Supplementary Notes for: Hunting for quantum-classical crossover in condensed matter problems

Nobuyuki Yoshioka,[1, 2, 3, *] Tsuyoshi Okubo,[4, 3, †] Yasunari Suzuki,[5, 3, ‡] Yuki Koizumi,[5] and Wataru Mizukami[6, 7, 3, §]

[1] *Department of Applied Physics, University of Tokyo,*
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[2] *Theoretical Quantum Physics Laboratory, RIKEN Cluster for Pioneering Research (CPR), Wako-shi, Saitama 351-0198, Japan*
[3] *JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan*
[4] *Institute for Physics of Intelligence, University of Tokyo,*
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[5] *NTT Computer and Data Science Laboratories, Musashino 180-8585, Japan*
[6] *Center for Quantum Information and Quantum Biology,*
Osaka University, 1-2 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan
[7] *Graduate School of Engineering Science, Osaka University,*
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan.

In this Supplemental Notes, we present the details on the classical and quantum resource estimation on ground state problem in condensed matter physics. In Sec. I, we provide the definitions and brief introduction to the nature of the representative lattice models considered in condensed matter physics. In Sec. II, we describe how to estimate the classical runtime of tensor-network algorithms, namely the Density-Matrix Remormalization-Group (DMRG) and the Projected Entangled-Pair State (PEPS) algorithms. In Sec. III, we introduce the adiabatic time evolution as the state preparation method and analyze its circuit complexity for 2d J_1-J_2 Heisenberg model. In Sec. IV, we provide a overview on the structure and cost of quantum algorithm to estimate ground state energy, namely the quantum phase estimation algorithm. In Sec. V, we describe the post-Trotter Hamiltonian simulation algorithms that are exponentially more efficient than the Trotter-based ones in terms of error scaling. In Sec. VI, we summarize the gate complexity, or more concretely the T-count, to execute basic operations in fault-tolerant quantum computing. In Sec. VII, we describe how to modify an oracle in the qubitization algorithm such that the boundary conditions in the Hamiltonian does not heavily affect the gate complexity. In Sec. VIII, we provide a systematic framework to estimate the runtime of the qubitization-based quantum phase estimation algorithm assuming Clifford+T formalism under surface code. Finally, in Sec. IX, we put all the analysis of quantum algorithm together and present the estimation on total quantum resource including the runtime, physical qubits, and code distance of the surface code.

CONTENTS

I. DEFINITION OF TARGET MODELS

In this paper, we discuss the quantum-classical crossover in condensed matter problems where the Hamiltonian is defined on d-dimensional translationally symmetric lattice with at most G-local terms:

$$H = \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} w_{\boldsymbol{\mu},\boldsymbol{\alpha}} \hat{\Lambda}_{\boldsymbol{p}}^{(\alpha_1)} \hat{\Lambda}_{\boldsymbol{p}+\boldsymbol{\mu}_1}^{(\alpha_2)} \cdots \hat{\Lambda}_{\boldsymbol{p}+\boldsymbol{\mu}_{G-1}}^{(\alpha_G)}, \tag{1}$$

where $\boldsymbol{p} = (p_1, \dots, p_d)$ labels the lattice site (including the sublattice structure), $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{G-1})$ is a set of vectors that identifies the connection between interacting sites, $\boldsymbol{\alpha}$ discriminates the interaction on the sites, $\hat{\Lambda}_{\boldsymbol{p}}^{(\alpha)}$ is an operator for microscopic degrees of freedom (e.g. spin, fermion, boson) on site $\boldsymbol{p}$, and $w_{\boldsymbol{\mu},\boldsymbol{\alpha}}$ is the amplitude of the interaction. In particular, we employ three representative models with long-standing problems; 2d J_1-J_2 Heisenberg model, 2d Fermi-Hubbard model, and spin-1 Heisenberg chain. In the following, we provide the definition of the target models accompanied with brief descriptions.

A. Spin-1/2 2d J_1-J_2 Heisenberg model

The physical nature of the paradigmatic J_1-J_2 Heisenberg model on the square lattice is under debate for over decades, especially regarding the property of the ground state [1–5]. The frustration, i.e., the competition between interactions that precludes satisfying all the energetic gain, is considered to suppress the formation of long-range order, and thus it amplifies the nontrivial effect of quantum fluctuation. Together with geometrically frustrated magnets defined on e.g., triangular, Kagomé, and pyrochlore lattices, the J_1-J_2 Heisenberg model on square lattice has been investigated intensively by classical algorithms. Considering that the model is commonly recognized as one of the most challenging and physically intriguing quantum spin models, it is genuinely impactful to elucidate the quantum-classical crossover if exists at all.

The Hamiltonian of spin-1/2 J_1-J_2 Heisenberg model on the square lattice is defined as

$$H = J_1 \sum_{\langle p,q \rangle} \sum_{\alpha \in \{X,Y,Z\}} S_p^\alpha S_q^\alpha + J_2 \sum_{\langle\langle p,q \rangle\rangle} \sum_{\alpha \in \{X,Y,Z\}} S_p^\alpha S_q^\alpha, \tag{2}$$

where the summation of the first and second terms concern pairs of nearest-neighbor and next-nearest-neighbor sites, respectively, and S_p^α is the spin-1/2 operator on the p-th site that is related with the α-th component of the Pauli matrices as $S_p^\alpha = \frac{1}{2}\sigma_p^\alpha$.

B. 2d Fermi-Hubbard model

As the second representative model, we choose the Fermi-Hubbard model on the square lattice. The emergent phenomena in Fermi-Hubbard model is truly abundant, and provides a significant insights into electronic and magnetic properties of materials; the unconventional superfluidity or superconductivity, Mott insulating behaviour, and quantum magnetism. However, the intricate nature of Fermi statistics has hindered us from performing scalable and quantitative description in the truly intriguing zero-temperature regime.

The Hamiltonian of Fermi-Hubbard model on the square lattice reads

$$H = -t \sum_{\langle p,q\rangle,\sigma} (c_{p,\sigma}^\dagger c_{q,\sigma} + \text{h.c.}) + U \sum_p n_{p,\uparrow} n_{p,\downarrow}, \tag{3}$$

where $c_{p,\sigma}^{(\dagger)}$ is the fermionic annihilation (creation) operator on site p with spin $\sigma \in \{\uparrow,\downarrow\}$, $n_{p,\sigma} = c_{p,\sigma}^\dagger c_{p,\sigma}$ is the corresponding number operator, t is the hopping amplitude, and U is the repulsive onsite potential.

C. Spin-1 Heisenberg chain

The gap structure of translation and $U(1)$-invariant antiferromagnetic spin-S chain has been argued to differ according to whether S is half-integer or integer. Such a statement has been known as the Haldane conjecture [6, 7], which has attracted interest of condensed matter physicists for over decades. While the ground states for half-integer S cases are considered to be gapless and hence there is no supporting argument to be simulated classically by polynomial time, it is reasonable to expect for integer S cases to be computed efficiently using, e.g., the matrix product state (MPS), since the entanglement entropy shall obey area law if the system is gapped as is predicted in Haldane conjecture [8]. In this context, it is not clear if there is any quantum advantage at all for simulating the ground state or the gap structure in integer-S Heisenberg chain.

In this paper, we consider the simplest gapped case in which quantum advantage is unlikely; the spin-1 antiferromagnetic Heisenberg chain. The Hamiltonian reads

$$H = \sum_p \sum_{\alpha\in\{X,Y,Z\}} S_p^\alpha S_{p+1}^\alpha, \tag{4}$$

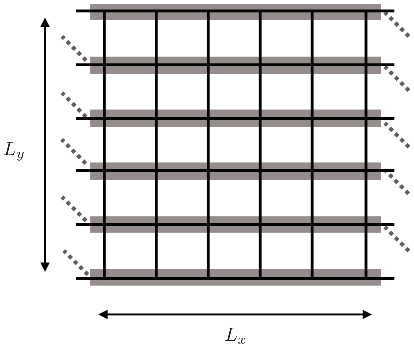
where S_p^α is the spin-1 operator acting on the p-th site.

II. CLASSICAL SIMULATION OF GROUND STATE USING DENSITY-MATRIX RENORMALIZATION GROUP

A. Overview of DMRG and PEPS algorithms

The Density-Matrix Renormalization Group (DMRG) has been established as one of the most powerful numerical tools to investigate the strongly correlated one-dimensional (1d) quantum lattice models [9]. DMRG can be understood as a variational calculation based on the Matrix Product State (MPS) [10, 11]. Thanks to the area law of the entanglement entropy expected to hold for low-energy states of the 1d quantum lattice models [12], MPS is a quantitatively accurate variational ansatz for ground states of many quantum many-body systems [13]. Although MPS cannot cover the area law scaling for two or higher dimensional systems, DMRG is still applicable for quasi 1d cylinders or finite-size 2d systems. Indeed, we can find recent applications of DMRG to the 2d models treated in this study [14, 15].

We can also consider a 2d tensor network state, called Projected Entangled-Pair State (PEPS) or Tensor Product State (TPS), to calculate 2d lattice models [16, 17]. Hereafter, we simply refer to such a state as PEPS. Although it might be more suitable than DMRG for simulating 2d systems because PEPS can cover 2d area law of the entanglement entropy, it has been widely used for direct simulations of infinite 2d systems. Furthermore, as we see later, for a small 2d systems treated in this study, the DMRG runs more efficiently.



Supplementary Figure 1. $L_x \times L_y$ square lattice used in the DMRG calculations. We consider the periodic boundary conditions along y direction and the open boundary condition along x direction. The gray lines indicate the assignment of MPS to the lattice sites.

One problem with applying DMRG to 2d systems is that its computation cost may increase exponentially as we increase the system size $N_{\rm site}$ while keeping the same accuracy in the ground state energy. Because the computation cost of PEPS is expected to increase polynomially as we increase $N_{\rm site}$, it can be more efficient than DMRG for larger $N_{\rm site}$. However, as we see in the main text, the crossover between classical and quantum computations seems to occur at the relatively small systems for the models we treated. For such small systems, the computation cost of the DMRG is indeed smaller than that of PEPS. Thus, we believe DMRG gives proper estimations of computational resources to simulate 2d systems in classical computers.

The computation time to obtain the ground state energy within a given target accuracy can depend on both the bond-dimension D and the optimization steps. Although the error in a ground state energy calculated by DMRG or variational PEPS is expected to decrease exponentially as a function of D asymptotically, its general relationship depends on the details of the target models. Similarly, it is not easy to estimate the necessary optimization steps *a priori*. Thus, to estimate the problem sizes where quantum computation will be superior to DMRG, we need to perform actual classical computations. In DMRG simulation, the computation time for the one optimization step, usually called sweep, scales as $O(N_{\rm site}D^3)$, where $N_{\rm site}$ is the number of lattice sites.

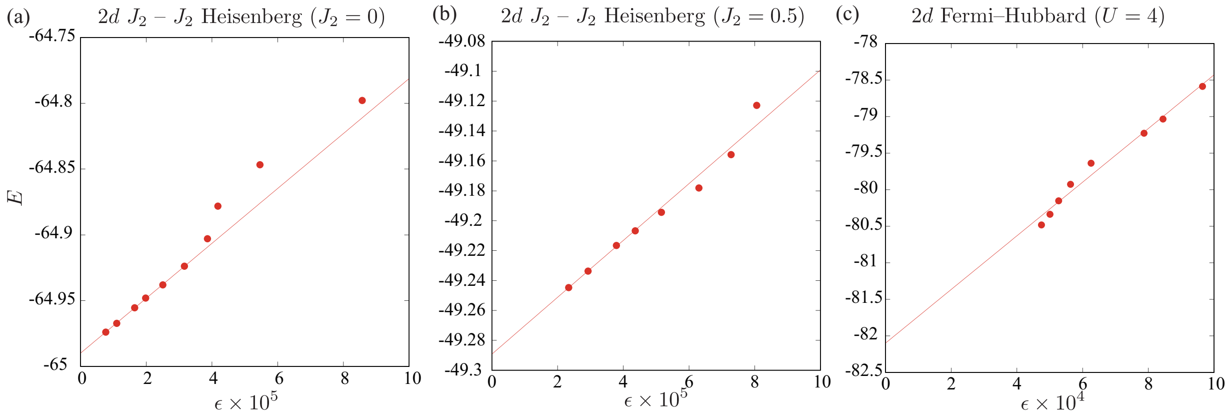
In the case of PEPS, two types of optimization algorithms have been widely used: imaginary time evolution [17–19] and the variational optimization similar to DMRG algorithm [20–22]. Usually, the yields more accurate simulation of ground states, although the environment tensor networks and contraction of them in the variational optimization becomes more complicated than the case of 1d tensor network. The recent development of the technique based on the automatic differentiation [22] provides us with a more stable implementation of the variational optimization for PEPS. Thus, to estimate the computation time for PEPS, here we employ the variation optimization with the automatic differentiation. In this case, the computation time for the one optimization step scale as $O(N_{\rm site}^{\alpha}D^{\beta})$, where $\alpha = 1 \sim 1.5$ and $\beta = 10 \sim 12$ depending on the approximate contraction algorithm.

B. Error analysis and computation time estimate for two-dimensional models by DMRG

In this subsection, we explain our procedure to estimate the computation time for the ground state energy estimation in 2d systems. For both the J_1-J_2 Heisenberg and the Hubbard models, we consider $L_x \times L_y$ square lattice with the periodic boundary condition along x direction and the open boundary condition at y direction so that the system forms a cylinder. In the DMRG calculation, the MPS string wraps around the cylinder as shown in Supplementary Fig. 1.

To estimate the computation times in the ground state calculation by a classical computer, we performed DMRG simulation using ITensor [23]. Typically, we set the maximum bond dimension as $D = D_{\rm max}$, and optimized the state starting from a random MPS with $D = 10$. We used $U(1)$ symmetric DMRG for the Heisenberg models, and similarly, we used particle number conservation for the Hubbard model. Most of the calculations have been done by a single CPU, AMD EPYC 7702, 2.0GHz, in the ISSP supercomputer center at the University of Tokyo. We performed multithreading parallelization using four cores.

The estimation of computational time to reach desired total energy accuracy ϵ is performed by the following three steps. Firstly, we optimize quantum states with various $D_{\rm max}$'s, and estimate the ground state energy by extrapolation. Then, we analyze the optimization dynamics of the energy errors. As we see in the main text, the dynamics of different $D_{\rm max}$ almost collapse into a single curve. Finally, we estimate the elapsed time to obtain a quantum state with the



Supplementary Figure 2. The extrapolations of the ground state energies for 10×10 square lattice models. Lines are the results of the linear fitting by Eq. (5). (a) J_1-J_2 Heisenberg model with $J_2 = 0$. (b) J_1-J_2 Heisenberg model with $J_2/J_1 = 0.5$. (c) The Hubbard model with $U/t = 4$.

desired accuracy by extrapolating this scaling curve. In the following, we elaborate on each step one by one.

