Partition-function estimation: Quantum and quantum-inspired algorithms

Andrew Jackson, Theodoros Kapourniotis, and Animesh Datta
Department of Physics, University of Warwick, Coventry CV4 7AL, United Kingdom

(Received 13 August 2022; accepted 2 December 2022; published 20 January 2023)

We present two algorithms, one quantum and one classical, for estimating partition functions of quantum spin Hamiltonians. The first is a DQC1 (deterministic quantum computation with one clean qubit) algorithm. The second, for real temperatures, achieves performance comparable to a state-of-the-art DQC1 algorithm [A. N. Chowdhury, R. D. Somma, and Y. Subaşi, Phys. Rev. A **103**, 032422 (2021)]. Both our algorithms take as input the Hamiltonian decomposed as a linear combination Pauli operators. We show this decomposition to be DQC1-hard for a given Hamiltonian, providing insight into the hardness of estimating partition functions.

DOI: 10.1103/PhysRevA.107.012421

I. INTRODUCTION

It is hoped that quantum computational devices will make more tractable classically intractable computations required for the study of condensed matter systems, including those in materials science, quantum chemistry [1], and farther afield [2]. Meanwhile, classical computers remain the sole route to unravelling the properties of strongly correlated quantum systems such as the two-dimensional Hubbard model [3] or a chain of hydrogen atoms [4].

Central among such problems is the computation of partition functions, which embody all thermodynamic information of a system at equilibrium. Evaluating partition functions of spin systems exactly is #P-hard [5] in the worst case [6,7]. Their exact evaluation is thus unlikely to be efficient with either classical or quantum devices. The complexity barriers to exact evaluation are often circumvented by aiming for approximations, provided the error can be limited. However, this can still be prohibitively hard [8]. Approximating the partition function for the classical Ising model with complex coefficients, even on the two-dimensional (2D) square lattice, to exponentially large additive (absolute) error, is bounded-error quantum polynomial (BQP)-hard [9,10]. This makes it unlikely to be tractable on classical computers, but may be possible on a BQP device [11,12]. Furthermore, for logarithmically local quantum Hamiltonians, obtaining an approximation (of the partition function) to additive error exponential in both the size (N) of the system and the product of the inverse temperature (β) with the sum of the lowest eigenvalue of each Hamiltonian term is deterministic quantum computation with one clean qubit (DQC1)-hard [13].

Deterministic quantum computation with one clean qubit (DQC1) [14,15] is a complexity class believed to properly contain BPP but not be equivalent to BQP [16–18]. It is therefore not expected to be efficiently simulable classically to within multiplicative (relative) error unless the polynomial hierarchy collapses to the second level [19].

Lately, the concepts of quantum computation have been deployed to develop quantum-inspired classical algorithms [20]. Not only can these often unexpected results be useful in and of themselves [21], but also provide *en passant* insight into the distinctions between classical and quantum computation [22]. They also can be used more readily, rather than waiting for quantum technology to mature.

In this paper, we present a quantum-inspired classical algorithm for approximating partition functions of quantum spin Hamiltonians at real temperatures to multiplicative error. Its complexity is comparable (see Table I) to a state-of-the-art DQC1 algorithm for the same problem [13]. Along the way, we also present a DQC1 algorithm for the same task at complex temperatures to additive error. The Hamiltonian input to our algorithms is expressed as a linear decomposition of Pauli operators with real coefficients. Given a Hamiltonian as a set of circuits implementing each term and the associated coefficients (the form required for input to the algorithm in Ref. [13]), we show the task of obtaining this decomposition to be DQC1-hard in the worst case.

Our paper thus provides an alternative perspective on the hardness of estimating partition functions, one requiring no mention of the sign problem [24]. It is rather based on the hardness of decomposing a Hamiltonian into a linear combination of tensor products of Pauli operators. As the task of simulating quantum spin systems often begins with such a decomposition, their hardness may be dubbed the *decomposition problem*.

II. DQC1 ALGORITHM

The DQC1 model of quantum computation takes as inputs the completely mixed state, a single qubit in the state $|0\rangle$, and a description of a unitary U as illustrated in Fig. 1. Estimating (via sampling) the expectation value of the first qubit with bounded error gives the output: an estimate of the normalized trace of the unitary operator U [14].

The partition function Z for a Hamiltonian $\mathcal{H}$ at a complex inverse temperature $\beta = b_R + ib_I$, $b_R, b_I \in \mathbb{R}$, is given by

$$Z = \text{Tr}(\exp(-\beta\mathcal{H})) = \text{Tr}[\exp(-b_R\mathcal{H})\exp(-ib_I\mathcal{H})]. \quad (1)$$

TABLE I. Lower bound on the expected runtime of our quantum-inspired classical algorithm *vis-a-vis* the DQC1 algorithm from Ref. [13] for approximating the partition function (Z) of an N-qubit Hamiltonian $\mathcal{H}$ consisting of L terms with multiplicative error ϵ_{m}. ξ is the largest absolute value of the coefficients of the Pauli decomposition of $\mathcal{H}$ and β is the inverse temperature. Finally, $C_{\mathcal{H}} \sim \mathcal{O}[\mathrm{polylog}(N)]$ is an overhead for implementing required unitaries in Ref. [13] using a quantum walk operator [23]. The two runtimes being similar despite one algorithm being quantum and the other classical is not a contradiction, from a complexity perspective, as mapping from the input for Ref. [13] to the corresponding input for the algorithm presented herein is DQC1-hard.

Algorithm	Computational model	Lower bound on expected runtime
This paper	Classical (BPP)	$O(\frac{L^2\beta^2\xi^2}{\epsilon_{\mathrm{m}}^2 \ln{(\epsilon_{\mathrm{m}}+1)}}[\frac{e^{L\beta\xi}2^N}{Z}]^2)$
Ref. [13]	Quantum (DQC1)	$O(C_{\mathcal{H}}\frac{L^3\beta^2\xi^2}{\epsilon_{\mathrm{m}}^2}[\frac{e^{L\beta\xi}2^N}{Z}]^2)$

Estimating the partition function using a DQC1 algorithm would require quantum circuits for the operators $\exp(-b_R\mathcal{H})$ and $\exp(-ib_I\mathcal{H})$. These may be thought of as imaginary and real time evolutions, respectively. We implement both approximately using the first-order Trotter decomposition. However, as $\exp(-b_R\mathcal{H})$ is not unitary, we cannot implement it directly and so have to implement it "in the aggregate" by implementing various unitaries with different probabilities to implement it up to an easily calculated factor (see lemma 1 for how this is done). This factor is corrected for by considering its effect on the DQC1 trace estimation algorithm (as in Fig. 1).

Our DQC1 algorithm partition function estimation has two sources of errors: from sampling and from the Trotter decomposition. Their origins lie in the trace estimation and the approximation by the first-order Trotter decomposition of the operators in Eq. (1), respectively, as outlined in the two preceding paragraphs. Denoting these errors in the additive case as ϵ_{aS} and ϵ_{aT}, the total additive error in our algorithm is bounded by $\epsilon_{\mathrm{a}} = \epsilon_{\mathrm{aS}} + \epsilon_{\mathrm{aT}}$.

Given an ϵ_{aT}, the number of Trotter steps ν required to achieve it depends on the specifics of the problem. Let $\mathcal{H} = \sum_{j=1}^{L} h_j\mathfrak{P}_j$ be a N-qubit Hamiltonian, where $h_j \in \mathbb{R}$, each $\mathfrak{P}_j = \mathcal{P}_1 \otimes \mathcal{P}_2 \otimes \cdots \otimes \mathcal{P}_n$, acts on some subset of $n \leqslant N$ qubits labeled by j and $\mathcal{P}_j \in \{\mathcal{I}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ is the set of single-qubit operators. Then the first-order Trotter decomposition of the imaginary time evolution is given by

$$\mathcal{T}_R \approx \exp(-b_R\mathcal{H}), \quad \text{where} \quad \mathcal{T}_R = \left[\prod_{j=1}^{L} \exp(c_j\mathfrak{P}_j)\right]^{\nu}, \tag{2}$$

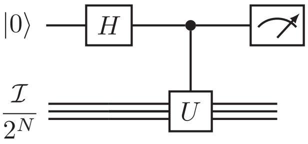

FIG. 1. DQC1 circuit for estimating $\mathrm{Tr}(U)/2^N$ for a N-qubit unitary U. Its real and imaginary parts are obtained by measuring the $\mathcal{X}$ and $\mathcal{Y}$ Pauli operators, respectively, on the first qubit [14].

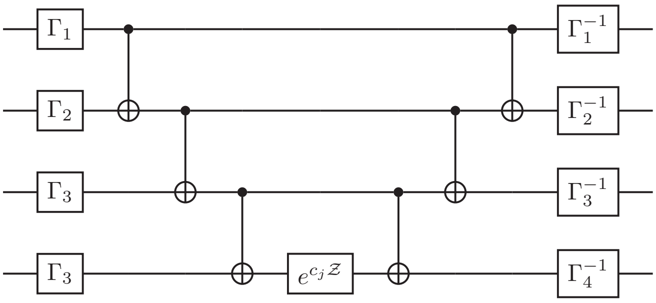

FIG. 2. Gadget implementing the imaginary time (nonunitary) evolution $\exp(-b_R\mathcal{H})$ on $n = 4$ qubits. With each $\Gamma_j \in \{H, R_x(\pi/2)\}$.

where ν is the number of Trotter steps and $c_j = -b_R h_j/\nu$. $\mathcal{T}_I \approx \exp(-ib_I\mathcal{H})$ for the real time evolution is defined similarly with b_R replaced by ib_I (and we use the same number of Trotter steps for convenience). Trotter expansions have been well studied for real time evolutions [25–28]. In the following, we adapt the approach in Ref. [25] for the imaginary time evolution. We restrict ourselves to the first order given the dominance of gate errors over Trotter errors in near-term devices [29].

