion with spin-$s$ on the $i$th site. Operators of fermions satisfy $\{c_{i,a}, c_{i',a'}^\dagger\} = 0$ and $\{c_{i,a}^\dagger, c_{i',a'}\} = \delta_{i,i'} \delta_{a,a'} \mathbb{1}$. Here, we modify the original Fermi-Hubbard model by adding a uniform on-site potential $-U/N$, 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 $i + j \in \text{Even}$, 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
\[
c_{i,\uparrow} = \frac{Y_{2i-1} - iZ_{2i-1}}{2} \prod_{l=2i-1} X_l,
\]
\[
c_{i,\downarrow} = \frac{Y_{2i} - iZ_{2i}}{2} \prod_{l=2i} X_l,
\]
where $X_a$, $Y_a$, and $Z_a$ are the Pauli operators of qubit $a$. The spin-$\uparrow$ and the spin-$\downarrow$ on the $i$th site are encoded on the qubits $(2i-1)$ and $2i$, respectively. According to the Jordan-Wigner transformation, the qubit Hamiltonian of
The Fermi-Hubbard model is

\[ H_{\text{FH}} = -\sum_{i<j} \frac{J_{ij}}{2} (Y_{2i-1,2j-1} + Z_{2i-1,2j-1}) + Y_{2i,2j} + Z_{2i,2j} + \frac{U}{4} \sum_i X_{2i-1}X_{2i}, \]  

where

\[ Y_{a,b} = Y_a Y_b \prod_{a<l<b} X_l, \]
\[ Z_{a,b} = Z_a Z_b \prod_{a<l<b} X_l. \]  

Each Pauli operator \( \sigma \) corresponds to an \( x \) binary string according to Eq. (47), and we define \( x(\sigma) \equiv (x_1, \ldots, x_n) \) as the \( x \) binary string of the Pauli operator \( \sigma \). A Hamiltonian does not have short-time interference if \( x(\sigma_1) \neq x(\sigma_2) \) for any pair of Pauli-operator terms \( \sigma_1 \) and \( \sigma_2 \) in the Hamiltonian. We use \( x_{a,b} \) 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,

\[ x(Y_{a,b}) = x_{a,b}, \]
\[ x(Z_{a,b}) = x_{a+1,b-1}, \]
\[ x(X_{2i-1}X_{2i}) = x_{2i-1,2i}. \]  

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, \( \frac{b-a}{2} \) and \( \frac{b'-a'}{2} \) are both odd; however, \( \frac{b'-1-a'-1}{2} \) is even if \( \frac{b'-a'}{2} \) is odd. Therefore, \( x \) strings of \( Y_{a,b} \) and \( Z_{a',b'} \) are always different.