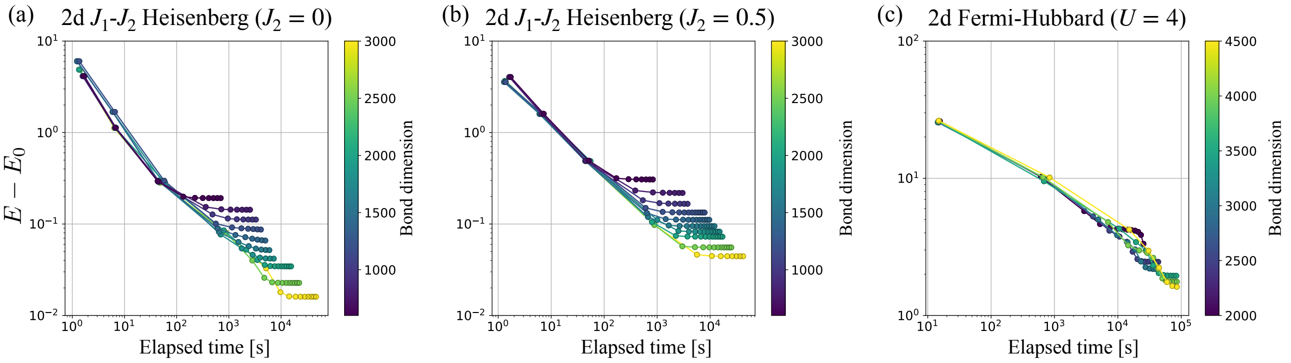
As described above, firstly, we estimate the ground state energy by varying D_{max}. It is known that the energy difference between the optimized and the exact energies is proportional to the so-called *truncation error*, which is a quantity obtained during the DMRG simulation. Namely, by iteratively updating the low-rank approximation by tensors for the MPS representation, one sums up the neglected singular values to compute the truncation error δ. For sufficiently small truncation errors, the energy difference is empirically known to behave as

$$E_{\text{DMRG}}(D_{\text{max}}) - E_{\text{exact}} \propto \delta(D_{\text{max}}), \tag{5}$$

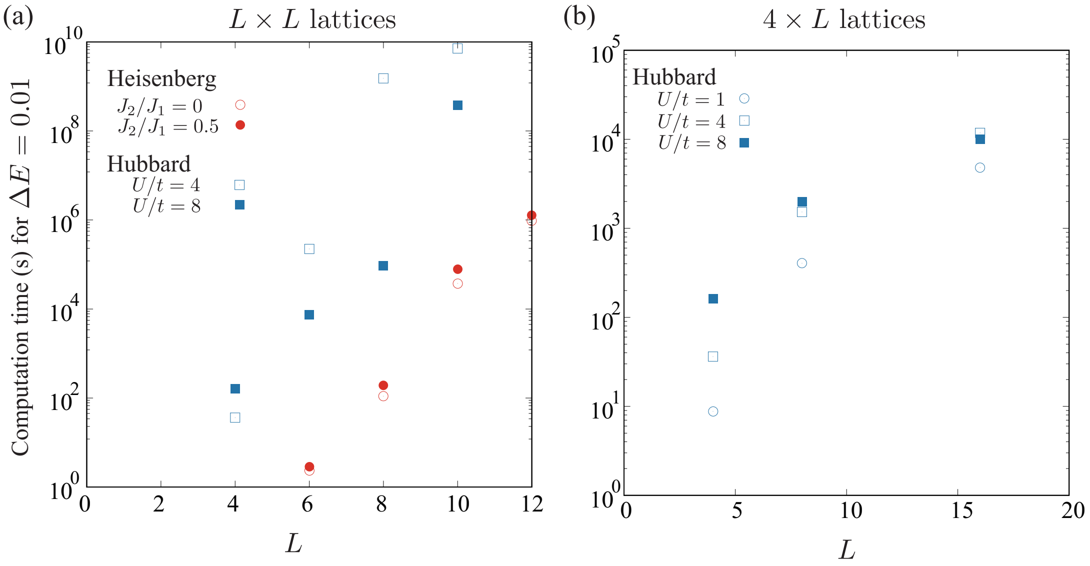
where $\delta(D_{\text{max}})$ is the truncation error for the optimized MPS state with D_{max} [24–26]. We extrapolated the obtained energies of target models to the zero truncation error using Eq. (5). Supplementary Fig. 2 shows typical extrapolation procedures for the J_1-J_2 Heisenberg model and the Fermi-Hubbard model on 10×10 square lattice. When the truncation errors are sufficiently small, we observe the expected linear behaviors as in Supplementary Figs. 2(a) and (b). However, in the case of the Hubbard model (Supplementary Fig. 2(c)), the fitting by the linear function is not so precise. This is due to the limitation of the maximum bond dimension treated in our study. We also observed similar behavior in larger lattices in the J_1-J_2 model. Although the extrapolations seem to be not so accurate in such parameters, it is still sufficient to estimate the order of computation time, which is relevant in comparison with quantum computations.

Using the estimated value of ground state energy E_0, next we focus on the dynamics of the energy errors for each calculation. In Supplementary Fig. 3, we show typical optimization dynamics as functions of the elapsed time. In the case of J_1-J_2 Heisenberg models with $J_2/J_1 = 0, 0.5$, we see that curves corresponding to various D_{max} almost collapse to a universal optimization curve. Because the universal curve is likely a power function, we probably extrapolate optimization dynamics beyond the maximum value of D_{max} treated in the calculations properly. In the case of the Hubbard model with $U/t = 4$, we did not see a perfect collapse into a universal curve. In addition, the optimization curve seems to be deviated from the power law. Similar behaviors are also observed in other combinations of L and U/t for the Hubbard model. Although the extrapolation to larger D_{max} becomes inaccurate for such cases, to estimate the order of the computation time, we performed a power-law fitting of $E - E_0$ for the largest D_{max} before the saturation.

Based on such power-law fittings of the maximum D_{max} dynamics, finally, we estimate the computation time to obtain the ground state with the total energy error $\epsilon := E - E_{\text{exact}} = 0.01$, where the unit of the energy is J_1 for the Heisenberg model and t for the Hubbard model. Since ϵ itself cannot be obtained in the simulation, we define the deviation from the extrapolated ground state energy as $\Delta E := E - E_0$, and assume $\Delta E \sim \epsilon$ such that ΔE is sufficiently accurate to predict the total runtime. Supplementary Fig. 4 shows the estimated computation times for obtaining $\Delta E = 0.01$ in classical DMRG simulations. The parameters for the DMRG calculations, together with the estimated computation times, are summarized in Supplementary Tables I and II. In the case of $L \times L$ geometry (Supplementary Fig. 4(a)), we see an expected exponential increase of the elapsed time as we increase L. We find that the nearest neighbor Heisenberg model ($J_2/J_1 = 0$) needs a slightly shorter elapsed time than that of the frustrated model ($J_2/J_1 = 0.5$). Although this is related to the difficulty of the problem, the difference is not so large when we consider the orders of the elapsed times. We observe a rather larger difference between the Heisenberg and the Hubbard models, which is mainly attributed to the difference in the number of local degrees of freedom. In particular,



Supplementary Figure 3. Dynamics of energy errors for (a) J_1-J_2 Heisenberg model with $J_2 = 0$. (b) J_1-J_2 Heisenberg model with $J_2/J_1 = 0.5$. (c) The Hubbard model with $U/t = 4$ in 10×10 square lattice models with various $D_{\max}$.



Supplementary Figure 4. Estimated computation time to obtain the ground states with the total energy error $\Delta E \equiv E - E_{\mathrm{exact}} = 0.01$. (a) The estimated computation time for the J_1-J_2 Heisenberg and Hubbard models for $L \times L$ square lattices. (b) The estimated computation time for the Hubbard models on $4 \times L$ quasi-1d square lattices.

the simulation in the Hubbard model is more time-consuming at $U/t = 4.0$ than that of the $U/t = 8.0$. This is probably explained by the larger energy gap in $U/t = 8.0$, which indicates lower entanglement in the ground state.

It may be informative to mention that the runtime under the quasi-1d geometry $(4 \times L)$ in the Hubbard model does not increase exponentially (See Supplementary Fig. 4(b)). Indeed, in this geometry, we expect 1d area law of the entanglement entropy even when we increase L, and therefore, the required bond dimension D to achieve a given target accuracy in the energy density becomes almost independent of L. Thus, we expect that the elapsed time to obtain $\epsilon = 0.01$ in the total energy increases in power law as a function of L.

In the above estimations, we used a single CPU. In practical calculations, we may expect a speed-up by GPU or parallelization using many CPUs. Indeed, in our simulation based on ITensor, we observed tens times faster execution by GPU when we did not impose the symmetries. Although ITensor simulation with the symmetry did not run with GPU, unfortunately, this observation indicates the estimated elapsed time can be shorter with a factor of 10^{-1}. In the case of multi-node parallelization, we may consider the real-space decomposition of the DMRG algorithm [27]. In this approach, each CPU optimizes different regions of the lattice (MPS), and communicates with each other to exchange information of the tensors. Because this approach needs to keep a sufficiently large block (set of tensors) for each CPU to obtain meaningful speed up, in the present system sizes, we divide the system into dozens at most. Thus, we estimate that the multi-node parallelization may give us a shorter elapsed time with a factor of 10^{-1}.

We remark that here, we estimated the computation time to actually obtain a quantum state whose energy error is within $\epsilon = 0.01$ so that other observables can be extracted consistently. Meanwhile, the runtime becomes much

Supplementary Table I. Parameters for estimating the computation time in DMRG simulations of J_1-J_2 Heisenberg model.

J_2/J_1	L_x	L_y	largest $D_{\max}$	lowest energy	E_0	Estimated computation time (s) for $\Delta E = 0.01$
0	6	6	800	-23.090068	-23.090117	2.3
0	8	8	3000	-41.348977	-41.349310	1.1×10^2
0	10	10	3000	-64.973863	-64.989896	3.7×10^4
0	12	12	2500	-93.847557	-93.951521	9.6×10^5
0.5	6	6	2000	-17.844007	-17.844015	2.9
0.5	8	8	1200	-31.538017	-31.544878	1.9×10^2
0.5	10	10	3000	-49.244805	-49.289216	7.9×10^4
0.5	12	12	2500	-70.699958	-70.782728	1.3×10^6

Supplementary Table II. Parameters for estimating the computation time in DMRG simulations of Hubbard model.

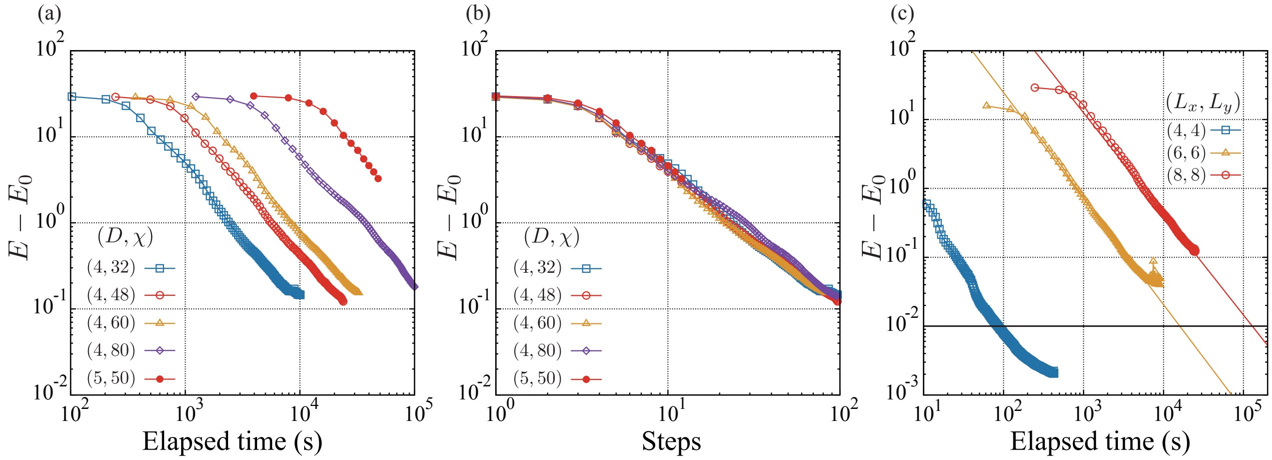
U/t	L_x	L_y	largest $D_{\max}$	lowest energy	E_0	Estimated computation time (s) for $\Delta E = 0.01$
1.0	4	4	2500	-21.175239	-21.175265	8.8
1.0	4	8	3500	-43.486042	-43.487663	4.1×10^2
1.0	4	16	3000	-88.230867	-88.240000	4.8×10^3
4.0	4	4	2500	-12.801348	-12.801536	3.6×10
4.0	6	6	3500	-29.081849	-29.302504	2.2×10^5
4.0	4	8	3500	-26.674309	-26.676826	1.5×10^3
4.0	8	8	5000	-52.008298	-52.802798	1.5×10^9
4.0	10	10	4500	-80.482124	-82.098947	7.1×10^9
4.0	4	16	3500	-54.495584	-54.509963	1.2×10^4
8.0	4	4	2500	-7.660925	-7.660942	1.6×10^2
8.0	6	6	3500	-17.737197	-17.767700	7.4×10^3
8.0	4	8	3500	-16.079582	-16.079696	2.0×10^3
8.0	8	8	5000	-31.806322	-31.993896	9.4×10^4
8.0	10	10	5000	-49.287802	-51.699085	3.7×10^8
8.0	4	16	3500	-32.922474	-32.923098	9.9×10^3

shorter if we merely focus on estimating the energy within the error $\epsilon = 0.01$ since the zero-truncation extrapolation within error ϵ itself can be done with smaller D. Indeed, we have observed that extrapolations allow us to reach smaller errors than the energies obtained from the actual DMRG simulations. This fact indicates that we might need shorter computation time to estimate the ground state energy within $\epsilon = 0.01$ in classical computers, although we point out that such a data analysis may also be available in quantum algorithms as well.

C. Error analysis and computation time estimate for two-dimensional models by PEPS

In this subsection, we explain our procedure to estimate the computation time for the ground state energy estimation in 2d J_1-J_2 Heisenberg model. Different from the DMRG calculations, we consider $L_x \times L_y$ square lattice with the open boundary condition in both x and y directions due to the restriction of PEPS contraction efficiency. We represent the ground state of such a system by a simple PEPS where we put a local tensor on each vertex of the square lattice.

To estimate the computation time in the ground state calculation by PEPS, we performed variational optimization using QUIMB [28]. Initially, we set random tensors with the bond dimension D, which is fixed during the entire optimization process. In order to compute expectation values and their derivatives to perform optimization, one must aim to compute the expectation values in an approximate sense, since the exact contraction of tensors in PEPS is a #P-hard problem [29]. Namely, one must introduce environment tensors with bond dimension χ such that we can avoid superexponential computational cost but still maintain simulation accuracy. In this regard, we emphasize that simulation by PEPS is strictly different from DMRG; the calculation of the expectation value for a given PEPS contains errors that can be systematically controlled by using larger χ. In our work, we have searched the appropriate value of $\chi \geq D^2$ so that the approximation error is sufficiently suppressed. Most of the calculations have been done



Supplementary Figure 5. (a) Optimization curves of energy for J_1-J_2 Heisenberg model with $J_2 = 0.5$ on $(L_x, L_y) = (8, 8)$ square lattice with various D and χ. (b) The energy error as a function of steps. (c) Typical optimization curves for $(L_x, L_y) = (4, 4)$, $(6, 6)$, and $(8, 8)$ with $D = 4$ and $\chi = 48$.