These approximate operators lead to an approximation of the partition function

$$Z \approx Z_{\mathrm{T}} \equiv \mathrm{Tr}(\mathcal{T}_R\mathcal{T}_I) \quad \text{such that} \quad |Z - Z_{\mathrm{T}}| \leqslant \epsilon_{\mathrm{aT}}. \tag{3}$$

The number of Trotter steps ν required to achieve this Trotter error is given by Theorem 1.

Theorem 1. The number of Trotter steps ν required to implement the operator $\exp(-\beta\mathcal{H})$ using first-order Trotter decomposition with additive Trotter error, in the trace, ϵ_{aT} is

$$\nu = O\left(\frac{1}{\epsilon_{\mathrm{aT}}} 2^N|\beta|^2\Omega^2\exp(b_R\Omega)\right), \tag{4}$$

where $\Omega = \sum_{j=1}^{L} |h_j|$.

The proof is in Appendix A.

Each of these Trotter steps can be implemented as in Ref. [25] (for real time) or via the gadget in Fig. 2 (for the imaginary time), which is based on the methods of the authors of Ref. [25]. The correctness of Fig. 2 is proven in Appendix B. It uses lemma 1, which holds because the Pauli operators square to the identity and is proved below.

Lemma 1. Let $c \in \mathbb{R}$ and $\mathfrak{P}$ ($\mathcal{I}$) be the tensor product of single-qubit Pauli, including identities, (identity) operators on any number of qubits. Then

$$\exp(c\mathfrak{P}) = (\cosh c)\mathcal{I} + \sinh|c|(\Sigma(c)\mathfrak{P}), \tag{5}$$

where $\Sigma(c)$ denotes the sign of c.

Proof. For $r \in \mathbb{R}$, expanding

$$\exp(r\mathfrak{P}) = \sum_{j=0}^{\infty}\left(\frac{(r\mathfrak{P})^j}{j!}\right) \tag{6}$$

$$= \sum_{j=0}^{\infty}\left(\frac{(r\mathfrak{P})^{2j}}{(2j)!} + \frac{(r\mathfrak{P})^{2j+1}}{(2j+1)!}\right) \tag{7}$$

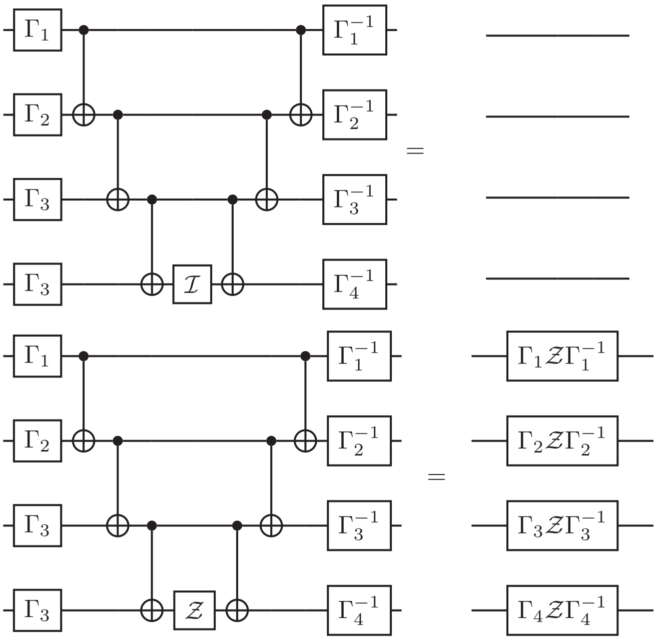

FIG. 3. Quantum circuits corresponding to the two terms in Eq. (5).

$$= \sum_{j=0}^{\infty}\left(\frac{r^{2j}(\mathfrak{P}^2)^j}{(2j)!} + \frac{r^{2j+1}(\mathfrak{P}^2)^j\mathfrak{P}}{(2j+1)!}\right). \tag{8}$$

As $\mathfrak{P}^2 = \mathcal{I}$,

$$\exp(r\mathfrak{P}) = \sum_{j=0}^{\infty}\left(\frac{r^{2j}\mathcal{I}}{(2j)!} + \frac{r^{2j+1}\mathfrak{P}}{(2j+1)!}\right) \tag{9}$$

$$= \sum_{j=0}^{\infty}\left(\frac{r^{2j}}{(2j)!}\right)\mathcal{I} + \sum_{j=0}^{\infty}\left(\frac{r^{2j+1}}{(2j+1)!}\right)\mathfrak{P} \tag{10}$$

$$= (\cosh r)\mathcal{I} + (\sinh r)\mathfrak{P}. \tag{11}$$

As $\cosh c/\exp|c|$ and $\sinh|c|/\exp|c|$ are positive and sum to unity, they may be treated as probabilities. This allows the nonunitary imaginary time evolution $e^{c_j\mathcal{Z}}$ (as in Fig. 2) to be effectively implemented, up to the constant $\exp|c_j|$, by instead applying $\Sigma(c)\mathcal{Z}$ with probability $\sinh|c|$. This implementation of $e^{c_j\mathcal{Z}}$ is exactly what is used in our partition-function estimation algorithm.

Indeed, the quantum circuits corresponding to both terms in Eq. (5) are reducible to single-qubit gates, as illustrated in Fig. 3. The sign $\Sigma(c_j)$ associated with c_j is omitted there and is applied as a phase when implementing the Pauli [as the entire implementation of $\exp(c\mathfrak{P})$ is controlled this is not an overall phase]. This probabilistic implementation of the nonunitary evolution contributes to the sampling error in our algorithm.

In particular, given an ϵ_{aS}, the number of samples ν_{S} (or repetitions of the circuit in Fig. 1) required to achieve it is

$$\nu_{\mathrm{S}} = O\left(\frac{1}{\epsilon_{\mathrm{aS}}^2}2^{2N}\exp(2|b_R|\Omega)\right). \tag{12}$$

Algorithm 1. Algorithm for partition function estimation to multiplicative (relative) error.

```
**Input :** $h_j, \mathfrak{P}_j, L, N, \beta, \nu_{\mathrm{B}}, Z_{\max}, \delta, \epsilon_{\mathrm{mS}}$
Set ApproxNum = 0
ZR = $Z_{\max}$
RequiredTrotterSteps = $\nu_{\mathrm{B}}$
ZRPrime = 0
**while** ZR $\geqslant$ ZRPrime **do**
    ApproxNum++
    ZR = $\frac{Z_{\max}}{2^{\mathrm{ApproxNum}}}$
    absError = $\frac{\epsilon_{\mathrm{mS}}}{2} \times$ ZR
    deltaPrime = $\frac{6\delta}{\pi^2 \times \mathrm{ApproxNum}^2}$
    Fraction = AES($h_j, \mathfrak{P}_j, \beta, \nu_{\mathrm{B}}, L, N, \delta\epsilon_{\mathrm{aS}}$)
    ZRPrime = Fraction × $2^N \exp(\beta\Omega)$
**Output :** ZRPrime
```

The runtime of our DQC1 algorithm is given by the product of the bounds in Eqs. (4) and (12).

III. CLASSICAL ALGORITHM

Our quantum-inspired classical algorithm differs in two respects from the preceding DQC1 algorithm. It takes a *real* inverse temperature $\beta \in \mathbb{R}$ as input and outputs an estimate of the partition function

$$Z = \mathrm{Tr}[\exp(-\beta\mathcal{H})] \tag{13}$$

to *multiplicative* error. This classical algorithm is stated as algorithm 1 (which, in turn, relies on algorithm 2). The classical algorithm is essentially the same as the aforementioned DQC1 algorithm but the insistence on the inverse temperature being real now means all the gates in Figs. 1 and 3 are Clifford (for our purposes, it is most important that the right-hand side of Fig. 3 consists entirely of Pauli gates as $\forall k \in \mathbb{N}$, $\Gamma_k\mathcal{Z}\Gamma_k^{-1}$ reduces to a single Pauli gate, via lemma 3. Consequently, the right-hand side of Fig. 3 can be implemented classically [30,31] and so can a controlled version of Fig. 3 (as required for our algorithm). Again, there are two sources of errors: from the Trotter decomposition and from sampling.

We consider the Trotter error first. If the multiplicative Trotter error in approximating Z by $Z_{\mathrm{T}} = \mathrm{Tr}(\mathcal{T}_R)$ is given by ϵ_{mT}, then

$$|Z - Z_{\mathrm{T}}| \leqslant Z\epsilon_{\mathrm{mT}}, \tag{14}$$

Algorithm 2. Additive error samples (AES).

```
**Input :** $h_j, \mathfrak{P}_j, \beta, \nu_{\mathrm{B}}, L, N, \delta, \epsilon_{\mathrm{aS}}$
Set SAMPLES = $\emptyset$
numSamples = $\lceil -2\ln(\delta)/\epsilon_{abs}^2 \rceil$
$k = 1$
**for** $k \leqslant$ numSamples **do**
    SAMPLE = SSA($h_j, \mathfrak{P}_j, \beta, \nu_{\mathrm{B}}, L, N$)
    Add SAMPLE to SAMPLES
    $k+ = 1$
**Output :** Average of SAMPLES
```

Algorithm 3. Single Sample Algorithm (SSA).

```
Input : $h_j, \mathfrak{P}_j, \beta, \nu_B, L, N$
Set CIRCUIT = $\mathcal{I}$.
while ($k \leqslant \nu_B$) do
    for ($j \leqslant L$) do
        RAND = uniformly random number in [0, 1]
        if (RAND $\geqslant \frac{\sinh |c_j|}{e^{|c_j|}}$) then
            CIRCUIT = CIRCUIT × $\Sigma(c_j)\mathfrak{P}_j$
DQC1 = Circuit in Fig. 1 applied with $U$ = CIRCUIT
INITIAL = uniformly selected $N$-bit binary string.
SAMPLE = Outcome of classically evaluating DQC1 on INITIAL.
Output : SAMPLE
```

and the number of Trotter steps ν is determined by theorem 2. Its proof is in Appendix C and uses methods from Ref. [32].