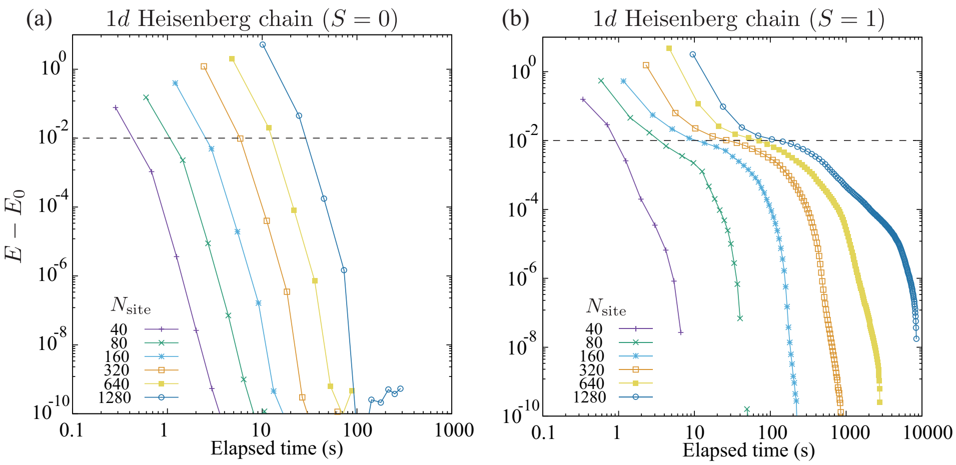
by a single CPU, Intel Xeon Platinum 8280 2.7GHz (28core), in the ISSP supercomputer center at the University of Tokyo. We performed multithreading parallelization using 28 cores.

Similar to the DMRG calculations, the estimation of the computation time to reach desired total energy accuracy ϵ is performed by the following steps. Firstly, we calculate the exact ground state energy for the open boundary system by DMRG with the extrapolation used in the previous subsection. Then, we analyze the elapsed time scaling of energy errors for various D's and χ's. As we will see later, the optimization curves of various D's and χ's do not collapse into a single curve when we plot them as a function of the elapsed time. Thus, in the case of PEPS, we extrapolate the curve of $D = 4$ to $\Delta E = \epsilon$, and estimate the approximate lower bound of the computation time. Considering that the bond dimension is expected to scale as $D = O(\log N)$ in the asymptotic limit in our target models, we envision that the runtime scaling with N becomes even worse than what we have observed in our work.

In Supplementary Fig. 5(a) we show the optimization curves of energy for J_1-J_2 Heisenberg model with $J_2 = 0.5$ on $(L_x, L_y) = (8, 8)$ square lattice with various D and χ. Different from the case of the DMRG simulation, the optimization curves of PEPS method do not show a collapse among different D and χ. One of the reasons for the difference is that we fix the bond dimensions D during the entire optimization process; wed do not dynamically increase during the simulation in the case of the PEPS method. In contrast, in DMRG simulation, we always start from a fixed value D_0 and dynamically increase D according to a common scheduling until we reach $D_{\max}$. Thus, for sufficiently large $D_{\max}$, the bond dimension in the transient region is almost independent on $D_{\max}$ in DMRG method. Although we observe that the optimization curves in PEPS are dependent on D and χ, we still see a data collapse into a single curve when we plot the curves as a function of optimization steps, instead of the actual elapsed time, as shown in Supplementary Fig. 5(b). The obtained curve seems to be well described by a power-law scaling, which is in common with the case of DMRG method.

This almost bond-dimension independent power-law decay indicates that we can extrapolate the dynamics of energy errors toward our target $\Delta E = 0.01$ even if the actual simulation could not achieve this accuracy. Although we do not know the proper set of bond dimensions D and χ to obtain $\Delta E = 0.01$, we can estimate the lower bound of the computation time by using smaller D and χ. Here we commonly use $D = 4$ and $\chi = 48$ for estimating such lower bounds for $(L_x, L_y) = (4, 4)$, $(6, 6)$, and $(8, 8)$. (Note that we clearly see the saturation of energy error for $D = 4$ and $\chi = 32$ in Supplementary Figs. 5(a). Thus we consider that $\chi = 32$ is too small to obtain $\Delta E = 0.01$). The energy dynamics of these parameters are shown in Supplementary Fig. 5 together with a fitting curve by the power law for $(6, 6)$ and $(8, 8)$. The extracted lower bound of the computation time to obtain the ground state with the total energy error $\Delta E = 0.01$ is shown in Supplementary Table III. Compared with the case of DMRG, the estimated computation times of PEPS are much longer for the present small sizes. Based on the present estimations, we can conclude that the computation time of DMRG is shorter than that of PEPS around the quantum-classical crossover, $L \simeq 10$, although the asymptotic scaling of the computation time of PEPS with respect to the system size N_{site} is expected to be polynomial instead of exponential as in DMRG.

As in the case of DMRG, we may expect a speed-up of PEPS simulations by GPU or multi-node parallelization. Based on the similar consideration with DMRG, we estimated that these techniques may give us a shorter elapsed



Supplementary Figure 6. Dynamics of energy errors for $S = 1$ Haldane chain for various $N_{\rm site}$. (a) $S = 0$ sector (b) $S = 1$ sector. Horizontal dashed line indicate $\epsilon = E - E_0 = 0.01$.

time with a factor of 10^{-1} for the present relatively small systems.

Supplementary Table III. Parameters for estimating the computation time in PEPS simulations of 2d J_1-J_2 Heisenberg model with $J_2 = 0.5$. The ground state energy E_0 was independently estimated from the extrapolation of the DMRG simulation.

J_2/J_1	L_x	L_y	D	χ	E_0	Estimated computation time (s) for $\Delta E = 0.01$
0.5	4	4	4	48	-7.505556	93
0.5	6	6	4	48	-17.247352	6.3×10^4
0.5	8	8	4	48	-30.967671	1.7×10^5

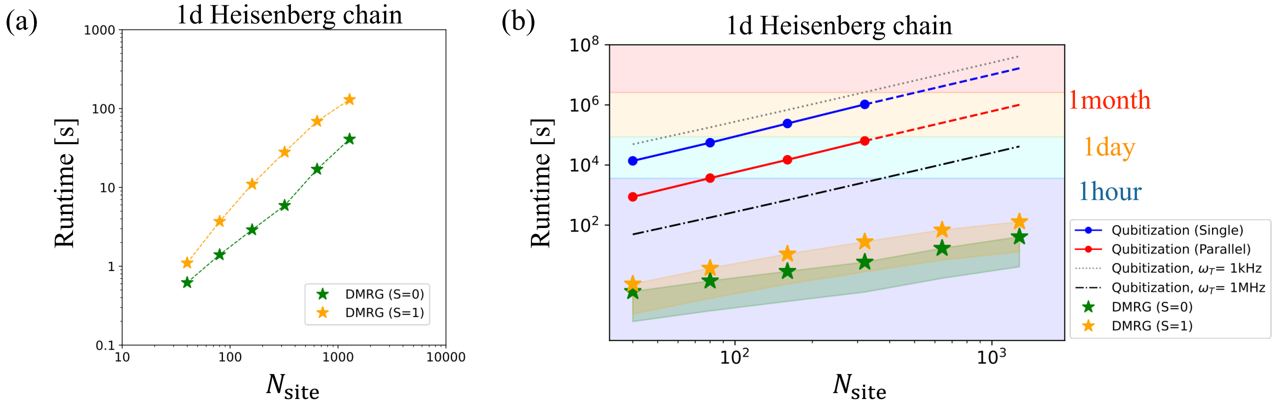
D. Error analysis and computation time estimate for one-dimensional $S = 1$ Heisenberg model

In this subsection, we explain the runtime analysis of the 1d $S = 1$ Heisenberg model. We consider the following Hamiltonian which is composed of $N_{\rm site}$ sites in total, with the edge sites being $S = 1/2$ spins and the $N_{\rm site} - 2$ bulk sites corresponding to $S = 1$ spins as

$$\mathcal{H} = \sum_{p=1}^{N_{\rm site}-3} \boldsymbol{S}_p \cdot \boldsymbol{S}_{p+1} + J_{\rm end} \left(\boldsymbol{s}_0 \cdot \boldsymbol{S}_1 + \boldsymbol{S}_{N_{\rm site}-2} \cdot \boldsymbol{s}_{N_{\rm site}-1} \right), \tag{6}$$

where $J_{\rm end}$ denotes the interaction amplitude of the $S = 1$ and $S = 1/2$ spins, and also we have introduced dot product to denote the Heisenberg interaction as $\boldsymbol{S}_p \cdot \boldsymbol{S}_q := \sum_{\alpha \in \{X,Y,Z\}} S_p^\alpha S_q^\alpha$. It has been pointed out that, thanks to the additional $S = 1/2$ spins $\boldsymbol{s}_0$ and $\boldsymbol{s}_{N_{\rm site}-1}$, we avoid degeneracy between the total spin $S = 0$ and $S = 1$ sectors in the thermodynamic limit by taking $J_{\rm end}$ sufficiently large [25, 30]. Here, we exclusively take $J_{\rm end} = 0.7$ in the current work. Since the entanglement entropy of the ground state is expected to obey 1d area law in this model, the DMRG algorithm is suitable to simulate larger system sizes. We also remark that, by extrapolating the energy gap between $S = 0$ and $S = 1$ sectors to $N_{\rm site} \to \infty$, we can estimate the famous Haldane gap [31].

To estimate the runtime of the classical DMRG algorithm, we executed the optimization using ITensor library as in the previous sections [23]. We used $U(1)$-symmetric DMRG to calculate the lowest energy states for $S_z = 0$ and $S_z = 1$ sectors, which we have confirmed to be the lowest energy states for $S = 0$ and $S = 1$, respectively. In this model, we easily obtain sufficiently small truncation errors around $\delta(D_{\rm max}) \simeq 10^{-10}$ with $D_{\rm max} \lesssim 300$. Thus, we regard the lowest energy obtained in each sector along DMRG simulation as E_0 and see $\epsilon = E - E_0$. Most of the calculations have been done by a single CPU, 3.3GHz dual-core Intel Core i7 on MacBook Pro (13-inch, 2016).



Supplementary Figure 7. (a) Estimated computation time to obtain $\epsilon = 0.01$ in the $S = 1$ Heisenberg chain defined in Eq. (6). (b) Runtime comparison with qubitization-based quantum phase estimation. Notations follow those of Supplementary Fig. 5 in the main text.

Supplementary Fig. 6 shows the dynamics of ϵ determined from the above procedure. In sharp contrast with the previous 2d models, we can easily reach $\epsilon \ll 10^{-2}$. In the case of the total $S = 0$ sector, we expect a gapped ground state, and therefore the required $D_{\max}$ is almost independent of N_{site}. This nature is also reflected in the very rapid decay of ΔE, as can be seen from Supplementary Fig. 6 (a). On the contrary, in the total $S = 1$ sector, we expect the lowest energy state to be gapless. Thus, the entanglement entropy gradually increases as $\log N_{\mathrm{site}}$, and therefore we need to increase $D_{\max}$ as we increase L. This fact is related to relatively slower decays of ϵ in Supplementary Fig. 6(b). We make another remark on the optimization dynamics in the total $S = 1$ sector; here, we observed that $D_{\max}$ becomes the largest at an intermediate step of the optimization dynamics. Such a non-monotonic behavior in the bond dimensions might be possibly caused by local minima, which presumably arose by imposing symmetry on tensors of MPS ansatz: To escape from local minima, we may need a larger bond dimension.

Finally, we estimate the computation time as the intersection of ϵ dynamics and $\epsilon = 0.01$. The obtained computation time is shown in Supplementary Fig. 7(a), and the parameters in the simulation are summarized in Supplementary Table IV. We see that the computation time is almost proportional to N_{site}, which is expected from the almost constant $D_{\max}$ in the optimized state. We see that the $D_{\max}$ in the total $S_z = 1$ sector slightly increases as we increase N_{site}. This is probably related to the gapless nature of the lowest energy state within the total $S = 1$ sector. Such weak N_{site} dependence also appears in the estimated computation time of the total $S_z = 1$ sector, which is always longer than the computation time for the gapped ground state in the total $S = 0$ sector.

Let us remark that runtime analysis on the quantum algorithm reveals that the quantum advantage in this model is unlikely. In Supplementary Fig. 7(b), we show the runtime in comparison with the qubitization-based quantum phase estimation (whose detail is described in Sec. VIII and IX). Since the phase estimation algorithm aiming for the current ϵ requires runtime that scales at least quadratically with respect to the system size, the large separation already at the moderate size of $N_{\mathrm{site}} = O(10^2)$ implies that there is a very low chance of DMRG simulation being surpassed by the quantum algorithm.

III. ESTIMATION OF COST ON GROUND STATE PREPARATION

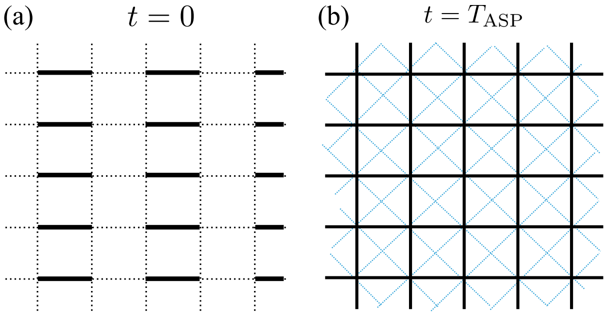
In this section, we provide details regarding the numerical simulation of Adiabatic State Preparation (ASP) of ground state. As in the main text, we consider the ASP up to infidelity $\epsilon_f = 1 - |\langle\psi_{\mathrm{GS}}|\psi(t_{\mathrm{ASP}})\rangle|$ where $|\psi_{\mathrm{GS}}\rangle$ is the exact ground state and $|\psi(t)\rangle$ is a quantum state that is realized via the following time-dependent Schrödinger equation:

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = H(t)|\psi(t)\rangle\,. \tag{7}$$

Here, the time-dependent Hamiltonian is defined as $H(t) = H(s(t)) = sH_f + (1-s)H_0$ for the target Hamiltonian H_f and the initial Hamiltonian H_0, and the smoothness of the dynamics is controlled by the interpolation function $s(t) : \mathbb{R} \mapsto [0, 1]$ $(s(0) = 0, s(t_{\mathrm{ASP}}) = 1)$. For the simulation of 2d J_1-J_2 Heisenberg model of J_2=0.5, we have taken

Supplementary Table IV. Parameters for estimating the computation time in DMRG simulations of Haldane chain defined as Eq. (6) with $J_{\mathrm{end}} = 0.7$.

total S_z	N_{site}	$D_{\max}$ in the optimized state	lowest energy	Estimated computation time (s) for $\Delta E = 0.01$
0	40	90	-53.293707633101	0.62
0	80	90	-109.35306916448	1.4
0	160	90	-221.47179222696	2.9
0	320	90	-445.70923835198	5.9
0	640	90	-894.18413060185	17
0	1280	90	-1791.13391510219	41
1	40	134	-52.861104764355	1.1
1	80	156	-108.934890997408	3.7
1	160	169	-221.058987851845	11
1	320	175	-445.298116034289	28
1	640	177	-893.773481949644	69
1	1280	179	-1790.7233920146	1.3×10^2



Supplementary Figure 8. Schematic picture of the initial and target Hamiltonian of the ASP for 2d J_1-J_2 Heisenberg Hamiltonian on square lattice. (a) Interaction configuration E_0 of the initial Hamiltonian H_0. (b) Interaction configuration of the target Hamiltonian. The black lines denote the nearest-neighbor J_1 interaction, where the blue dashed lines represent the J_2 interaction.

the initial state to denote a nearest-neighbor dimer covering E_0 as shown in Supplementary Fig. 8 as

$$H_0 = J_1 \sum_{(p,q)\in E_0} \sum_{\alpha \in \{X,Y,Z\}} S_p^\alpha S_q^\alpha. \tag{8}$$

It has been argued that the smoothness of the interpolating function $s(t)$ plays a crucial role to suppress nonadiabatic transition, such that we obtain logarithmic dependence $t_{\mathrm{ASP}} \propto \log(1/\epsilon_f)$ for carefully designed $s(t)$ [32]. While the bounds are derived under assumptions that $s(t)$ belongs to the Gevrey class (whose derivative at $t = 0, t_{\mathrm{ASP}}$ vanishes up to infinite order), it is practically sufficient to consider functions with vanishing derivative up to finite order (see Supplementary Fig. 4(a) in the main text). In this work, we have studied three types of interpolating functions as

$$s(t) = \begin{cases} t/t_{\mathrm{ASP}} \\ \mathcal{S}_\kappa(t/t_{\mathrm{ASP}}) \\ \mathcal{B}_\kappa(t/t_{\mathrm{ASP}}) \end{cases} \tag{9}$$

where $\mathcal{S}_\kappa(x) := \sin^2(\frac{\pi}{2}\mathcal{S}^{\kappa-2}(x))$ ($\mathcal{S}_0(x) = \frac{\pi}{2}x$). The last line is called the incomplete Beta functions that is defined as

$$\mathcal{B}_\kappa(x) := \frac{B_x(1+\kappa, 1+\kappa)}{B_1(1+\kappa, 1+\kappa)} \tag{10}$$

with $B_\lambda(a,b) := \int_0^\lambda z^{a-1}(1-z)^{b-1}dz$. The superscript κ denotes that the derivatives of functions $\mathcal{S}_\kappa$ and $\mathcal{B}_\kappa$ vanishes up to κ-th order at $t = 0, t_{\mathrm{ASP}}$. Obviously we have $\kappa = 0$ for the linear interpolation.