To fully exploit mutual noncommutativity among the terms of $\mathcal{H}$, we require the notion of the noncommuting set [32].

Definition 1 (kth noncommuting set). The kth noncommuting set ζ_k is the set of all terms in the Hamiltonian $\mathcal{H} = \sum_{j=1}^{L} h_j \mathfrak{P}_j$ that do not commute with the kth term of the Hamiltonian $\mathfrak{P}_k$.

Theorem 2. The number of Trotter steps ν required to approximate $Z = \mathrm{Tr}[\exp(-\beta\mathcal{H})]$ up to multiplicative error ϵ_{mT} using the first-order Trotter decomposition is

$$\nu \geqslant \frac{\beta^2 \Omega \mathfrak{h}}{\ln(1+\epsilon_{\mathrm{mT}})}, \tag{15}$$

where $\Omega = \sum_{j=1}^{L} |h_j|$ and $\mathfrak{h} = \sum_{k=2}^{L} |h_k| N_k$ with $N_k = |\{z \leqslant k$ such that $z \in \zeta_k\}|$.

For the sampling error, we follow Ref. [13, algorithm 3] (For completeness, we present it as algorithm 1.). It shows that estimates with successively smaller additive error lead to an estimate to within the required multiplicative error, with probability as high as required. If the multiplicative sampling error is ϵ_{mS} and $Z_{\max}$ an upper bound on the value of the partition function, then this algorithm provides an estimate Z_{S} of Z_{T} in Eq. (14) such that

$$\Pr(|Z_{\mathrm{T}} - Z_{\mathrm{S}}| \leqslant Z_{\mathrm{T}}\epsilon_{\mathrm{mS}}) \geqslant 1 - \delta, \tag{16}$$

where $\delta > 0$ is the upper bound on the probability of obtaining an estimate beyond the precision ϵ_{mS}. The runtime of this algorithm is a random variable with an expected value of [13]

$$T_{\mathrm{S}} = O\left[\frac{2^{2N}\exp(2\beta\Omega)}{\epsilon_{\mathrm{mS}}^2 Z^2} \log_2\left(\frac{1}{\delta}\right) \log_2\left(\frac{Z_{\max}}{Z_{\mathrm{T}}}\right)\right]. \tag{17}$$

The multiplicative Trotter and sampling errors in our estimation algorithm for the partition function combine to give a total multiplicative error ϵ_{m} such that

$$\epsilon_{\mathrm{m}} = \epsilon_{\mathrm{mS}} + \epsilon_{\mathrm{mT}} + \epsilon_{\mathrm{mS}}\epsilon_{\mathrm{mT}}. \tag{18}$$

This follows from Eqs. (14) and (16), the positivity of the partition functions, their estimates, and the errors. Using this in conjunction with Eqs. (15) and (17) and neglecting the $\log_2(\cdot)$ contributions gives the total expected runtime of our classical algorithm as in Table I. It is comparable to that of the DQC1 algorithm in Ref. [13], despite the belief that DQC1 is more powerful than classical computation [14–18]. This is because the two algorithms take as input the Hamiltonian in two different forms which encode different degrees of hardness, as we will show in Sec. IV

To complete this section, we formally present our classical algorithm. Our algorithm takes as inputs a N-qubit Hamiltonian $\mathcal{H} = \sum_{j=1}^{L} h_j \mathfrak{P}_j$ described by $h_j, \mathfrak{P}_j, L$, a real inverse temperature β, the multiplicative sampling error ϵ_{mS}, the upper bound on the probability δ of obtaining an estimate beyond this precision, the number of Trotter steps ν_{B} given by the RHS of [Eq. (15)] achieving a designated multiplicative Trotter error in the estimate of the partition function, as well as its maximum value $Z_{\max}$. The methods of obtaining multiplicative error bounds from additive error bounds is identical to algorithm 3 in Ref. [13], except for the exponential factors in the last line.

IV. HARDNESS OF PARTITION FUNCTION ESTIMATION

Our classical algorithm requires the Hamiltonian input as a linear combination of tensor products of Pauli operators. This is a special case of the Hamiltonian input as a linear combination of unitary operators, each of which can be implemented by an efficient quantum circuit, which is the input format Ref. [13] requires. We now show that given a Hamiltonian, as a linear combination of unitary operators with the associated circuit, obtaining the same Hamiltonian as a linear combination of tensor products of Pauli operators (what our algorithm takes as input) is DQC1-hard. This shows that part of the hardness of estimating partition functions of quantum spin Hamiltonians may be ascribed to a *decomposition problem.*

We begin with a formal statement of the decomposition problem.

Definition 2 (Obtaining an H decomposition). Given an N-qubit Hermitian operator H as a set of poly(N) positive coefficients c_i and unitary operators U_i, the latter described by poly(N)-sized quantum circuits each such that $H = \sum_i c_i U_i$, a N-qubit Pauli operator σ, $\delta \geqslant 0$, $\delta = \mathrm{poly}(n)$, and a bit Re or Im, the goal is to output either $\hat{r}$ or $\hat{t}$, depending on whether the bit is Re or Im, such that

$$|\hat{r} - \mathrm{Re}(\mathrm{Tr}(\sigma H))| \leqslant \delta, \tag{19}$$

$$|\hat{t} - \mathrm{Im}(\mathrm{Tr}(\sigma H))| \leqslant \delta. \tag{20}$$

We also define a problem of known hardness: that of estimating up to polynomial additive error the real or imaginary part of the coefficient of a unitary in its Pauli expansion.

Definition 3 (Obtaining a U decomposition). Given an N-qubit unitary operator U as a poly(N)-sized quantum circuit, an N-qubit Pauli operator σ, $\delta \geqslant 0$, $\delta = \mathrm{Poly}(n)$, and a bit Re or Im, the goal is to output either $\hat{r}$ or $\hat{t}$, depending on whether the bit is Re or Im, such that

$$|\hat{r} - \mathrm{Re}[\mathrm{Tr}(\sigma U)]| \leqslant \delta, \tag{21}$$

$$|\hat{t} - \mathrm{Im}[\mathrm{Tr}(\sigma U)]| \leqslant \delta. \tag{22}$$

Theorem 3 ([14]). Obtaining a U decomposition is DQC1-hard.

We prove that obtaining a U decomposition can be polynomially transformed into obtaining a H decomposition, thus the hardness of the latter.

Theorem 4. Obtaining a U decomposition $\leqslant_p$ Obtaining an H decomposition. Therefore, obtaining an H decomposition is DQC1-hard.

Proof. It suffices to provide a polynomial-time algorithm for transforming the inputs to the problem of obtaining a U decomposition into the inputs of obtaining an H decomposition problem, such that the transformed problem has same output as the original. This algorithm does the following: It takes as input the circuit description of U. It generates the description of $U^\dagger$ by reversing the circuit and conjugating all the gates. In the case we are asked to calculate the real part of the coefficient corresponding to σ, the algorithm generates the output

$$\{\{\tfrac{1}{2}, U\}, \{\tfrac{1}{2}, U^\dagger\}\}, \sigma, \delta, \text{Re}. \tag{23}$$

This is a valid input to the problem of obtaining a H-decomposition as $\frac{1}{2}(U + U^\dagger)$ is Hermitian and the descriptions of the unitary operators are of polynomial size. We also have

$$\text{Re}\left[\text{Tr}\left(\sigma\frac{U+U^\dagger}{2}\right)\right] = \frac{\text{Re}[\text{Tr}(\sigma U)] + \text{Re}\{\text{Tr}[(U\sigma)^\dagger]\}}{2} \tag{24}$$

$$= \frac{\text{Re}[\text{Tr}(\sigma U)] + \text{Re}[\text{Tr}(\sigma U)^*]}{2} \tag{25}$$

$$= \text{Re}[\text{Tr}(\sigma U)]. \tag{26}$$

Thus, the output of the transformed problem will be the same estimate with the same precision as in the original problem.

Similarly, if we are asked to calculate the imaginary part of the coefficient that corresponds to Pauli σ, the algorithm generates the output

$$\left\{\left\{-\frac{i}{2}, U\right\}, \left\{\frac{i}{2}, U^\dagger\right\}\right\}, \sigma, \delta,$$

$$\text{A bit specifying the real or imaginary part.} \tag{27}$$

Again, this is a valid input to the problem of obtaining a H decomposition because $\frac{i}{2}(U^\dagger - U)$ is Hermitian and the descriptions of the unitary operators are of polynomial size. We also have

$$\text{Re}\left[\text{Tr}\left(i\sigma\frac{U^\dagger - U}{2}\right)\right]$$

$$= \frac{\text{Re}[-i\text{Tr}(\sigma U)] + \text{Re}\{\text{Tr}[(-iU\sigma)^\dagger]\}}{2} \tag{28}$$

$$= \frac{\text{Re}[\text{Tr}(-i\sigma U)] + \text{Re}[\text{Tr}(-i\sigma U)^*]}{2} \tag{29}$$

$$= \text{Im}[\text{Tr}(\sigma U)], \tag{30}$$

and the same reasoning as before applies for the output. The hardness follows directly from theorem 3. ■

V. NUMERICAL RESULTS

As the purpose of an algorithm is to solve a problem, we now present results of numerical investigations into our classical partition function estimation algorithm (algorithm 1). To recapitulate, our classical algorithm is designed for real temperatures and estimates the partition function up to a multiplicative error.