The numerical simulation of dynamics can be done by the using a standard technique to solve ordinary differential equation. We have concretely used the `sesolve` function in the Library QuTiP [33], which is based on the zvode routine provided by SciPy [34]. On the other hand, the dynamics simulation of the MPS state have been done following the time-dependent variational principle (TDVP) [35] solver implemented in ITensor. In our simulation, the time evolution is sliced into a time step of $\delta t = 0.05$, and have kept the maximum bond dimension to be $D = 512$ since the DMRG simulations for current setups suggest $1 - |\langle\psi_{D=512}|\psi_{D=3000}\rangle| < 10^{-3}$. This is sufficiently small compared to the target infidelity ϵ_f presented in Supplementary Fig. 4 in the main text, and therefore we safely take $D = 512$ for the adiabatic time evolution.

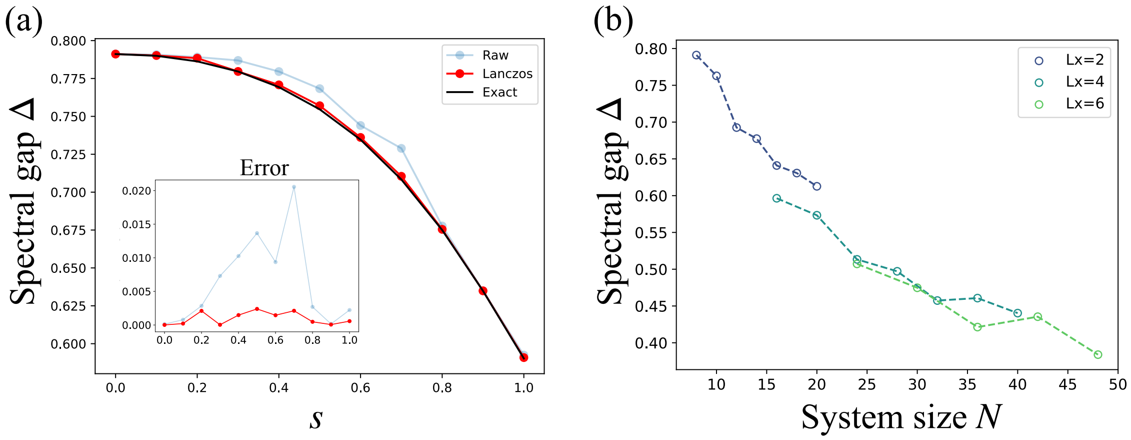
Finally, we describe how to estimate the spectral gap using the MPS state. Since the SU(2) symmetry is preserved along the entire ASP procedure, we focus on the spectral gap within the $S = 0$ symmetry sector instead of other low-lying states such as the triplet excitations (i.e., the excited state with $S = 1$). We find that it is convenient to employ the penalty method to compute the first excited state in $S = 0$ sector [36]. Concretely, we modify the Hamiltonian H as

$$H_{\rm symm} = H + \upsilon S^2, \tag{11}$$

where $\lambda \in \mathbb{R}$ is the penalty coefficient for the total spin operator $S^2 = \sum_{\alpha\in\{X,Y,Z\}}(S^\alpha_{\rm tot})^2 = \sum_{\alpha\in\{X,Y,Z\}}(\sum_p S^\alpha_p)^2$. We have exclusively used $\upsilon = 5$ in our numerical simulation. Once the l-th excited states of $H_{\rm symm}$ is simulated as $|\phi_l\rangle$, we perform a single Lanczos step so that

$$|\Psi_{\rm ex}\rangle = \sum_{l=1}^{l_{\max}} \eta_l |\phi_l\rangle \tag{12}$$

denotes the first excited state with coefficients $\{\eta_l\}$. For instance, a single Lanczos step with $l_{\max} = 5$ improves the accuracy of Δ by an order for system size of 4×4, and thus yields a reliable computational results (see Supplementary Fig. 9). The simulation is done with MPS of bond dimension $D \leq 3000$.



Supplementary Figure 9. Estimation of spectral gap Δ between ground state and the first excited state in the symmetry sector $S = 0$. (a) Improvement of accuracy along adiabatic path for 2d J_1-J_2 Heisenberg model on 4×4 lattice with open boundary condition. The accuracy is improved by an order for all s. (b) Scaling of Δ with lattice size of $L_x \times L_y$ for $L_x = 2, 4, 6$, where L_y is taken as most $L_y \leq 10$.

IV. QUANTUM ALGORITHMS FOR ESTIMATING ENERGY SPECTRA

A. Overview on quantum phase estimation

Quantum phase estimation (QPE) is a quantum algorithm designed to extract the eigenphase ϕ of a given unitary U up to m digits, using m ancilla qubits that indirectly read out the complex phase of the target system. More precise description of the algorithm originally proposed by Kitaev [37] is as follows: given an efficiently prepared trial state $|\psi\rangle = |0\rangle$ whose fidelity with the k-th eigenstate $|k\rangle$ of the unitary is given as $f_k = \|\langle k|\psi\rangle\|^2$, a single run of QPE returns a random variable $\hat{\phi}$ which corresponds to a m-digit readout of ϕ_k with probability f_k. By plugging a time evolution operator $U = e^{-iHt}$ of Hamiltonian H and take $|\psi\rangle$ to be efficiently-preparable approximation of the eigenstate, for instance, one can simulate the eigenspectrum of the target Hamiltonian H [38].

In this work, we focus on the simulation of ground state and its eigenenergy, within some target accuracy, of a quantum many-body Hamiltonian considered in condensed matter physics. Although the simulation of the ground state of a general Hamiltonian is a QMA-hard problem, we envision that quantum advantage can be realized in a practical sense, since the Hamiltonian simulation is BQP-complete and cannot be replaced be by any classical algorithm efficiently. In this regard, our goal is to elucidate the quantum-classical crosspoint based on the actual runtime to prepare a many-body quantum state up to some fixed accuracy in terms of the energy. The detailed procedure for classical algorithm has already been provided in Sec. II, and thus we detail on the quantum algorithm in the following. In particular, we explain the abstract protocol of cost/runtime estimation strategy in Sec. IV B, and then proceed to Sec. IV C, IV D in order to provide different flavors of Hamiltonian simulation techniques, which constitute the main building block of the phase estimation algorithm.

B. General cost estimation strategy for phase estimation

As we have briefly introduced in the main text, the gate complexity for running the QPE can be abstractly given as

$$C_{\text{total}} = C_{\text{SP}} + C_{\text{HS}} + C_{\text{QFT}^\dagger}, \tag{13}$$

where the cost C_{SP} is for the state preparation of the approximate ground state, $C_{\text{HS}} = 2^m C_{\text{CU}}$ is controlled Hamiltonian simulation, and $C_{\text{QFT}^\dagger}$ for the inverse quantum Fourier transformation (QFT). As we have explained in the main text, we exclusively focus on the cost from the second term of Eq. (13), and discuss the runtime solely from it.

Estimation of the runtime of the phase estimation is performed via two steps. First, we perform a comprehensive comparison between multiple flavors of Hamiltonian simulation, which are roughly divided into Trotter-based and post-Trotter methods. Given the explicit quantum circuit structure in individual algorithms, we compute the T-count respectively to determine the best among the existing methods. Note that such an estimation procedure is reasonable; in a situation when the logical Clifford operations can be efficiently implemented through stabilizer measurements and feedback, the runtime is dominated by the execution of T-gates, which is the most basic unit for non-Clifford operations under the surface code. In the second step, which is detailed in Sec. VIII, the runtime of the presumably best quantum algorithm is further analyzed by compiling the quantum circuit at the level of executable instructions. This yields a more precise runtime estimate with plausible quantum hardware setups.

One shall keep in mind that there are multiple source of algorithmic error in phase estimation. Our error analysis is based on findings in Ref. [39] that related the energy error ϵ with the error δ in the unitary circuit. For simplicity, in our work we have assumed $\delta = \epsilon/\lambda$ where λ is the L1 norm of Pauli operator coefficients. This total implementation error can be decomposed into several contributions as

$$\delta = \delta_{\text{PEA}} + \delta_{\text{syn}} \tag{14}$$

where δ_{PEA} and δ_{syn} denote the error from the phase estimation itself (due to finite readout digits) and the synthesis error, respectively. Concretely, we have

$$\delta_{\text{syn}} = \begin{cases} \delta_{\text{HS}} + \delta_{\text{Rot}} & (\text{qDRIFT}) \\ \delta_{\text{HS}} + \delta_{\text{Rot}} & (\text{random Trotter}) \\ \delta_{\text{HS}} + \delta_{\texttt{PREP}} & (\text{Taylorization}) \\ \delta_{\texttt{PREP}} & (\text{Qubitization}) \end{cases}, \tag{15}$$

where δ_{HS} is the error in approximating the Hamiltonian simulation unitary, δ_{Rot} is the gate synthesis error for Pauli rotation, and $\delta_{\texttt{PREP}}$ is the state synthesis error for PREPARE oracle which is practically dominated by the rotation synthesis. While previous works on Trotter-based method have revealed how to determine error budgets to minimize the error, there is no known closed-form expression that can analytically determine the ratio between error resources. We have performed numerical search on the error resource to find that we may practically fix as $\delta_{\text{PEA}} = 0.9\epsilon/\lambda$ and set other subdominant (i.e. logarithmically contributing) factors equally. Note that the scaling is inverse-linear with respect to δ_{PEA}, while for others it scales only polylogarithmically. This difference in the scaling already gives us sense that the dominant error resource shall be the finite-digit truncation in phase estimation.

In the remainder of this section, we introduce the Hamiltonian simulation techniques considered in our work, and also discuss their T-complexity and the effect of prefactors.

C. Trotter-based Hamiltonian Simulation methods for phase estimation

Among various flavors of Hamiltonian simulation algorithms for quantum computers, the ones based on the Suzuki-Trotter decomposition have been known from the earliest days. Since universal quantum computers are mostly designed to execute quantum gates that operate only on constant number of qubits, they cannot directly implement the matrix exponentiation as $U = e^{-iHt}$ where we assume that the Hamiltonian is decomposed into sum of Pauli operators as $H = \sum_l w_l H_l$ (we tune the phase so that $w_l > 0$). The Suzuki-Trotter decomposition refers to the deterministic procedure that approximates the unitary evolution by using a sequence of Hermitian operators $\mathcal{M} = (..., M_j, ...)$ to implement the product formula which is both spacially and temporally discretized as

$$U := \exp\left(-i \sum_l w_l H_l t\right) \approx \left(\prod_j e^{-iM_j t/r}\right)^r. \tag{16}$$

For instance, in the case of the first-order product formula, the Hermitian operators are simply taken as $\mathcal{M} = (..., w_l H_l, ...)$ such that

$$U_{\text{Trotter},1} = \left(\prod_l e^{-iw_l H_l t/r}\right)^r, \tag{17}$$

whose error from the exact implementation is bounded by the operator norm as

$$\|U - U_{\text{Trotter},1}\| = O\left(\frac{(L\Lambda t)^2}{r}\right). \tag{18}$$

Here, L denotes the number of decomposition for Hamiltonian, r is the number of repetitions, and $\Lambda = \max_l w_l$ is the largest coefficient. The higher order product formulae were also invented by Suzuki to improve the scaling with respect to the simulated time t [40]. For instance, the second order product formula is given by taking $\mathcal{M} = (w_1 H_1/2, ..., w_{L-1} H_{L-1}/2, w_L H_L, w_{L-1} H_{L-1}/2, ..., w_1 H_1/2)$ as

$$S_2(\tau) := \prod_{l=1}^{L} \exp(w_l H_l \tau/2) \prod_{l=L}^{1} \exp(w_l H_l \tau/2), \tag{19}$$

and further the $2k$-th order product formula is given by the following recursive procedure as

$$S_{2k}(\tau) := [S_{2k-2}(g_k \tau)]^2 S_{2k-2}((1-4g_k)\tau)[S_{2k-2}(g_k \tau)]^2, \tag{20}$$

where $g_k = 1/(4 - 4^{1/(2k-1)})$ $(k > 1)$. By taking the time evolution unitary as $U_{\text{Trotter},2k} = (S_{2k}(-it/r))^r$, it follows that the error of the Hamiltonian simulation can be given as

$$\|U - U_{\text{Trotter},2k}\| = O\left(\frac{(L\Lambda t)^{2k+1}}{r^{2k}}\right). \tag{21}$$

The product formula enables one to systematically improve the error with respect to, e.g., simulation time t. Meanwhile, when we count the gate complexity on quantum computers, the prefactor suffers from exponential increase with respect to the order k [41], and therefore is not preferred in practice. An intriguing alternative is to introduce the randomization into the gate compilation [42–44]. For instance, it has been pointed out that, by randomly permuting the sequence of operators $\{H_l\}$ in Eq. (19) for the second-order product formula improves the error dependence on L [44]. See Supplementary Table V for the comparison on the gate complexity and error scaling.

While the randomized Trotter methods have performed randomization concerning the order of the Pauli rotations, it was proposed by Campbell that we may stochastically choose the Pauli rotations according to their coefficients in the Hamiltonian to suppress the simulation cost for Hamiltonians that are composed of large numbers of Pauli with small coefficients [41]. To be concrete, according to the probability $p_l = w_l/\lambda$ where $\lambda := \sum_l w_l$, we choose the Pauli operator H_l and append the Pauli rotation $e^{-iH_l t/r}$ to the circuit, which is repeated for r times. It has been shown that the error scaling in such a technique, called the qDRIFT, is given rather by the L1 norm of the coefficients λ rather than the number of Paulis L [See Supplementary Table V.].

Algorithm	Error	Gate complexity
Trotter, 1st order	$O\left(\frac{(L\Lambda t)^2}{r}\right)$	$O\left(\frac{L^3(\Lambda t)^2}{\delta_{\rm HS}}\right)$
Trotter, $2k$-th order	$O\left(\frac{(L\Lambda t)^{2k+1}}{r^{2k}}\right)$	$O\left(\frac{L^{2+1/2k}(\Lambda t)^{1+1/2k}}{\delta_{\rm HS}^{1/2k}}\right)$
Random Trotter, $2k$-th order	$O\left(\frac{(L^{2k}(\Lambda t)^{2k+1})}{r^{2k}}\right)$	$O\left(\frac{L^2(\Lambda t)^{1+1/2k}}{\delta_{\rm HS}^{1/2k}}\right)$
qDRIFT	$O\left(\frac{2\lambda^2 t^2}{r}\right)$	$O\left(\frac{(\lambda t)^2}{\delta_{\rm HS}}\right)$

Supplementary Table V. Gate complexity and error scaling of Trotter-based methods to perform Hamiltonian simulation for time t.

Algorithm	$N_{\rm Rot}$	T-count	T-complexity
Random, 2nd order Trotter	$16\frac{\Lambda^3 L^2}{\epsilon^{3/2}}$	$N_{\rm Rot} \times (\Gamma \log_2 \delta_{\rm SS}^{-1} + \Xi)$	$O\left(\frac{N^2}{\epsilon^2}\log(N/\epsilon^{3/4})\right)$
qDRIFT	$35.5192\frac{\lambda^2}{\epsilon^2}$	$N_{\rm Rot} \times (\Gamma \log_2 \delta_{\rm SS}^{-1} + \xi)$	$O\left(\frac{N^2}{\epsilon^2}\log(N/\epsilon)\right)$

Supplementary Table VI. T-count of Trotter-based methods to perform phase estimation up to energy accuracy ϵ. Here we assume that the Pauli rotations are implemented with gate synthesis error $\delta_{\rm SS} = \delta_{\rm syn}/2N_{\rm Rot}$ where $\delta_{\rm S}$ is the total gate synthesis error related with the total energy accuracy as $\delta_{\rm syn} = 0.1\epsilon/\lambda$. The number of controlled Pauli rotations are estimated under the assumption of using the Hodges-Lehmann estimator, from which the T-counts are calculated by multiplying the cost per controlled Pauli rotation (Also see Sec. VI for T-count of basic quantum operations).

1. *T-count in Trotter-based methods*

The error analysis on the Hamiltonian simulation eventually allows one to analyze the cost to perform the phase estimation up to a target energy accuracy ϵ. For instance, Campbell estimated the number of digits m required to read out the energy with ϵ with success probability $1 - p_f$ as [41]

$$m = \log_2(\epsilon/2\lambda) + \log_2\left(\frac{1}{p_f} + 1\right) - 2. \tag{22}$$

By optimizing the gate synthesis accuracy for each control-U^{2^j} such that the overall gate count is minimized under a fixed total error, one finds that the total number of rotation gates is given as

$$N_{\rm Rot} \sim 133\frac{\lambda^2}{\epsilon^2((3/2)p_f)^3}. \tag{23}$$

While this analysis assumes to run the QPE for a single shot, Lee *et al.* introduced another layer of "randomization" [45]; by running the phase estimation multiple times, one may sample from the somewhat erroneous probability distribution, whose error can be efficiently suppressed by the Hodges-Lehman estimator to deal with the fat-tail errors. By assuming that the error in the output is symmetric, one obtains the improved number of rotations as

$$N_{\rm Rot} = 35.5192\frac{\lambda^2}{\epsilon^2}. \tag{24}$$

Given the number of rotations, one may estimate the T-count by multiplying with the gate synthesis cost for controlled rotations. See Supplementary Table VI for the summary of the cost required to perform the phase estimation. Note that we also perform a similar analysis in the case of the 2nd order random Trotterization.

D. Post-Trotter Hamiltonian simulation methods for quantum phase estimation

One of the main bottlenecks of the Trotter-based Hamiltonian simulation techniques is that their gate complexity is polynomial regarding the dynamics simulation accuracy δ_{HS}. Although one may in principle improve the gate-depth scaling as $O(\delta_{\mathrm{HS}}^{-1/2k})$ by employing $2k$-th order product formulas, one must pay the cost of significantly increased prefactor, which practically negates the advantage of the scaling in moderate-size problems. This has been recognized as roadblock that prevents us from achieving quantum advantage in a practical sense.

A class of Hamiltonian simulation techniques that achieves exponential improvement with respect to the accuracy δ_{HS} is called the "post-Trotter methods." While Trotter-based methods mainly consist of exponentiated Pauli rotations that act on the system qubits, in post-Trotter methods the class of available operations on the target system is vastly extended to include even non-unitary operations by making use of an extended Hilbert space. This technique is dubbed as *block-encoding*. For instance, Ref. [46, 47] proposed an algorithm that block-encodes the truncated Taylor series for the time evolution operator. The gate complexity to perform the phase estimation based on this so-called *Taylorization* method for lattice systems with size N with energy accuracy ϵ is $\frac{L\lambda}{\epsilon}\log(N/\delta_{\mathrm{HS}})/\log\log(N/\delta_{\mathrm{HS}})$ [48]. A major innovation pointed out by Refs. [49, 50] after the proposal of Taylorization is that one may completely eliminate the Trotter or Taylor errors in phase estimation by simulating $e^{i\arccos(H/\lambda)}$ instead of $e^{iHt/\lambda}$. This idea has been combined with the concept of *qubitization* [51] to achieve even more efficient implementation of phase estimation algorithm [52]. As we later describe more in detail in Sec. IV D 2, we construct oracles such that the operation of the block-encoding unitary can be "qubitized" into a direct sum of two-dimensional subspace. Each of such "single-qubit rotation" encodes the eigenvalue of the target operator, and therefore can be directly plugged in as a phase estimation subroutine.

In the following, we first introduce the concept of block encoding and its two core subroutines PREPARE and SELECT. Then, we further introduce the qubitization and Taylorization algorithms, which are two representative post-Trotter methods to perform Hamiltonian simulation for phase estimation.

1. *Block encoding for linear combination of unitaries*

It is instructive to start from the definition of the term block encoding. Let us define that block encoding of some bounded-norm target operator $A \in M_d(\mathbb{C})$ (d: Hilbert space dimension) is realized when the action of an unitary operator in an extended Hilbert space $V \in M_{d+d_a}(\mathbb{C})$ (d_a: auxiliary Hilbert space dimension) reproduces A at specific subspace. For concreteness, let us consider n-qubit target system with a ancillary qubits. We say that "A is block-encoded into V via signal state $|\mathcal{L}\rangle$" if the following holds:

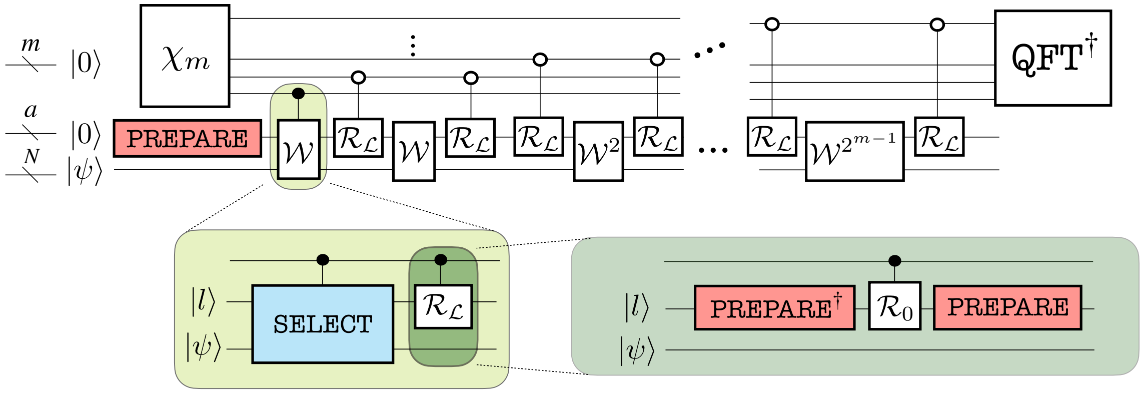
$$(\langle\mathcal{L}| \otimes I^{\otimes n})V(|\mathcal{L}\rangle \otimes I^{\otimes n}) = A, \tag{25}$$

where $|\mathcal{L}\rangle$ is a state in auxiliary a-qubit Hilbert space and I denotes the identity. Note that we have assumed that $\|A\| \leq 1$ without loss of generality, since we can rescale the operator as A/α ($\alpha \in \mathbb{R}$) if needed. While we may also consider block encoding under multiple signal states as $\Pi V \Pi = A$ where $\Pi = \sum_{\mathcal{L}} \Pi_{\mathcal{L}} = \sum_{\mathcal{L}} |\mathcal{L}\rangle\langle\mathcal{L}|$, here we focus on Eq. (25) for simplicity. It is illustrative to take $|\mathcal{L}\rangle = |0\rangle^{\otimes a}$ and rewrite Eq. (25) into an equivalent expression that holds for any target state:

$$V = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \tag{26}$$

Observe from Eq. (26) why the technique is called "block-encoding." Since we can obtain the desired operation $A|\psi\rangle$ with finite probability by post-selecting the measured ancillary qubits to be $|0\rangle^{\otimes a}$, this is also referred to as "probabilistic implementation of A." In reality, we design the algorithm (e.g. use amplitude amplification) so that post selection is not explicitly required.

In phase estimation algorithm based on post-Trotter methods, one aims to extract information on the eigenspectra by making multiple queries to the oracular unitary which block-encodes the Hamiltonian. The underlying assumption is that, there is a systematic and efficient way to encode the spectral information of the Hamiltonian into the oracular unitary V. As we later observe in the algorithms such as the qubitization and Taylorization, this question essentially falls into the problem of implementing two oracular subroutines named PREPARE and SELECT efficiently. Roughly speaking, PREPARE takes the ancillary quantum state to the properly block-encoded subspace, and SELECT performs the block encoding of the Hamiltonian in the subspace given by the first oracle PREPARE.



Supplementary Figure 10. Quantum circuit structure of qubitization-based phase estimation algorithm. The definitions of PREPARE and SELECT are given in Eqs. (28), (30), where the target block-encoded operator A is taken as the Hamiltonian here.

To explicitly write down the operation by two subroutines, first, let us note that any operator A can be decomposed into a linear combination of unitaries as

$$A = \sum_l w_l A_l \quad (w_l > 0), \tag{27}$$

where w_l is a real positive coefficient of the unitary operator A_l. For n-qubit systems, A_l can be thought to be a tensor product of multiple Pauli operators with arbitrary complex phase. The first oracle PREPARE is often introduced to perform the following transformation that encodes all L coefficients of the Hamiltonian into the ancillary qubits as

$$|\mathcal{L}\rangle := \mathtt{PREPARE}\,|0\rangle^{\otimes a} = \sum_{l=1}^{L} \sqrt{\frac{w_l}{\lambda}}\,|l\rangle\,, \tag{28}$$

where $\lambda = \sum_l w_l$ is the L_1 norm of the coefficients of Hamiltonian. Given the orthogonal bases $\{|l\rangle\}_l$ that encodes the information of the coefficients, next we construct the second oracle SELECT so that the desired operator A is realized in the subspace as

$$(\langle\mathcal{L}| \otimes I)\mathtt{SELECT}(|\mathcal{L}\rangle \otimes I) = \frac{A}{\lambda}. \tag{29}$$

While the construction of SELECT oracle is not unique, it is instructive to introduce an example as follows:

$$\mathtt{SELECT} = \sum_l |l\rangle\langle l| \otimes A_l + \sum_{\bar{l}} |\bar{l}\rangle\langle\bar{l}| \otimes I, \tag{30}$$

where we have introduced $\bar{l}$ to denote computational bases of ancillary qubit system on which SELECT acts trivially.

2. *Qubitization*

Ref. [51] pointed out that, by choosing the operator A in Eq. (27) to be the Hamiltonian of the target system, one may construct a unitary that invokes rotational operation for each eigenstates. Since all the rotation is performed disjointly on two-dimensional subspace, such a specific form of block encoding is referred to as the *qubitization*.

Let $A = \sum_\chi \chi\,|\chi\rangle\langle\chi|$ be a target Hermitian operator which we wish to block-encode into the SELECT operator via the signal state $|\mathcal{L}\rangle = \mathtt{PREPARE}\,|0\rangle$ as in Eq. (28), (30), (29). To understand the "qubitized" picture, we focus on a single eigenstate $|\chi\rangle$, and also introduce an orthogonal state $|\perp_\chi\rangle = \frac{1}{\sqrt{1-\chi^2}}(\mathtt{SELECT} - \chi I)\,|\mathcal{L}\rangle\,|\chi\rangle$ so that

$$\mathtt{SELECT}\,|\mathcal{L}\rangle\,|\chi\rangle = \chi\,|\mathcal{L}\rangle\,|\chi\rangle + \sqrt{1-\chi^2}\,|\perp_\chi\rangle\,. \tag{31}$$

We say that the oracles are qubitized if the action of the SELECT is closed within the two-dimensional subspace $\mathcal{H}_\chi = \mathrm{Span}(|\mathcal{L}\rangle\,|\chi\rangle\,, |\perp_\chi\rangle)$, which is known to be satisfied if and only if $\langle\mathcal{L}|\mathtt{SELECT}^2|\mathcal{L}\rangle = I$ [51]. Observe that

$\texttt{SELECT}$ defined as Eq. (30) satisfies this condition; A is now Hermitian and hence can be decomposed as $A = \sum_l w_l A_l$ with $w_l > 0$ and $A_l^2 = I$. Meanwhile, this does not hold if A is not Hermitian (as in the case of truncated Taylor series of unitary time evolution operator). We find that other transition amplitudes can be computed explicitly as

$$\begin{aligned} \langle\mathcal{L}|\langle\chi|\,\texttt{SELECT}\,|\perp_\chi\rangle &= \langle\mathcal{L}|\langle\chi|\,\texttt{SELECT}(\texttt{SELECT}-\chi I)\,|\mathcal{L}\rangle|\chi\rangle/\sqrt{1-\chi^2} && (32)\\ &= \frac{1-\chi^2}{\sqrt{1-\chi^2}} = \sqrt{1-\chi^2}, && (33)\\ \langle\perp_\chi|\texttt{SELECT}|\perp_\chi\rangle &= \langle\mathcal{L}|\langle\chi|\,(\texttt{SELECT}-\chi I)\texttt{SELECT}(\texttt{SELECT}-\chi I)\,|\mathcal{L}\rangle|\chi\rangle/(1-\chi^2) && (34)\\ &= \langle\mathcal{L}|\langle\chi|\,(\texttt{SELECT}-2\chi I+\chi^2\texttt{SELECT})\,|\mathcal{L}\rangle|\chi\rangle/(1-\chi^2) && (35)\\ &= \frac{-\chi(1-\chi^2)}{1-1\chi^2} = -\chi, && (36) \end{aligned}$$

and therefore the action within the effective Hilbert space H_χ is given as follows:

$$\texttt{SELECT}_\chi = \begin{pmatrix} \chi & \sqrt{1-\chi^2} \\ \sqrt{1-\chi^2} & -\chi \end{pmatrix}. \tag{37}$$

Observe that the action satisfies $\texttt{SELECT}_\chi\texttt{SELECT}_\chi^\dagger = I$ and indeed closed within the subspace.