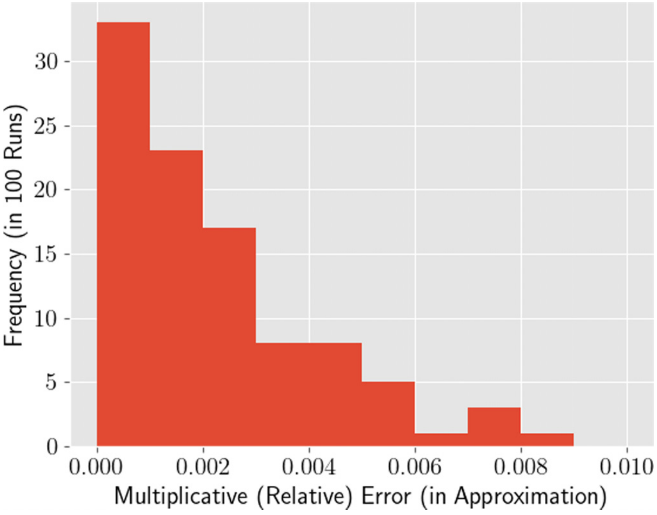

FIG. 4. Histogram of the multiplicative (relative) error in our algorithm's estimation of 100 randomly generated Hamiltonians on between one and three spins.

We first experimentally verify the correctness of our algorithm 1, although it follows formally from Ref. [13]. To that end, we generate 100 Hamiltonians with (uniformly in [−1,1]) random coefficients h_j of up to $L = 4$ random Pauli terms each acting on up to $N = 3$ spins. We then estimate their partition functions at random real inverse temperatures β using our algorithm 1 for $\epsilon_{\text{mS}} = \epsilon_{\text{mT}} = 0.048$, $\delta = 0.15$, and Z_{max} was set to twice the true value of the partition function. The latter was obtained, up to numerical error, using full diagonalization. Comparing the estimate and the true values shows that our algorithm 1 indeed produces estimates well within $\epsilon_{\text{m}} \approx 0.098$ of the exact value, as shown in Fig. 4.

To illustrate the performance of our algorithm in a problem of interest, we resort to the two-dimensional (2D) one-band Fermi-Hubbard model, solutions of which have been widely studied using different numerical algorithms [3]. Its simplest rendition is given by

$$\mathcal{H} = -t\sum_{\substack{\langle i,j\rangle \\ \sigma\in\{\uparrow,\downarrow\}}}(\hat{C}^\dagger_{i,\sigma}\hat{C}_{j,\sigma} + \hat{C}^\dagger_{j,\sigma}\hat{C}_{i,\sigma}) + U\sum_k \hat{n}_{k,\uparrow}\hat{n}_{k,\downarrow}, \tag{31}$$

where $\langle i, j\rangle$ indicates adjacent vertices i, j of a 2D graph, $\hat{C}^\dagger_{i,\sigma}, \hat{C}_{i,\sigma}$, and $\hat{n}_{i,\sigma} = \hat{C}^\dagger_{i,\sigma}\hat{C}_{i,\sigma}$ denote Fermionic creation, annihilation, and number operators, respectively, for spin σ at vertex i of the graph. t denotes the nearest-neighbor hopping strength and U the onsite interaction strength. It is typical to set the energy scales in the Hamiltonian in terms of t. Thus, the dimensionless onsite interaction strength is given by $\tilde{U} = U/t$ and the dimensionless inverse temperature by $\tilde{\beta} = \beta t$.

Being inspired by a quantum algorithm (Ref. [13]), the above Hamiltonian must undergo a Fermion to qubit mapping before being fed into our algorithm 1. We use a recent low-weight encoding [33] for this. For a 3 × 3 square lattice, this leads to a Hamiltonian on $N = 26$ qubits with $L = 60$ terms.

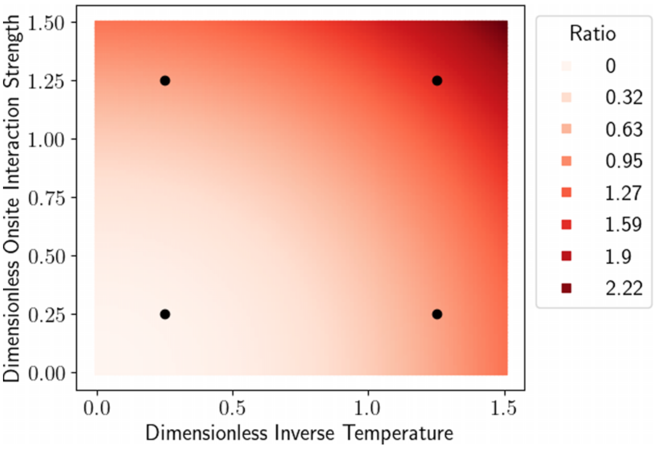

FIG. 5. Numerical test of time requirements. Darker red indicates a greater value of the ratio of the time our algorithms takes compared to full diagonalisation using ALPS. The black dots indicate the value is given in Table II. These numbers were obtained using a Dell PowerEdge C6420 with 2 x Intel Xeon Platinum 8268 (Cascade Lake) 2.9 GHz 24-core processors with 192 GB of (DDR4-2933) RAM.

The projected ratio between the time to estimate the partition function using our algorithm 1 (with $\epsilon_{\mathrm{mS}} = \epsilon_{\mathrm{mT}} = 0.15$, $\delta = 0.20$) and the time to estimate using full diagonalization in ALPS [34], is shown in Fig. 5. This enables a comparison of the performance of our algorithm 1 against a recognised benchmark, as illustrated in Table II. In line with keeping this comparison valid, the choice of both the Hubbard model and the low-weight encoding are not made to benefit either our algorithm or exact diagonalization. Both are chosen as an illustration of a scenario relevant for quantum simulation.

Unsurprisingly, in Fig. 5 and Table II, larger $\tilde{\beta}$ and $\tilde{U}$ results in our algorithm performing comparatively worse. Furthermore, an increase in $\tilde{\beta}$ has a larger effect than an equivalent one in $\tilde{U}$. The latter follows from Eq. (15) where an increase in $\tilde{U}$ only increases some of the terms in Ω and $\mathfrak{h}$, whereas an equivalent increase in $\tilde{\beta}$ is tantamount to increasing all the terms in Ω and $\mathfrak{h}$.

As to the space requirements, our algorithm requires memory scaling linearly in N. This is favorable compared to full diagonalization, which requires processing matrices exponentially large in N and is thus very memory-intensive.

VI. CONCLUSION

Our paper provides a different perspective on the root of the hardness of estimating partition functions. In this view, the hardness of estimating partition functions may be due to the *decomposition problem*. This vantage may provide useful insights into the complexity of simulating hard quantum problems, as has been the case for the sign problem [35,36]. For instance, the exponential contributions in our runtime (Table I) arise from sampling [Eq. (17)]. These, and the runtime as a whole, are independent of the sign of the Hamiltonian in certain bases. Thus our classical algorithm may be exponential even for sign-problem-free systems. However, it could be significantly more efficient when the partition function is large. Low temperatures, of typical interest, do not correspond to such a scenario. Endeavors to explore other Hamiltonians and parameters could be worthwhile given the general hardness and importance of estimating partition functions.

TABLE II. The ratio of the time our algorithms takes compared to full diagonalization using ALPS for the four marked points in Fig. 5.

$\tilde{U}$ \ $\tilde{\beta}$	0.25	1.25
1.25	0.0522	1.3043
0.25	0.0215	0.5351

ACKNOWLEDGMENTS

We are grateful to Andreas Honecker and David Quigley for sharing their expertise. This work was supported, in part, by the UK Networked Quantum Information Technologies (NQIT) Hub (Grant No. EP/M013243/1), the UKRI ExCALIBUR project QEVEC (Project No. EP/W00772X/2), and a Leverhulme Trust Early Career Fellowship. We acknowledge the use of the Scientific Computing Research Technology Platform at the University of Warwick.