The framework of the qubitization enables one to perform a vast class of polynomial transformation on the eigenspectrum of A as Poly(A). This implies that, one can build up a new block encoding of an operator from an existing block encoded operator, just as one develops a new algorithm by combining subroutines [53]. For instance, given a block encoding of A, one may construct a block encoding of polynomial approximation for e^{iA} , from which one may also derive a block encoding of A^{-1}. Such a methodology to manipulate eigenvalues enable one to extract various information from the block encoded operator.

Now we have confirmed that the oracles defined as in Eqs. (28), (30) are readily qubitized. Let us next proceed to construct a Szegedy-type walk opeartor as $\mathcal{W} = \mathcal{R}_\mathcal{L}\cdot\texttt{SELECT}$, which combines $\texttt{SELECT}$ with another reflection operator defined as $\mathcal{R}_\mathcal{L} = 2|\mathcal{L}\rangle\langle\mathcal{L}| - I$. It is straightforward to find that $\mathcal{W}$ is also qubitized in each two-dimensional subspace $\mathcal{H}_k$ which is defined from the k-th eigenstate of the Hamiltonian:

$$\mathcal{W}_k = \begin{pmatrix} E_k/\lambda & \sqrt{1-(E_k/\lambda)^2} \\ -\sqrt{1-(E_k/\lambda)^2} & E_k/\lambda \end{pmatrix} = e^{i\arccos(E_k/\lambda)Y}, \tag{38}$$

where E_k denotes the k-th eigenenergy. Therefore, this walk operator not only block-encodes the Hamiltonian in the signal space as $\langle\mathcal{L}|\mathcal{W}|\mathcal{L}\rangle = e^{i\arccos(H/\lambda)}$, but also provides efficient access to its entire eigenspectra.

Using the oracles $\texttt{PREPARE}$ and $\texttt{SELECT}$ introduced above, we construct the entire circuit as shown in Supplementary Fig. 10, where we have also introduced the preparation circuit for the readout qubits as [52]

$$\chi_m|0\rangle^m = \sqrt{\frac{2}{2^m+1}}\sum_{i=0}^{2^m-1}\sin\left(\frac{\pi(i+1)}{2^m+1}\right)|i\rangle\,. \tag{39}$$

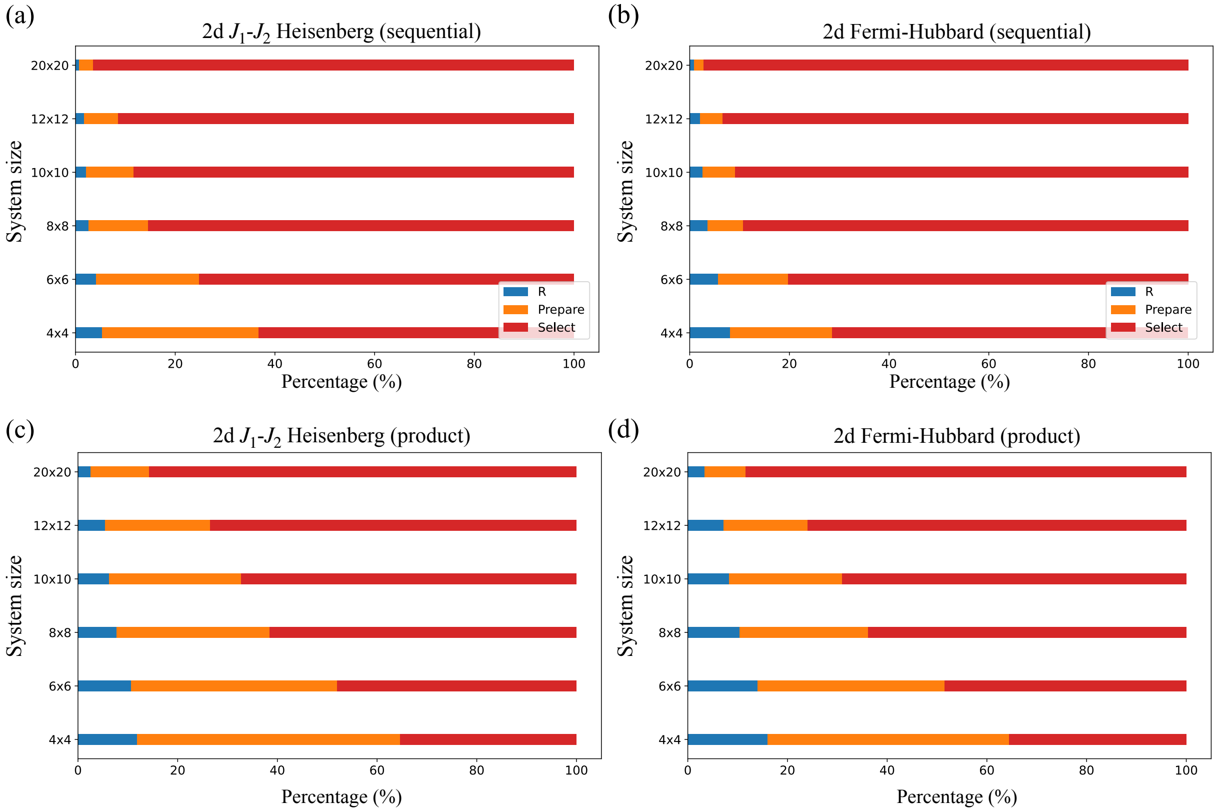
This consumes T-gates of $\tilde{O}(m)$ and hence neglected in the resource analysis in the present work. It is crucial to the resource estimate that the number of readout digits m is determined from the number of repetition r regarding the the quantum walk operations as [45, 54, 55]

$$r = \left\lceil\frac{\pi\lambda}{2\epsilon}\right\rceil < 2^m. \tag{40}$$

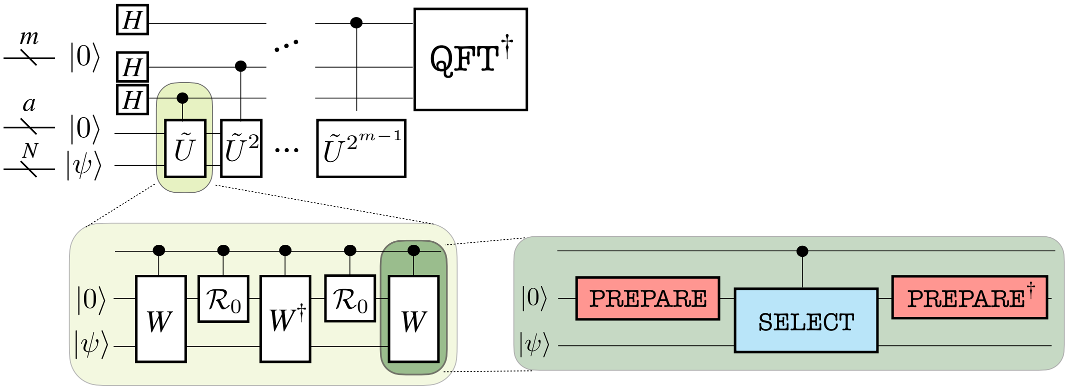
This leads us to obtain the relation between the cost (i.e., T-count or detailed runtime) of the global algorithm and the oracles as

$$C_\text{HS} \sim r\times(C_\text{P} + C_{\text{P}^\dagger} + C_\text{S} + 2C_{\mathcal{R}_0}), \tag{41}$$

where $C_{\text{P}(\dagger)}$ and C_S denote the cost required for the $\texttt{PREPARE}$ and $\texttt{SELECT}$ oracles, respectively, and $C_{\mathcal{R}_0}$ is the cost for the reflection by $|0\rangle$. By explicitly constructing the quantum circuits as shown in Sec. V, we find that the cost for $\mathcal{R}_0$ and $\texttt{PREPARE}$, which is only logarithmic with system size, indeed become asymptotically negligible (See Supplementary Fig. 11).



Supplementary Figure 11. The origin of T-count for qubitization algorithm with two flavors that sequentially block-encodes each Pauli one by one (sequential) and the one that exploits the translational symmetric local connectivity (product) for (a, c) 2d J_1-J_2 Heisenberg model (J_2/J_1=0.5) and (b, d) 2d Fermi-Hubbard model (U/t=4) on square lattice. The logarithmic contribution from the reflection and PREPARE oracles become negligible in the asymptotic limit.



Supplementary Figure 12. Quantum circuit structure of Taylorization-based phase estimation algorithm. The definitions of PREPARE and SELECT are given in Eqs. (28), (30), where the target operator is taken as the truncated Taylor series of the time evolution unitary.

3. *Taylorization*

Taylorization is one of the earliest attempts to overcome the polynomial gate complexity Poly($1/\delta_{\mathrm{HS}}$) in Trotter-based Hamiltonian simulation methods [49, 50]. In essence, the algorithm uses the technique of block encoding to approximate the time evolution operator $U = \exp(-iAt)$ with the repetitive operation of truncated Taylor series up to K-th order (See Supplementary Fig. 12), which results in polylog($1/\delta_{\mathrm{HS}}$) gate complexity. To be concrete, the time evolution is divided into r steps as $U = (U_r)^r$ with $U_r = \exp(-iAt/r)$, where each step is approximated as

$$\widetilde{U}_r := \sum_{k=0}^{K} \frac{(-iAt/r)^k}{k!} H^k = \sum_{k=0}^{K} \sum_{l_1,\ldots,l_k} w_{l_1} \cdots w_{l_K} A_{l_1} \cdots A_{l_K}. \tag{42}$$

Hyperparameters such as K and r shall be chosen in order to predict the output energy with target precision ϵ. Following the discussion of the qubitization and also Ref. [47, 56], we take

$$r = \frac{\pi}{2\delta_{PEA}} \left(\sim \frac{\pi\lambda}{2\epsilon} \right), \quad K = \left\lceil -1 + \frac{2\log(2r/\delta_{\rm HS})}{\log\log(2r/\delta_{\rm HS}) + 1} \right\rceil, \tag{43}$$

where $\lambda = \sum_l w_l$ is the total sum of coefficients for $A = \sum_l w_l A_l$. Note that the scaling of $O(1/\epsilon)$ reflects the time-energy uncertainty relation in order to extract the eigenstate energy.

Once we construct `PREPARE` and `SELECT` for truncated Taylor series Eq. (42), we find that the block encoding is realized via a walk operator

$$W := (\texttt{PREPARE}^\dagger)(\texttt{SELECT})(\texttt{PREPARE}), \tag{44}$$

whose explicit operation is given using $s = \sum_{k=0}^{K} \sum_{l_1,\ldots,l_K} w_{l_1} \cdots w_{l_K}$ as

$$W |0\rangle |\psi\rangle = \frac{1}{s} |0\rangle \widetilde{U}_r |\psi\rangle + \sqrt{1 - \frac{1}{s^2}} |\perp\rangle. \tag{45}$$

Although the unwanted second term seem to deteriorate the simulation, we may tune the amplitude of the target state via the oblivious amplitude estimation introduced in [46]. Namely, we rescale the time evolution as $s = 2$ so that the combination of the walk operator and a reflection operator on the ancillary qubits $\mathcal{R}_0 = I - 2|0\rangle\langle 0|$ gives [47]

$$W\mathcal{R}_0 W^\dagger \mathcal{R}_0 W |0\rangle |\psi\rangle = -|0\rangle \widetilde{U} |\psi\rangle. \tag{46}$$

Finally, by plugging the controlled operation of $\widetilde{U}$ into the quantum circuit of phase estimation, we obtain the circuit structure of the entire algorithm as in Supplementary Fig. 12. The overall cost is given as follows,

$$C_{\rm HS} \sim r \times (3(C_{\rm P} + C_{\rm P^\dagger} + C_{\rm S}) + 2C_{\mathcal{R}_0}), \tag{47}$$

where $C_{\rm P}$ and $C_{\rm S}$ is obtained from K-fold multiplication on cost for the qubitization algorithm, while $C_{\mathcal{R}_0}$ is simply the cost for multi-controlled Z-gate operated on ancillas. In practice, the number of repetitions r is around tens of thousands (see also Sec. IX) and K is around 30 to 40 for the target models with system size of above hundreds. This indicates that each unit oracles are executed for 10^{5-6} times.

V. POST-TROTTER ORACLES FOR LATTICE MODELS

In Sec. IV D, we have introduced the technique of block encoding, which allows us to perform Hamiltonian simulation with gate complexity that scales polylogarithmically with respect to the target accuracy $\delta_{\rm HS}$. In this section, we further provide the explicit construction of quantum circuits that cover a broad range of lattice models considered in condensed matter physics. We introduce two methods to construct oracles, which we refer to as sequential and product-wise constructions.

The sequential construction is designed to block-encode each Pauli operators one by one. Given L terms, we use $\lceil \log L \rceil$ ancillary qubits for `PREPARE` oracle, so that l-th computational basis is used to block-encode the l-th Pauli operator P_l by the `SELECT` oracle. On the other hand, the core idea of product-wise structure is utilize the translational symmetric structure of the connectivity; we encode the information on local connectivity and the interaction amplitude into `PREPARE` oracle, while we keep `SELECT` rather agnostic to the geometry. Such a construction turns out to be beneficial compared to the sequential one when the connectivity is dense; for instance, when we have all-to-all G-local interaction, the gate complexity drops from $O(N^G)$ to $O(GN)$ [57].

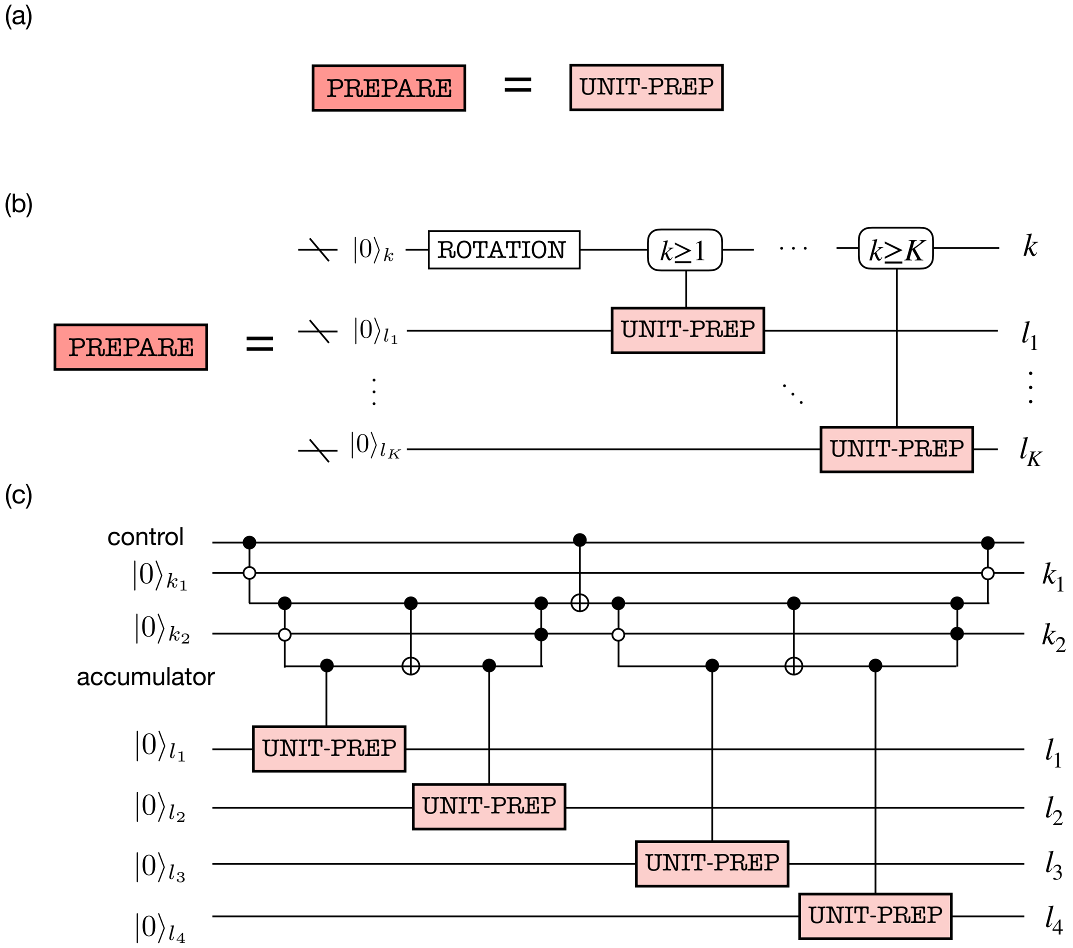
In the following, abstract descriptions are provided each for `PREPARE` and `SELECT` oracles in Secs. V A and V B, respectively. Subsequently we give constructions for quantum spin models and fermionic models whose interaction is geometrically local and translationally invariant. Then, we proceed to show examples in concrete models: spin-1/2 J_1-J_2 Heisenberg model and Fermi-Hubbard model on the square lattice, and also the spin-1 Heisenberg chain. From the results of T-count analysis combined with that for Trotter-based methods presented in Sec. IV C, we compare the order of computational cost between various phase estimation subroutines to conclude that the qubitization algorithm gives the best practice among the Hamiltonian simulation subroutines. We further guide readers to Sec. VIII for more detailed runtime analysis that compiles quantum circuit of the qubitization algorithm into executable instructions. We importantly remark that we have chosen the product-wise construction to the compare T-counts between Hamiltonian simulation subroutines, while the for the runtime analysis we focus on the sequential type of `SELECT` oracle for the sake of simplicity.

A. Abstract structure of `PREPARE`

Here, we present the abstract structure of `PREPARE` circuit that achieves polylogamithmic cost, in terms of both T-count and ancillary qubit count, for translationally-invariant G-local Hamiltonians introduced in Sec. I as

$$H = \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} w_{\boldsymbol{\mu},\boldsymbol{\alpha}} \hat{\Lambda}^{(\boldsymbol{\alpha})}_{\boldsymbol{p}} \hat{\Lambda}^{(\boldsymbol{\alpha})}_{\boldsymbol{p}+\boldsymbol{\mu}_1} \cdots \hat{\Lambda}^{(\boldsymbol{\alpha})}_{\boldsymbol{p}+\boldsymbol{\mu}_{G-1}}, \tag{48}$$

where $\boldsymbol{p} = (p_1, \dots, p_d)$ labels the site (including the sublattice structure), $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{G-1})$ is a set of vectors that identifies the connection between interacting sites, $\hat{\Lambda}^{\boldsymbol{\alpha}}_{\boldsymbol{p}}$ is an operator for microscopic degrees of freedom (e.g. spin, fermion, boson) on site $\boldsymbol{p}$, and $w_{\boldsymbol{\mu},\boldsymbol{\alpha}}$ is the amplitude of the interaction. As we display in Supplementary Fig. 13, both the `PREPARE` for qubitization and Taylorization consist of the identical subroutine which we call in this paper as `UNIT-PREP`. Therefore, let us first focus on the case of qubitization and later mention the case of Taylorization.



Supplementary Figure 13. `PREPARE` structure of (a) qubitization and (b) Taylorization, whose detailed circuit structure for $K = 4$ case is described in (c).

First we introduce the sequential construction, which straightforwardly encodes all the information of coefficients as

$$\texttt{UNIT-PREP} : |0\rangle \mapsto |\mathcal{L}\rangle = \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{\frac{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}}{\lambda}} |\boldsymbol{p}\rangle |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle . \tag{49}$$

Thanks to the translational invariance of the Hamiltonian, we may split the ancilla for geometrical position $\boldsymbol{p}$ and interaction configurations $\boldsymbol{\mu}, \boldsymbol{\alpha}$. Note that in the graphical description in Supplementary Fig. 14(a) we have introduced $\texttt{UNIFORM}_N |0\rangle = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} |n\rangle$.

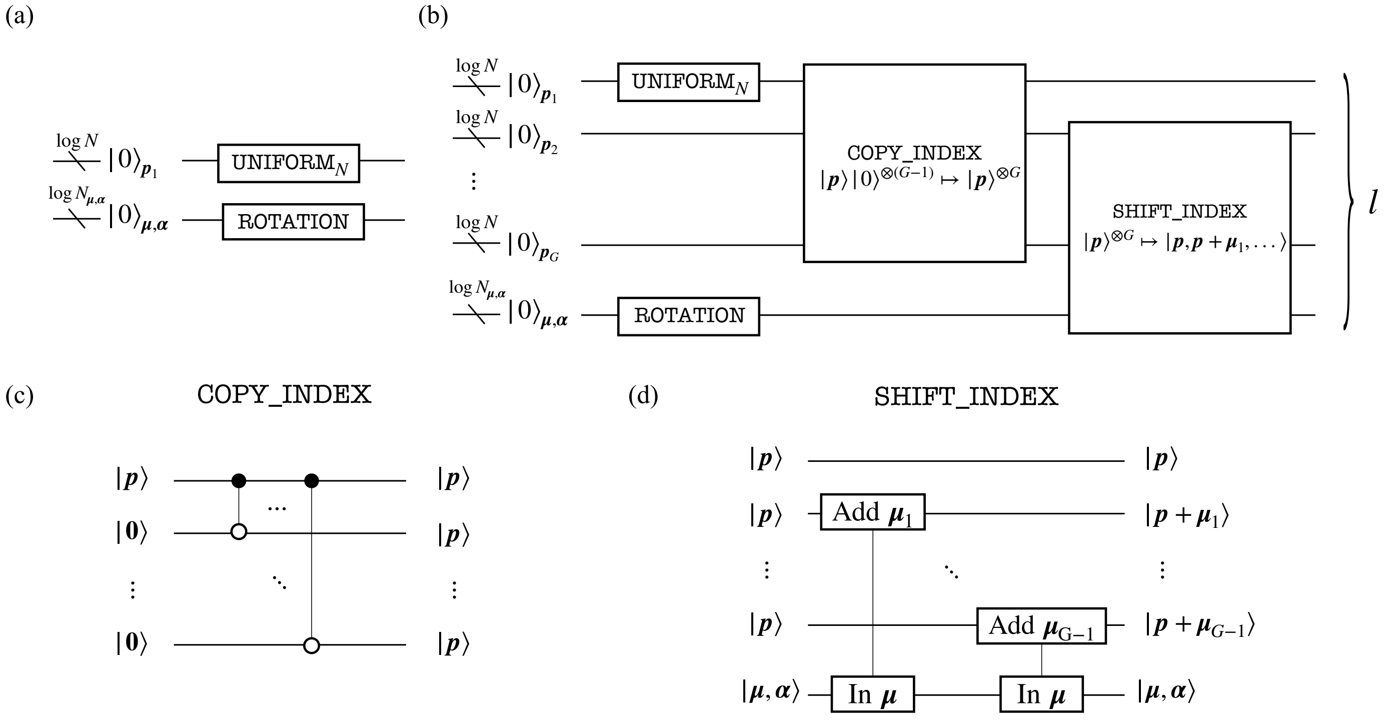
One evident difference between the product-wise construction from the sequential one is the G-fold increase in use of ancilla qubits, from which we benefit in terms of T-count in SELECT oracle (See. V B). The general form of product-wise UNIT-PREP encodes the interaction amplitudes into ancillary qubits as follows:

$$\texttt{UNIT-PREP} : |0\rangle \mapsto |\mathcal{L}\rangle = \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{\frac{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}}{\lambda}} |\boldsymbol{p}, \boldsymbol{p}+\boldsymbol{\mu}_1, ..., \boldsymbol{p}+\boldsymbol{\mu}_{G-1}\rangle |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle . \tag{50}$$

As shown in Supplementary Fig. 14(b), this can roughly be decomposed into three steps as

$$\begin{aligned} |0\rangle &\xrightarrow{\text{Rotations}} \left(\sum_{\boldsymbol{p}} |\boldsymbol{p}\rangle |0\rangle^{(G-1)\log N}\right) \otimes \left(\sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}} |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle\right) = \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}} |\boldsymbol{p}\rangle |0\rangle^{(G-1)\log N} |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle \\ &\xrightarrow{\texttt{COPY\_INDEX}} \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}} |\boldsymbol{p}, \boldsymbol{p}, ..., \boldsymbol{p}\rangle |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle \\ &\xrightarrow{\texttt{SHIFT\_INDEX}} \sum_{\boldsymbol{p}} \sum_{\{\boldsymbol{\mu},\boldsymbol{\alpha}\}} \sqrt{\frac{w_{\boldsymbol{\mu},\boldsymbol{\alpha}}}{\lambda}} |\boldsymbol{p}, \boldsymbol{p}+\boldsymbol{\mu}_1, ..., \boldsymbol{p}+\boldsymbol{\mu}_{G-1}\rangle |\boldsymbol{\mu},\boldsymbol{\alpha}\rangle \, (= |\mathcal{L}\rangle), \end{aligned} \tag{51}$$

where N is the total system size. Note that we have abbreviated overall coefficients in the first and second lines for the sake of simplicity.



Supplementary Figure 14. General structure of PREPARE for lattice systems with G-local translationally invariant interactions. (a) PREPARE oracle for sequential-type construction. (b) PREPARE oracle for product-wise construction. The quantum circuit structure are provided for (c) COPY_INDEX and (d) SHIFT_INDEX. For detailed information on constant adders, see Sec. VI.

	Sequential	Product		
T-count	$4L-4$	Rotations	COPY_INDEX	SHIFT_INDEX
		$8\log N+(N_{\boldsymbol{\mu},\boldsymbol{\alpha}}+2)\log(1/\delta_{\rm ss})$	0	$8(G-1)\log N+4(G-1)\log N_{\boldsymbol{\mu},\boldsymbol{\alpha}}+O(1)$
		Total		
		$8G\log N+4(G-1)\log N_{\mu,\alpha}+(N_{\boldsymbol{\mu},\boldsymbol{\alpha}}+2)\log(1/\delta_{\rm SS})$		
Ancilla count	$\log N+\log N_{\boldsymbol{\mu},\boldsymbol{\alpha}}$	Rotations	COPY_INDEX	SHIFT_INDEX
		$\log N+\log N_{\boldsymbol{\mu},\boldsymbol{\alpha}}$	$G\log N$	$(G+1)\log N+\log N_{\boldsymbol{\mu},\boldsymbol{\alpha}}$
		Total		
		$(G+1)\log N+\log N_{\boldsymbol{\mu},\boldsymbol{\alpha}}$		

Supplementary Table VII. Quantum resource required to implement PREPARE oracle of sequential and product-wise construction of qubitization method. Note that the ancilla qubit count for the product-wise construction denotes the number of involved qubits, and thus not simply the summation over the ancilla qubit counts. Also, we have neglected the $O(1)$ contributions here for simplicity.

The main resource of T-count is the synthesis of the rotation gates in the first step, and the adder circuit for SHIFT_INDEX (note that COPY_INDEX can be performed solely by CNOT gates). Regarding the dynamics simulation error $\delta_{\rm HS}$, the order of the T-count can be obtained as $8\log N+N_{\boldsymbol{\mu},\boldsymbol{\alpha}}\log(1/\delta_{\rm ss})$ where $\delta_{\rm ss}=\delta_{\rm syn}/r$ where the required number of ancillary qubits can be estimated as $(G+1)\log N+\log N_{\boldsymbol{\alpha},\boldsymbol{\mu}}+O(1)$, with $N_{\boldsymbol{\alpha},\boldsymbol{\mu}}=N_{\boldsymbol{\alpha}}N_{\boldsymbol{\mu}}$ denoting the total number of combinations of $(\boldsymbol{\alpha},\boldsymbol{\mu})$. Here, we assume to use the efficient implementation of logical adder that use auxiliary qubits to store the information of "carry-on" [58], resulting in additional use of ancillary qubits of $\log N_{\boldsymbol{\alpha},\boldsymbol{\mu}}$. Note that, for the PREPARE circuit for Taylorization, we prepare $N_{\boldsymbol{\alpha},\boldsymbol{\mu}}\log L+\log N_{\boldsymbol{\alpha},\boldsymbol{\mu}}$ ancilla qubits to encode information on higher order contribution from Taylor expansion, as can be seen from Supplementary Fig. 13.

Last but not least, let us mention how to deal with the boundary condition. Be aware that the above-mentioned construction assumes translationally invariant Hamiltonian defined under the periodic boundary condition. However, of course we are not limited to this situation: open, cylinder, or twisted boundary conditions can also be readily treated as well. One straightforward way is to encode the boundary terms individually, which can be done by sparing T-count of $\log(N^{d-1/d})$. However, in this work we consider an alternative method to simulate Hamiltonians defined on cylindrical boundary condition which solely relies on simple postprocessing and rescaling of the Hamiltonian. The details are explained in Sec. VII.

B. Abstract structure of SELECT

Next, we present the sequential and product-wise structure of SELECT oracles that are used to block-encode an operator $H=\sum_{l=1}^{L}w_lH_l$. In the sequential construction, we perform one-hot encoding of individual unitary H_l as defined in Eq. (30) as

$$\mathtt{SELECT}=\sum_{l}|l\rangle\langle l|\otimes H_l+\sum_{\bar{l}}|\bar{l}\rangle\langle\bar{l}|\otimes I. \tag{52}$$

In particular, we assume that every H_l is a tensor product of Pauli operators (with global phase $\pm i$ or ± 1) so that we can implement H_l and controlled H_l with Clifford operations. Babbush *et al.* has proposed a quantum circuit that realizes Eq. (52) with T-count of $4L-4$ [52], whose structure is explicitly shown in Supplementary Fig. 15(a).