APPENDIX A: PROOF OF THEOREM 1

Following the main text, let $\mathcal{T}_R$, $\mathcal{T}_I$ be the first-order Trotter approximations to $e^{-b_R\mathcal{H}}$, $e^{-ib_I\mathcal{H}}$, respectively, with ν Trotter steps. Denote

$$\Delta = |\mathrm{Tr}(e^{-\beta\mathcal{H}}) - \mathrm{Tr}(\mathcal{T}_R\mathcal{T}_I)|. \tag{A1}$$

Then,

$$\begin{aligned}\Delta &= |\mathrm{Tr}(e^{-\beta\mathcal{H}}) - \mathrm{Tr}(\mathcal{T}_R\mathcal{T}_I) + \mathrm{Tr}(e^{-b_R\mathcal{H}}\mathcal{T}_I) - \mathrm{Tr}(e^{-b_R\mathcal{H}}\mathcal{T}_I)|\\ &= |\mathrm{Tr}(e^{-b_R\mathcal{H}}[e^{-ib_I\mathcal{H}} - \mathcal{T}_I]) + \mathrm{Tr}([-\mathcal{T}_R + e^{-b_R\mathcal{H}}]\mathcal{T}_I)|\end{aligned} \tag{A2}$$

$$\leqslant |\mathrm{Tr}(e^{-b_R\mathcal{H}}[e^{-ib_I\mathcal{H}} - \mathcal{T}_I])| + |\mathrm{Tr}([\mathcal{T}_R - e^{-b_R\mathcal{H}}]\mathcal{T}_I)|. \tag{A3}$$

Denoting $||\cdot||$ as the spectral norm (the largest singular value)

$$\begin{aligned}\Delta &\leqslant 2^N||e^{-b_R\mathcal{H}}[e^{-ib_I\mathcal{H}} - \mathcal{T}_I]|| + 2^N||[\mathcal{T}_R - e^{-b_R\mathcal{H}}]\mathcal{T}_I||\\ &\leqslant 2^N||e^{-b_R\mathcal{H}}||\,||e^{-ib_I\mathcal{H}} - \mathcal{T}_I|| + 2^N||\mathcal{T}_R - e^{-b_R\mathcal{H}}||,\end{aligned} \tag{A4}$$

where we used the submultiplicative property of the spectral norm and $||\mathcal{T}_I|| = 1$ as $\mathcal{T}_I$ is a unitary. Lemma 2 shows that

$$||\exp(-ib_I\mathcal{H}) - \mathcal{T}_I|| \leqslant \nu\left|\left|\exp\left(-i\frac{b_I}{\nu}\mathcal{H}\right) - \mathcal{T}_{I,1}\right|\right|, \tag{A5}$$

$$||\mathcal{T}_R - \exp(-b_R\mathcal{H})|| \leqslant \nu\exp\left[\left(1 - \frac{1}{\nu}\right)|b_R|\Omega\right]\left|\left|\mathcal{T}_{R,1} - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right|, \tag{A6}$$

where $\mathcal{T}_{R,1} = [\mathcal{T}_R]^{1/\nu} = \prod_{j=1}^{L} \exp(c_j \mathfrak{P}_j)$ denotes one Trotter step for the imaginary time evolution, $\mathcal{T}_{I,1}$ similarly denotes one Trotter step for the real time evolution, and $\Omega = \sum_{j=1}^{L} ||h_j \mathfrak{P}|| = \sum_{j=1}^{L} |h_j|$.

Using these with lemma 1 in Ref. [32] for the first-order Trotter expansion

$$||\mathcal{T}_R - \exp(-b_R\mathcal{H})|| = O\left(\frac{b_R^2}{\nu}\Omega^2 \exp(b_R\Omega)\right) \text{ and}$$
$$||\exp(-ib_I\mathcal{H}) - \mathcal{T}_I|| = O\left(\frac{b_I^2}{\nu}\Omega^2\right). \tag{A7}$$

Substituting these in Eq. (A4), and using Eq. (A18)

$$\Delta \leqslant O\left(\frac{1}{\nu}2^N\Omega^2|\beta|^2 \exp(b_R\Omega)\right) \leqslant \epsilon_{\text{aT}}, \tag{A8}$$

where the last inequality uses Eq. (3). Thus,

$$\nu \geqslant O\left(\frac{1}{\epsilon_{\text{aT}}}2^N\Omega^2|\beta|^2 \exp(b_R\Omega)\right). \tag{A9}$$

Lemma 2. For a Hamiltonian $\mathcal{H} = \sum_{j=1}^{L} h_j\mathfrak{P}$ and $b_R, b_I \in \mathbb{R}$,

$$||\mathcal{T}_R - \exp(-b_R\mathcal{H})|| \leqslant \nu \exp\left[\left(1 - \frac{1}{\nu}\right)|b_R|\Omega\right]\left|\left|\mathcal{T}_{R,1} - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right|, \tag{A10}$$

$$||\exp(-ib_I\mathcal{H}) - \mathcal{T}_I|| \leqslant \nu\left|\left|\exp\left(-i\frac{b_I}{\nu}\mathcal{H}\right) - \mathcal{T}_{I,1}\right|\right|, \tag{A11}$$

where $\Omega = \sum_{j=1}^{L} ||h_j\mathfrak{P}|| = \sum_{j=1}^{L} |h_j|$.

Proof. Using the triangle inequality and submultiplicativity of the spectral norm

$$||\mathcal{T}_R - \exp(-b_R\mathcal{H})|| = ||[\mathcal{T}_{R,1}]^\nu - \exp(-b_R\mathcal{H})|| \tag{A12}$$

$$= \left|\left|[\mathcal{T}_{R,1}]^\nu - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)[\mathcal{T}_{R,1}]^{\nu-1} + \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)[\mathcal{T}_{R,1}]^{\nu-1} - \exp(-b_R\mathcal{H})\right|\right| \tag{A13}$$

$$\leqslant \left|\left|[\mathcal{T}_{R,1}]^\nu - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)[\mathcal{T}_{R,1}]^{\nu-1}\right|\right| + \left|\left|\exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)[\mathcal{T}_{R,1}]^{\nu-1} - \exp(-b_R\mathcal{H})\right|\right| \tag{A14}$$

$$\leqslant ||[\mathcal{T}_{R,1}]^{\nu-1}||\left|\left|\mathcal{T}_{R,1} - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right| + \left|\left|\exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right|\left|\left|[\mathcal{T}_{R,1}]^{\nu-1} - \exp\left(-b_R\left(1-\frac{1}{\nu}\right)\mathcal{H}\right)\right|\right|. \tag{A15}$$

Recursively applying the above procedure for the second term

$$||\mathcal{T}_R - \exp(-b_R\mathcal{H})|| \leqslant \left(\sum_{\substack{j,k\geqslant 0\\ j+k=\nu-1}} \left|\left|\exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right|^j ||\mathcal{T}_{R,1}||^k\right)\left|\left|\mathcal{T}_{R,1} - \exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right|. \tag{A16}$$

Since $||\mathfrak{P}|| = 1$ and $c_j = -b_R h_j/\nu$,

$$||\mathcal{T}_{R,1}|| = \left|\left|\prod_{j=1}^{L}\exp(c_j\mathfrak{P})\right|\right| \leqslant \prod_{j=1}^{L}||\exp(c_j\mathfrak{P})|| \leqslant \prod_{j=1}^{L}\exp(|c_j|) = \exp\left(\sum_{j=1}^{L}|c_j|\right) = \exp\left(\frac{|b_R|}{\nu}\Omega\right), \tag{A17}$$

and

$$\left|\left|\exp\left(-\frac{b_R}{\nu}\mathcal{H}\right)\right|\right| \leqslant \exp\left(\frac{|b_R|}{\nu}\Omega\right). \tag{A18}$$

Using the above bounds on each of the terms in the sum of Eq. (A16) gives the final result. The result for the real time evolution follows similarly, albeit more simply as $||\mathcal{T}_{I,1}|| = 1 = ||\exp(-i\frac{b_I}{\nu}\mathcal{H})||$. ■

APPENDIX B: IMPLEMENTING IMAGINARY TIME EVOLUTION

In this Appendix, we show that gadgets in Fig. 2, analogous to those in Ref. [25], can be used to implement imaginary time evolutions such as $\exp(b_R\mathcal{H})$.

1. Overview and preparatory lemmas

We denote by $\mathcal{P} = \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ the set of single-qubit Pauli operators, by $\mathcal{P}_j \in \mathcal{P}$ a Pauli operator on qubit j, and by $\Gamma_j \in \{H, R_x(\pi/2)\}$ the single-qubit Hadamard gate and that which implements a rotation about the x axis by π. Then

Lemma 3. $\mathcal{X} = H\mathcal{Z}H$ and $\mathcal{Y} = R_x(\pi/2)\mathcal{Z}R_x(-\pi/2)$.

Proof. Follows from matrix multiplication. ■

Lemma 4. For any real or complex number h_j, given a gadget for the time evolution of the Hamiltonian $\mathcal{H}_1 = h_j \cdot \mathcal{Z}_1 \otimes \mathcal{Z}_2 \otimes \mathcal{Z}_3 \otimes \cdots \otimes \mathcal{Z}_n$, on n qubits, the time evolution of any Hamiltonian $\mathcal{H}_2 = h_j \cdot \mathcal{P}_1 \otimes \mathcal{P}_2 \otimes \cdots \otimes \mathcal{P}_n$ of Pauli operators may be implemented through the addition of the single-qubit gates Γ.

Proof. As each Pauli operator acts on a different qubit, they mutually commute, giving

$$e^{\mathcal{H}_2} = e^{h_j \cdot \mathcal{P}_1 \otimes \mathcal{P}_2 \otimes \cdots \otimes \mathcal{P}_n} = \sum_{k=0}^{\infty}\left(\frac{(h_j \cdot \mathcal{P}_1\mathcal{P}_2\cdots\mathcal{P}_n)^k}{k!}\right) = \sum_{k=0}^{\infty}\left(\frac{(h_j)^k(\mathcal{P}_1)^k(\mathcal{P}_2)^k\cdots(\mathcal{P}_n)^k}{k!}\right), \tag{B1}$$

where we suppressed the $\otimes$ for brevity. Using lemma 3, $\mathcal{P}_i = \Gamma_i\mathcal{Z}_i\Gamma_i^{-1}$, gives

$$\begin{aligned} e^{\mathcal{H}_2} &= \sum_{k=0}^{\infty}\left(\frac{(h_j)^k\left(\Gamma_1\mathcal{Z}_1\Gamma_1^{-1}\right)^k\cdots\left(\Gamma_n\mathcal{Z}_n\Gamma_n^{-1}\right)^k}{k!}\right) = \sum_{k=0}^{\infty}\left(\frac{(h_j)^k\Gamma_1(Z_1)^k\Gamma_1^{-1}\cdots\Gamma_n(\mathcal{Z}_n)^k\Gamma_n^{-1}}{k!}\right) \\ &= \sum_{k=0}^{\infty}\left(\Gamma_1\cdots\Gamma_n\frac{(h_j)^k(Z_1)^k\cdots(\mathcal{Z}_n)^k}{k!}\Gamma_1^{-1}\cdots\Gamma_n^{-1}\right) = \Gamma_1\cdots\Gamma_n\sum_{k=0}^{\infty}\left(\frac{\left(h_j\cdot\mathcal{Z}_1\cdots\mathcal{Z}_n\right)^k}{k!}\right)\Gamma_1^{-1}\cdots\Gamma_n^{-1} \\ &= (\Gamma_1\cdots\Gamma_n)e^{\mathcal{H}_1}\left(\Gamma_1^{-1}\cdots\Gamma_n^{-1}\right), \end{aligned} \tag{B2}$$

where each Γ_j can be identified and implemented efficiently a single gate. ■

To obtain a quantum circuit for implementing $e^{\mathcal{H}_2}$, we begin with a quantum circuit for implementing $e^{\mathcal{H}_1}$.

2. Implementing $e^{\mathcal{H}_1}$

We begin with a quantum circuit for implementing $e^{\mathcal{H}_1}$ for just two qubits.

Lemma 5. The quantum circuit in Fig. 6 implements $e^{\mathcal{H}}$ where $\mathcal{H} = h_j \cdot \mathcal{Z}_1 \otimes \mathcal{Z}_2$.

Proof. Considering the matrix representation of the circuit in the computational basis gives

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}\left(\mathcal{I} \otimes \begin{bmatrix} e^{h_j} & 0 \\ 0 & e^{-h_j} \end{bmatrix}\right)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} e^{h_j} & 0 & 0 & 0 \\ 0 & e^{-h_j} & 0 & 0 \\ 0 & 0 & e^{-h_j} & 0 \\ 0 & 0 & 0 & e^{h_j} \end{bmatrix}, \tag{B3}$$

where $\mathcal{I}$ denotes the identity gate. This is identical to

$$e^{\mathcal{H}} = \begin{bmatrix} e^{h_j} & 0 & 0 & 0 \\ 0 & e^{-h_j} & 0 & 0 \\ 0 & 0 & e^{-h_j} & 0 \\ 0 & 0 & 0 & e^{h_j} \end{bmatrix}, \quad \text{where } \mathcal{H} = h_j \cdot \mathcal{Z}_1 \otimes \mathcal{Z}_2 = \begin{bmatrix} h_j & 0 \\ 0 & -h_j \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} h_j & 0 & 0 & 0 \\ 0 & -h_j & 0 & 0 \\ 0 & 0 & -h_j & 0 \\ 0 & 0 & 0 & h_j \end{bmatrix}. \tag{B4}$$

■

Lemma 6. Let $\mathcal{U}_n = e^{h_j \cdot \mathcal{H}_1}$ act on n qubits, and Cx denote a CNOT gate with the target on qubit n and the control on a new qubit labeled $n + 1$ then

$$\mathcal{U}_{n+1} = Cx\mathcal{U}_nCx. \tag{B5}$$

Proof. Denoting $\mathcal{L}_{h_j} = \cosh(h_j)$ and $\mathcal{S}_{h_j} = \sinh(h_j)$, $\mathcal{U}_n = e^{h_j}(\mathcal{L}_{h_j}\mathcal{I}_1\mathcal{I}_2\cdots\mathcal{I}_n + \mathcal{S}_{h_j}\mathcal{Z}_1\mathcal{Z}_2\cdots\mathcal{Z}_n)$. For each of these two terms, the circuit identities in Fig. 7 show that conjugating them with CNOTs increases n by 1.

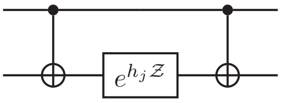

FIG. 6. Quantum circuit to implement $e^{\mathcal{H}}$ where $\mathcal{H} = h_j \cdot \mathcal{Z}_1 \otimes \mathcal{Z}_2$.

3. Implementing $e^{\mathcal{H}_2}$

A quantum circuit implementing $e^{\mathcal{H}_2}$ can be obtained by combining lemmas 6 and 4. It leads to circuits of the form of Fig. 8, reminiscent of Ref. [25].

APPENDIX C: PROOF OF THEOREM 2

This Appendix uses methods from Ref. [32] to prove Theorem 2. Other methods [37,38] may also be used.

1. Preparatory lemmas

Lemma 7 (Inverse product). For a sequence of operators $\{A_1, \cdots, A_L\}$ and $j, k \in \{1, \cdots, L\}$,

$$\prod_{j=k}^{L}(e^{\tau A_j})\prod_{j=L}^{1}(e^{-\tau A_j}) = \prod_{j=k-1}^{1}(e^{-\tau A_j}). \tag{C1}$$

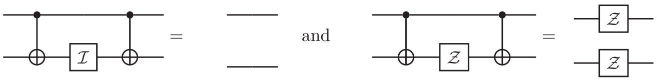

FIG. 7. Circuit identities for conjugation by CNOTs.

Proof. Noting that

$$\prod_{j=k}^{L}(e^{\tau A_j})\prod_{j=L}^{1}(e^{-\tau A_j}) = \prod_{j=k}^{L-1}(e^{\tau A_j})\, e^{\tau A_L}e^{-\tau A_L}\prod_{j=L-1}^{1}(e^{-\tau A_j})$$
$$= \prod_{j=k}^{L}(e^{\tau A_j})\prod_{j=L-1}^{1}(e^{-\tau A_j}), \tag{C2}$$

the result follows by repeatedly redefining L, and iterating. ■

Lemma 8 (Lemma A.3 of Ref. [32]). Let $\mathcal{U}(t)$ be an invertible and continuously differentiable operator-valued function for $t \in \mathbb{R}$. Then there exists $\mathcal{G}(t)$ such that

$$\mathcal{U}(t) = \exp\left\{\int_0^t(\mathcal{G}(\tau))d\tau\right\}\mathcal{U}(0), \quad \text{and}$$
$$\mathcal{G}(\tau) = \frac{d}{d\tau}(\mathcal{U}(\tau))\mathcal{U}^{-1}(\tau). \tag{C3}$$

Proof. Using the chain rule on $\mathcal{U}(t)$ which is invertible and continuously differentiable, let

$$\mathcal{G}(\tau) = \frac{d}{d\tau}(\ln(\mathcal{U}(\tau))) = \frac{d}{d\tau}(\mathcal{U}(\tau))\mathcal{U}^{-1}(\tau). \tag{C4}$$

Then

$$\int_0^t(\mathcal{G}(\tau))d\tau = \int_0^t\left[\frac{d}{d\tau}(\ln(\mathcal{U}(\tau)))\right]d\tau$$
$$= \ln[\mathcal{U}(t)] - \ln[\mathcal{U}(0)], \tag{C5}$$

whereby $\exp\{\int_0^t(\mathcal{G}(\tau))d\tau\}\mathcal{U}(0) = \exp\{\ln(\mathcal{U}(t)) - \ln[\mathcal{U}(0)]\}\mathcal{U}(0)$. As $\mathcal{U}(t)$ is invertible, $\ln[\mathcal{U}(t)] - \ln[\mathcal{U}(0)] = \ln[\mathcal{U}(t)]\,\mathcal{U}^{-1}(0)$ and $\exp\{\int_0^t(\mathcal{G}(\tau))d\tau\}\mathcal{U}(0) = \exp\{\ln[\mathcal{U}(t)\mathcal{U}^{-1}(0)]\}\mathcal{U}(0) = \mathcal{U}(t)$. ■

Lemma 9 (Lemma A.2 of Ref. [32]). If $\mathcal{A}(\beta)$ and $\mathcal{B}(\beta)$ are continuous operators, then

$$\exp\left\{\int_0^\beta[\mathcal{A}(\tau)+\mathcal{B}(\tau)]d\tau\right\}$$
$$= \exp\left\{\int_0^\beta\mathcal{A}(\tau)d\tau\right\}\cdot\exp\left\{\int_0^\beta\left(\exp\left\{-\int_0^{\tau_1}\mathcal{A}(\tau_2)d\tau_2\right\}\right.\right.$$
$$\left.\left.\cdot\ \mathcal{B}(\tau_1)\cdot\exp\left\{\int_0^{\tau_1}[\mathcal{A}(\tau_2)]d\tau_2\right\}\right)d\tau_1\right\}. \tag{C6}$$

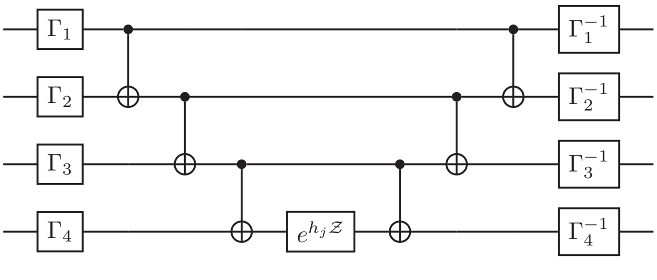

FIG. 8. Nonunitary gadget for four qubits.

2. From operator Trotter error to scalar multiplicative (relative) error

Lemma 10. Let $\mu_j(\cdot)$, and $\sigma_j(\cdot)$ be the j'th smallest singular value and eigenvalue, respectively, of their arguments. Then, for a Hamiltonian $\mathcal{H}$, $\beta \in \mathbb{R}$ and $\nu \in \mathbb{Z}^+$,

$$\left\{\mu_j\left[\exp\left(-\frac{\beta}{\nu}\mathcal{H}\right)\right]\right\}^\nu = \sigma_j(\exp[-\beta\mathcal{H}]. \tag{C7}$$

Proof. As $\exp(-\frac{\beta}{\nu}\mathcal{H})$ is positive-semi-definite, $\mu_j[\exp(-\frac{\beta}{\nu}\mathcal{H})] = \sigma_j[\exp(-\frac{\beta}{\nu}\mathcal{H})]$. Furthermore, as $\sigma_j[\exp(-\frac{\beta}{\nu}\mathcal{H})] = \exp(-\frac{\beta}{\nu}\sigma_j(\mathcal{H}))$, the result follows from raising both sides to the power ν. ■

Recall from Appendix A that $\mathcal{T}_{R,1}$ denotes one Trotter step for the imaginary time evolution. As this Appendix deals exclusively with real temperatures which correspond to imaginary time evolutions, we suppress the subscript R and denote $\mathcal{T}_1 \equiv \mathcal{T}_{R,1}$ for brevity. Note that the number of Trotter steps ν is implicit in $\mathcal{T}_1$ [see Eq. (2)]. We denote the operator error $\mathcal{W}_\nu$ in one Trotter step as

$$\exp\left(-\frac{\beta}{\nu}\mathcal{H}\right)\mathcal{W}_\nu = \mathcal{T}_1 \equiv \prod_{j=1}^{L}\exp\left(-\frac{\beta}{\nu}h_j\mathfrak{P}_j\right). \tag{C8}$$

Lemma 11. The multiplicative (relative) Trotter error ϵ_{mT} in Eq. (14) in approximating $Z = \mathrm{Tr}[\exp(-\beta\mathcal{H})]$ by $Z_{\mathrm{T}} = \mathrm{Tr}([\mathcal{T}_1]^\nu)$ is bounded by $\epsilon_{\mathrm{mT}} \geqslant ||\mathcal{W}_\nu||^\nu - 1$.

Proof. Using Lemma 10, the submultiplicity of the spectral norm and that the spectral norm is greater than or equal to all singular values

$$Z_{\mathrm{T}} = \mathrm{Tr}([\mathcal{T}_1]^\nu) = \mathrm{Tr}\left(\left[\exp\left(-\frac{\beta}{\nu}\mathcal{H}\right)\mathcal{W}_\nu\right]^\nu\right)$$
$$\leqslant \sum_{j=1}^{2^N}\sigma_j(\exp(-\beta\mathcal{H}))||\mathcal{W}_\nu||^\nu = Z||\mathcal{W}_\nu||^\nu, \tag{C9}$$

where the Hamiltonian $\mathcal{H}$ acts on N qubits. Thus, $|Z - Z_{\mathrm{T}}| \leqslant (||\mathcal{W}_\nu||^\nu - 1)Z$. The result follows by setting $\epsilon_{\mathrm{mT}} \geqslant ||\mathcal{W}_\nu||^\nu - 1$ in Eq. (14). ■

3. Proof of Theorem 2

The following proof follows the strategy of Ref. [32] to exploit the noncommutativity among the terms of Hamiltonian. However, for simplicity, it is specialized to the first-order Trotterization. It uses lemma 8 to set up the use of lemma 9. This gives us an expression of an operator essentially applying the error $\mathcal{W}(\beta)$. This then leads to a bound of the spectral norm of the error operator in a single Trotter step $\mathcal{W}(\beta/\nu) = \mathcal{W}_\nu$.

Proof. Denote $A_j \equiv h_j\mathfrak{P}_j$, and

$$\Theta(\beta) = \prod_{j=1}^{L}(e^{-\beta A_j}). \tag{C10}$$

Note that $\Theta(\beta/\nu) = \mathcal{T}_1 = \prod_{j=1}^{L}\exp(-\frac{\beta}{\nu}h_j\mathfrak{P}_j)$. By differentiation

$$\frac{d\Theta(\beta)}{d\beta} = -A_1\Theta(\beta) - \sum_{k=2}^{L}\left[\prod_{j=1}^{k-1}(e^{-\beta A_j})\cdot A_k\cdot\prod_{j=k}^{L}(e^{-\beta A_j})\right] \tag{C11}$$

$$= \left\{-A_1 - \sum_{k=2}^{L}\left[\prod_{j=1}^{k-1}(e^{-\beta A_j})\cdot A_k\cdot\prod_{j=k}^{L}(e^{-\beta A_j})\right]\prod_{j=L}^{1}(e^{\beta A_j})\right\} \times \Theta(\beta) \tag{C12}$$

$$= \left\{-A_1 - \sum_{k=2}^{L}\left[\prod_{j=1}^{k-1}(e^{-\beta A_j})\cdot A_k\cdot\prod_{j=k-1}^{1}(e^{\beta A_j})\right]\right\}\Theta(\beta), \tag{C13}$$

where we used lemma 7 in the last step. Denoting

$$\mathcal{F}(\beta) \equiv -A_1 - \sum_{k=2}^{L}\mathcal{S}_k(\beta) \text{ with}$$

$$\mathcal{S}_k(\beta) = \prod_{j=1}^{k-1}(e^{-\beta A_j})\cdot A_k\cdot\prod_{j=k-1}^{1}(e^{\beta A_j}), \tag{C14}$$

whereby

$$\frac{d\Theta(\beta)}{d\beta} = \mathcal{F}(\beta)\Theta(\beta). \tag{C15}$$

Lemma 8 gives

$$\Theta(\beta) = \exp\left\{\int_0^{\beta}\left(\frac{d\mathcal{T}_1(\tau)}{d\tau}\mathcal{T}_1^{-1}(\tau)\right)d\tau\right\}$$

$$= \exp\left\{\int_0^{\beta}\mathcal{F}(\tau)\,d\tau\right\}$$

$$= \exp\left\{\int_0^{\beta}(-\mathcal{H} + [\mathcal{F}(\tau) + \mathcal{H}])\,d\tau\right\}. \tag{C16}$$

Lemma 9 gives

$$\Theta(\beta) = \exp\left\{\int_0^{\beta}(-\mathcal{H})\,d\tau\right\}$$

$$\times \exp\left\{\int_0^{\beta}\left(\exp\left\{\int_0^{\tau_1}\mathcal{H}\,d\tau_2\right\}[\mathcal{F}(\tau_1) + \mathcal{H}]\right.\right.$$

$$\left.\left.\times \exp\left\{\int_0^{\tau_1}(-\mathcal{H})\,d\tau_2\right\}\right)d\tau_1\right\} \tag{C17}$$

$$= \exp\{-\beta\mathcal{H})\exp\left\{\int_0^{\beta}\left(e^{\mathcal{H}\tau_1}[\mathcal{F}(\tau_1) + \mathcal{H}]e^{-\mathcal{H}\tau_1}\right)d\tau_1\right\} \tag{C18}$$

$$\equiv \exp\{-\beta\mathcal{H})\mathcal{W}(\beta), \tag{C19}$$

where $\mathcal{W}$, denotes the error operator when $\exp\{-\beta\mathcal{H})$ is approximately implemented via $\Theta(\beta)$. Then using Eq. (C14)

$$\mathcal{W}(\beta)$$

$$= \exp\left\{\int_0^{\beta}\left(e^{\mathcal{H}\tau_1}[\mathcal{F}(\tau_1) + \mathcal{H}]e^{-\mathcal{H}\tau_1}\right)d\tau_1\right\}$$

$$= \exp\left\{\int_0^{\beta}\left(e^{\mathcal{H}\tau_1}\mathcal{F}(\tau_1)e^{-\mathcal{H}\tau_1} + \mathcal{H}\right)d\tau_1\right\} \tag{C20}$$

$$= \exp\left\{\int_0^{\beta}\left(e^{\mathcal{H}\tau_1}\left[-A_1 - \sum_{k=2}^{L}\mathcal{S}_k(\tau_1)\right]e^{-\mathcal{H}\tau_1} + \mathcal{H}\right)d\tau_1\right\}. \tag{C21}$$

Let z be the greatest index in ζ_k less than k. Then

$$\mathcal{S}_k(\tau_1) = \prod_{j=1}^{k-1}(e^{-\tau_1 A_j})\cdot A_k\cdot\prod_{j=k-1}^{1}(e^{\tau_1 A_j})$$

$$= \prod_{j=1}^{z-1}(e^{-\tau_1 A_j})\cdot e^{-\tau_1 A_z}A_k e^{\tau_1 A_z}\cdot\prod_{j=z-1}^{1}(e^{\tau_1 A_j}) \tag{C22}$$

$$= \prod_{j=1}^{z-1}(e^{-\tau_1 A_j})\cdot A_k\cdot\prod_{j=z-1}^{1}(e^{\tau_1 A_j}) \tag{C23}$$

$$+ \prod_{j=1}^{z-1}(e^{-\tau_1 A_j})\cdot[A_z, A_k]\tau_1\cdot\prod_{j=z-1}^{1}(e^{\tau_1 A_j})$$

$$+ \prod_{j=1}^{z-1}(e^{-\tau_1 A_j})\cdot\int_0^{\tau_1}\int_0^{\tau_2}(e^{-\tau_3 A_z}[A_z, [A_z, A_k]]e^{\tau_3 A_z})$$

$$\times\, d\tau_3 d\tau_2\cdot\prod_{j=z-1}^{1}(e^{\tau_1 A_j}), \tag{C24}$$

where this last line follows from the identity

$$e^{tA}Be^{-tA} = B + [A, B]t + \int_0^{t}dt_2\int_0^{t_2}dt_3 e^{t_3 A}[A, [A, B]]e^{-t_3 A}. \tag{C25}$$

Note that the first term in Eq. (C23) is devoid of A_z. Repeating the above on the first term

$$\mathcal{S}_k(\tau_1) = A_k + \mathcal{D}_k(\tau_1), \tag{C26}$$

where

$$\mathcal{D}_k(\tau_1) = \sum_{\substack{z\leqslant k\\ z\in\zeta_k}}\left[\prod_{j=1}^{z-1}(e^{-\tau_1 A_j})[A_z, A_k]\tau_1\prod_{j=z-1}^{1}(e^{\tau_1 A_j})\right.$$

$$+ \prod_{j=1}^{z-1}(e^{-\tau_1 A_j})\int_0^{\tau_1}\int_0^{\tau_2}(e^{-\tau_3 A_z}[A_z, [A_z, A_k]]e^{\tau_3 A_z})$$

$$\left.\times\, d\tau_3 d\tau_2\prod_{j=z-1}^{1}(e^{\tau_1 A_j})\right]. \tag{C27}$$

Thus, the quantity in the square brackets in Eq. (C21) equals $\mathcal{H}+\sum_{k=2}^{L}\mathcal{D}_k(\tau_1)$, whereby

$$\mathcal{W}(\beta)=\exp\left\{-\int_0^\beta\left(e^{\mathcal{H}\tau_1}\left[\sum_{k=2}^{L}\mathcal{D}_k(\tau_1)\right]e^{-\mathcal{H}\tau_1}\right)d\tau_1\right\}. \tag{C28}$$

Thus,

$$||\mathcal{W}(\beta)||\leqslant\exp\left\{\int_0^\beta\left(e^{||\mathcal{H}||\tau_1}\left[\sum_{k=2}^{L}||\mathcal{D}_k(\tau_1)||\right]e^{||\mathcal{H}||\tau_1}\right)d\tau_1\right\}. \tag{C29}$$

As the commutator of Paulis are Paulis, and their spectral norm is unity, $||[A_z,A_k]||=2|h_z||h_k|$ and $||[A_z,[A_z,A_k]]||=4|h_z|^2|h_k|$. Denoting

$$\Omega_k=\sum_{j=1}^{k}||A_j||=\sum_{j=1}^{k}|h_j|,\quad\text{and}\quad\Omega\equiv\Omega_L, \tag{C30}$$

$$\begin{aligned}||\mathcal{D}_k(\tau_1)||&\leqslant 2|h_z||h_k|\sum_{\substack{z\leqslant k\\ z\in\zeta_k}}\Big[e^{2\tau_1\Omega_{z-1}}\tau_1+2|h_z|e^{2\tau_1\Omega_{z-1}}\\&\quad\times\int_0^{\tau_1}\int_0^{\tau_2}e^{2|h_z|\tau_3}d\tau_3d\tau_2\Big]\\&=|h_k|\sum_{\substack{z\leqslant k\\ z\in\zeta_k}}[e^{2\tau_1\Omega_z}-e^{2\tau_1\Omega_{z-1}}].\end{aligned} \tag{C31}$$

Denoting $\omega_z=2(\Omega+\Omega_z)$,

$$||\mathcal{W}(\beta)||\leqslant\exp\left\{\sum_{k=2}^{L}|h_k|\sum_{\substack{z\leqslant k\\ z\in\zeta_k}}\left[\frac{e^{\beta\omega_z}-1}{\omega_z}-\frac{e^{\beta\omega_{z-1}}-1}{\omega_{z-1}}\right]\right\}. \tag{C32}$$

Setting $w=2\beta\Omega/\nu$, the spectral norm of the Trotter error in ν Trotter steps of size β/ν each is

$$||\mathcal{W}_\nu||^\nu\leqslant\exp\left\{\beta\mathfrak{h}\left[\frac{e^{2w}-1}{2w}-\frac{e^{w}-1}{w}\right]\right\}\approx\exp\left\{\beta\mathfrak{h}\frac{w}{2}\right\}, \tag{C33}$$

where $\mathfrak{h}=\sum_{k=2}^{L}|h_k|N_k$, with $N_k=|\{z\leqslant k \text{ such that } z\in\zeta_k\}|$ accounting for the mutual noncommutativity amongst the terms of $\mathcal{H}$, The last approximation neglects higher-order terms in w because for a fixed Hamiltonian (Ω) and temperature (β), $w\to 0$ as $\nu\to\infty$. Finally, using lemma 11,

$$\nu\geqslant\frac{\beta^2\Omega\mathfrak{h}}{\ln(1+\epsilon_{\mathrm{mT}})}. \tag{C34}$$

■

[1] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Chem. Rev. **120**, 12685 (2020).

[2] I. Y. Dodin and E. A. Startsev, Phys. Plasmas **28**, 092101 (2021).

[3] J. P. F. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K.-L. Chan, C.-M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jiménez-Hoyos, E. Kozik, X.-W. Liu, A. J. Millis, N. V. Prokof'ev, M. Qin, G. E. Scuseria, H. Shi, B. V. Svistunov, L. F. Tocchio, I. S. Tupitsyn *et al.* (Simons Collaboration on the Many-Electron Problem), Phys. Rev. X **5**, 041041 (2015).

[4] M. Motta, D. M. Ceperley, G. K.-L. Chan, J. A. Gomez, E. Gull, S. Guo, C. A. Jiménez-Hoyos, T. N. Lan, J. Li, F. Ma, A. J. Millis, N. V. Prokof'ev, U. Ray, G. E. Scuseria, S. Sorella, E. M. Stoudenmire, Q. Sun, I. S. Tupitsyn, S. R. White, D. Zgid *et al.*, Phys. Rev. X **7**, 031059 (2017).

[5] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh, Mathematical Proceedings of the Cambridge Philosophical Society **108**, 35 (1990).

[6] D. Welsh, in *Disorder in Physical Systems*, edited by G. G. Dominic Welsh (Oxford University Press, Oxford, 1990), pp. 307–321.

[7] D. A. Lidar, New J. Phys. **6**, 167 (2004).

[8] L. A. Goldberg and H. Guo, Comput. Complex. **26**, 765 (2017).

[9] G. De las Cuevas, W. Dür, M. Van den Nest, and M. A. Martin-Delgado, New J. Phys. **13**, 093021 (2011).

[10] A. Matsuo, K. Fujii, and N. Imoto, Phys. Rev. A **90**, 022304 (2014).

[11] E. Bernstein and U. Vazirani, SIAM J. Comput. **26**, 1411 (1997).

[12] S. Aaronson, BQP and the polynomial hierarchy, in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing, STOC '10* (Association for Computing Machinery, New York, 2010), pp. 141–150.

[13] A. N. Chowdhury, R. D. Somma, and Y. Subasi, Phys. Rev. A **103**, 032422 (2021).

[14] E. Knill and R. Laflamme, Phys. Rev. Lett. **81**, 5672 (1998).

[15] D. Shepherd, Computation with unitaries and one pure qubit, arXiv:quant-ph/0608132.

[16] P. W. Shor and S. P. Jordan, Quantum Info. Comput. **8**, 681 (2008).

[17] D. Aharonov, V. Jones, and Z. Landau, Algorithmica **55**, 395 (2009).

[18] A. Datta and A. Shaji, Int. J. Quantum. Inform. **09**, 1787 (2011).

[19] K. Fujii, H. Kobayashi, T. Morimae, H. Nishimura, S. Tamate, and S. Tani, Phys. Rev. Lett. **120**, 200502 (2018).

[20] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum **4**, 307 (2020).

[21] E. Tang, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019* (Association for Computing Machinery, New York, 2019), pp. 217–228.

[22] E. Tang, Phys. Rev. Lett. **127**, 060503 (2021).

[23] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019).

[24] G. Pan and Z. Y. Meng, Sign problem in quantum Monte Carlo simulation, arXiv:2204.08777.

[25] S. Lloyd, Science **273**, 1073 (1996).

[26] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, Phys. Rev. A **92**, 062318 (2015).

[27] M. Heyl, P. Hauke, and P. Zoller, Sci. Adv. **5**, eaau8342 (2019).

[28] M. Benedetti, M. Fiorentini, and M. Lubasch, Phys. Rev. Res. **3**, 033083 (2021).

[29] G. C. Knee and W. J. Munro, Phys. Rev. A **91**, 052327 (2015).

[30] D. Gottesman, The Heisenberg Representation of Quantum Computers, in *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, Hobart, July 13–17, 1998* (International, Boston, 1998).

[31] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004).

[32] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Phys. Rev. X **11**, 011020 (2021).

[33] C. Derby, J. Klassen, J. Bausch, and T. Cubitt, Phys. Rev. B **104**, 035118 (2021).

[34] Algorithms and Libraries for Physics Simulations, Version 2.3. Available at http://alps.comp-phys.org/mediawiki/index.php/Main_Page.

[35] I. Hen, Phys. Rev. Res. **3**, 023080 (2021).

[36] C. Berger, L. Rammelmüller, A. Loheac, F. Ehmann, J. Braun, and J. Drut, Phys. Rep. **892**, 1 (2021).

[37] L. Clinton, J. Bausch, and T. Cubitt, Nat. Commun. **12**, 4989 (2021).

[38] Y.-X. Liu, J. Hines, Z. Li, A. Ajoy, and P. Cappellaro, Phys. Rev. A **102**, 010601(R) (2020).