While Supplementary Fig. 15(a) is applicable to general form of Hamiltonian, we find that in many cases the translational symmetry allows even more efficient implementation via product-wise construction. The product-wise qubitization is designed so that the (hyper-)connectivity of local terms are encoded solely in PREPARE, while the SELECT oracle is agnostic to the geometric information. Assume that we have PREPARE oracle that generates a signal state

$$\mathtt{PREPARE}|0\rangle=|\mathcal{L}\rangle=\sum_{\boldsymbol{\alpha},\boldsymbol{\mu}}\sum_{\boldsymbol{p}}\sqrt{\frac{w_{\boldsymbol{\alpha},\boldsymbol{\mu}}}{\lambda}}|\boldsymbol{p}_1,\boldsymbol{p}_1+\boldsymbol{\mu}_1,...,\boldsymbol{p}_1+\boldsymbol{\mu}_{G-1}\rangle, \tag{53}$$

using the construction method discussed in the previous subsection. Then, it suffices to construct the SELECT oracle without encoding the information of $\boldsymbol{\mu}$ to block-encode the entire Hamiltonian. Concretely, consider the following operation:

$$\mathtt{SELECT}=\sum_{\boldsymbol{\alpha},\boldsymbol{p}_1,\ldots,\boldsymbol{p}_G}|\boldsymbol{\alpha}\rangle\langle\boldsymbol{\alpha}|\otimes|\boldsymbol{p}_1,...,\boldsymbol{p}_G\rangle\langle\boldsymbol{p}_1,...,\boldsymbol{p}_G|\otimes H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}_1}H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}_2}\cdots H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}_G}. \tag{54}$$

Observe that this satisfies $\mathtt{SELECT}^2 = I$ and the following:

$$\langle\mathcal{L}|\mathtt{SELECT}|\mathcal{L}\rangle = \sum_{\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{p}} \frac{w_{\boldsymbol{\alpha},\boldsymbol{\mu}}}{\lambda} H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}} H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}+\boldsymbol{\mu}_2}\cdots H^{(\boldsymbol{\alpha})}_{\boldsymbol{p}+\boldsymbol{\mu}_{G-1}} = H/\lambda. \tag{55}$$

This allows us to block-encode the Hamiltonian with simpler quantum circuit structure. In fact, by observing that Eq. (54) can be written as tensor product as

$$\mathtt{SELECT} = \sum_{\boldsymbol{\alpha}} |\boldsymbol{\alpha}\rangle\langle\boldsymbol{\alpha}| \otimes \left[\left(\sum_{\boldsymbol{p}} |\boldsymbol{p}_1\rangle\langle\boldsymbol{p}_1| \otimes H^{\boldsymbol{\alpha}}_{\boldsymbol{p}_1}\right) \otimes \left(\sum_{\boldsymbol{p}} |\boldsymbol{p}_2\rangle\langle\boldsymbol{p}_2| \otimes H^{\boldsymbol{\alpha}}_{\boldsymbol{p}_2}\right) \otimes\cdots\otimes \left(\sum_{\boldsymbol{p}} |\boldsymbol{p}_G\rangle\langle\boldsymbol{p}_G| \otimes H^{\boldsymbol{\alpha}}_{\boldsymbol{p}_G}\right)\right], \tag{56}$$

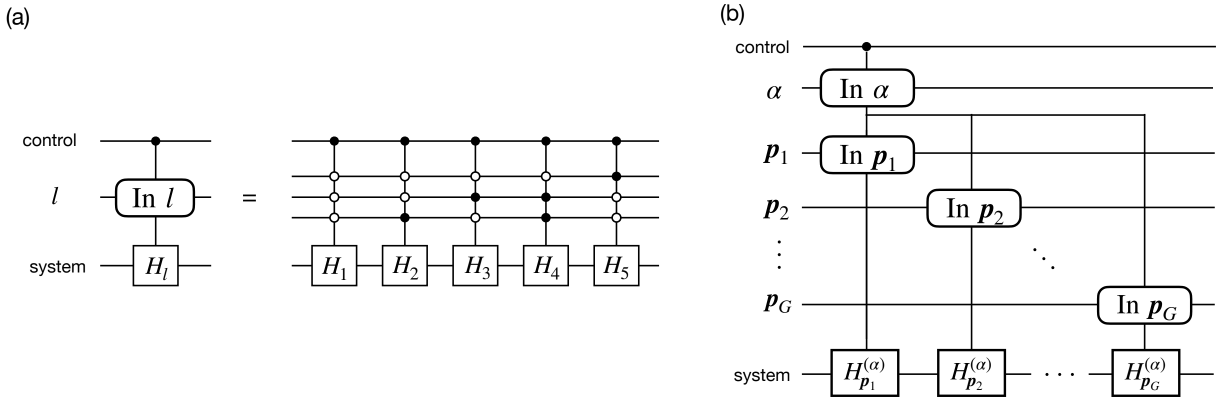
it is natural to employ the structure as shown in Supplementary Fig. 15(b). We find that each “thread” is used to block-encode $2SN_{\boldsymbol{\alpha}}N_{\text{site}}$ Pauli operators, respectively. Namely, we have $G \times 2SN_{\boldsymbol{\alpha}}N_{\text{site}}$ terms in total, and thus the T-count per SELECT oracle is $4\times G\times 2SN_{\boldsymbol{\alpha}}N_{\text{site}} - 4$ (see Supplementary Table VIII.) For concrete circuit structures of 2d J_1-J_2 Heisenberg model, 2d Fermi-Hubbard model, and spin-S Heisenberg chain, see Sec. V C 1, V C 2, and V C 3, respectively.

The connectivity-agnostic construction of SELECT oracle turns out to be beneficial in particular when we consider a densely coupled system due to, e.g., high dimensionality or long-range interaction. For instance, let us consider a spin-S Heisenberg model on d-dimensional hypercube with r-nearest-neighbor interaction, each site of which is coupled to $\gamma_{r,d}$ sites. Under such a procedure, the number of Pauli operator is given as

$$L = 3\times(2S)^2\times\frac{\gamma_{r,d}}{2}\times N_{\text{site}}. \tag{57}$$

Note the factor $2S$ due to encoding of high-spin operator as a sum of Pauli operator as $S_p^a = \frac{1}{2}\sum_{\nu=1}^{2S}\sigma^a_{p,\nu}$. Meanwhile, as we show in Supplementary Table VIII, the product-wise construction of SELECT is not affected by the connectivity of the interaction profile; this gives improvement for arbitrary interaction profile except for the case of spin-1/2 1d nearest-neighbor interaction. Furthermore, when we consider a quantum spin-S model with all-to-all G-local interactions, the reduction on the T-count is even more significant; each sequential-type SELECT oracle requires $O(S^G N^G)$ while the product-wise type can be implemented by $O(GSN)$, in exchange for additional $O(G\log N)$ ancilla qubits. Systematic discussion on such advantage is discussed elsewhere [57].

Let us remark briefly on the estimation of the actual runtime. While we may use T-counts as the baseline to compare the performance between different Hamiltonian simulation algorithms, we further discuss in Sec. VIII that the actual runtime (which is more related with the T-depth) can be further improved by distributed implementation of the controlled H_l operation.



Supplementary Figure 15. General quantum circuit structure of SELECT oracle. (a) Sequential structure based on the unary encoding of efficiently-implementable unitaries. Here we assume that the Hamiltonian is a sum over $L = 5$ Pauli terms. (b) Product-wise structure exploiting the G-local translational symmetric structure of Hamiltonian. Note that, instead of sharing the carry-on register for $\boldsymbol{\alpha}$, we may also consider G consecutive structure by uncomputing $\boldsymbol{\alpha}$ every time, which can be done without increasing the T-count. Meanwhile, in practice the proposed approach shall reduce the T-*depth*, since it allows us to perform the gates in parallel, as discussed in Sec. VIII.

`SELECT` type	General T-count	Heisenberg chain	2d J_1-J_2 Heisenberg	3d J_1-J_2-J_3 Heisenberg	General lattice Heisenberg (r, d, S)
Sequential	$4L-4$	$48S^2N_{\text{site}}-4$	$192S^2N_{\text{site}}-4$	$624S^2N_{\text{site}}-4$	$24S^2\gamma_{r,d}N_{\text{site}}-4$
Product-wise	$8SGN_{\boldsymbol{\alpha}}N_{\text{site}}-4$	$48SN_{\text{site}}-4$	$48SN_{\text{site}}-4$	$48SN_{\text{site}}-4$	$48SN_{\text{site}}-4$

Supplementary Table VIII. T-count of `SELECT` for spin-S Heisenberg model on d-dimensional hypercube with r-nearest-neighbor interactions. Here, we denote the number of edges per site as $\gamma_{r,d}$ and the total number of sites as N_{site}. To be concrete, Heisenberg chain satisfies $(r,d)=(1,1)$ with $\gamma_{r,d}=2$, 2d J_1-J_2 Heisenberg corresponds to $(r,d)=(2,2)$ with $\gamma_{r,d}=8$, where 3d J_1-J_2-J_3 Heisenberg model is described by $(r,d)=(3,3)$ with $\gamma_{r,d}=26$.

C. Explicit construction of oracles

We are now ready to dive into the explicit construction of quantum circuits for executing post-Trotter methods. Since it is straightforward to modify the circuits generated for qubitization into those for Taylorization, we exclusively discuss the oracles for qubitization.

1. 2d J_1-J_2 Heisenberg model

The Hamiltonian of spin-1/2 J_1-J_2 Heisenberg model on the square lattice is defined as in Eq. (2) as

$$H = J_1 \sum_{\langle p,q\rangle} \sum_{\alpha\in\{X,Y,Z\}} S_p^\alpha S_q^\alpha + J_2 \sum_{\langle\langle p,q\rangle\rangle} \sum_{\alpha\in\{X,Y,Z\}} S_p^\alpha S_q^\alpha, \tag{58}$$

where the summation of the first and second terms concerns nearest-neighbor and next-nearest-neighbor sites, respectively, and S_p^α is now the spin-1/2 operator. Here we exclusively consider a square lattice of $N = M\times M$ sites, and focus on the model with $J_1 = 1, J_2 = 0.5$ where quantum spin liquid phase is expected to show up.

First, let us briefly introduce the sequential construction of `UNIT-PREP` oracle. The circuit structure shown in Supplementary Fig. 16(a) realizes the following action:

$$\texttt{UNIT-PREP}\,|0\rangle \mapsto \sum_p \sum_{\alpha\in\{X,Y,Z\}} \sum_{\mu=1,2} \sqrt{\frac{J_1}{\lambda}}\,|p\rangle\,|\alpha\rangle\,|\mu\rangle + \sum_{\mu=3,4} \sqrt{\frac{J_2}{\lambda}}\,|p\rangle\,|\alpha\rangle\,|\mu\rangle\,, \tag{59}$$

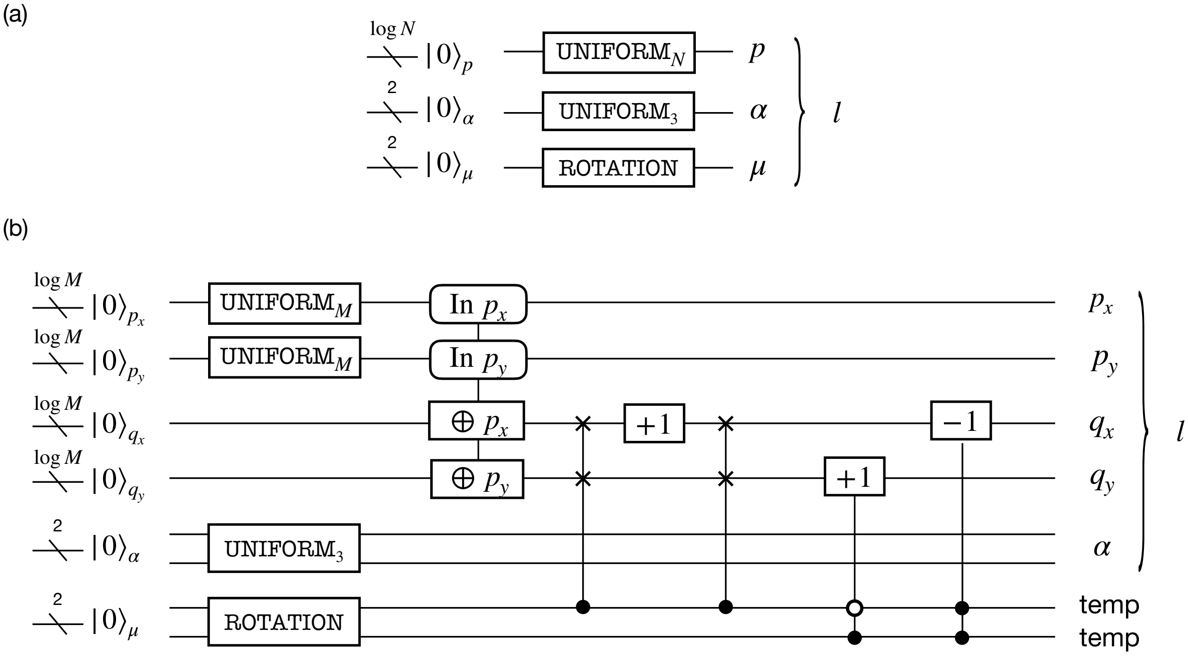
where $\mu\in\{1,2,3,4\}$ correspond to the coupling between (p_x, p_y) and $(p_x+1, p_y), (p_x, p_y+1), (p_x+1, p_y+1), (p_x-1, p_y+1)$, respectively.

On the other hand, for product-wise construction, `UNIT-PREP` circuit is designed to generate the following state:

$$\begin{aligned}\texttt{UNIT-PREP}\,|0\rangle \mapsto \sum_{p_x,p_y} \sum_{\alpha\in\{X,Y,Z\}} &\sqrt{\frac{J_1}{\lambda}}\,|p_x,p_y\rangle\,|p_x+1,p_y\rangle\,|\alpha\rangle \\ &+\sqrt{\frac{J_1}{\lambda}}\,|p_x,p_y\rangle\,|p_x,p_y+1\rangle\,|\alpha\rangle \\ &+\sqrt{\frac{J_2}{\lambda}}\,|p_x,p_y\rangle\,|p_x+1,p_y+1\rangle\,|\alpha\rangle \\ &+\sqrt{\frac{J_2}{\lambda}}\,|p_x,p_y\rangle\,|p_x-1,p_y+1\rangle\,|\alpha\rangle\,,\end{aligned} \tag{60}$$

whose graphical description is shown in Supplementary Fig. 16(b). Note that, the required number of ancillary qubits is $3\log N + O(1)$ rather than $2\log N + O(1)$ that is obtained by simply adding up the numbers in Supplementary Fig. 16(b). This is because the most efficient implementation of adders (in terms of T-count) require as many quantum registers as the number of qubits to be simulated [58]. Furthermore, by counting the number of required T-gates using the table presented in Sec. VI, we find that the `UNIT-PREP` can be implemented with T-count of $20\log N + 48\log(1/\delta_{\text{SS}})$.

As for the `SELECT` oracle, we follow the framework presented in Sec. V B for sequential construction. Meanwhile, if we use the product-wise `SELECT` oracle, the operation on computational bases that are not included in the signal state also becomes nontrivial, while this does not affect the dynamics encoded in the signal state. In this case, the T-count becomes $24N-4$, which is approximately halved compared to the vanilla sequential type of `SELECT` oracle.



Supplementary Figure 16. Circuit structure of `UNIT-PREP` for spin-1/2 J_1-J_2 Heisenberg model on square lattice. (a) The sequential construction requires T-count of $8\lceil\log N\rceil + 4\Gamma\log(1/\delta_{\rm SS})$ and $\log N + 4$ ancilla qubits, where (b) product-wise construction consumes T-count of $14\log N + 7\Gamma\log(1/\delta_{\rm SS})$ and spans $3\log N + 4$ ancillary qubits. Note that we have introduced $M^2 = N$ in the figure, and denoted the synthesis-dependent constant as Γ which are known to be $\Gamma \sim 1$ [59].

2. 2d Fermi-Hubbard model

The Hamiltonian of Fermi-Hubbard model on the square lattice reads as defined in Sec. I as

$$H = -t\sum_{\langle p,q\rangle,s}(c^\dagger_{p,s}c_{q,s} + \text{h.c.}) + U\sum_p n_{p,\uparrow}n_{p,\downarrow}, \tag{61}$$

where $c^{(\dagger)}_{p,\sigma}$ is the fermionic annihilation (creation) operator on site p with spin $\sigma \in \{\uparrow,\downarrow\}$, $n_{p,\sigma} = c^\dagger_{p,\sigma}c_{p,\sigma}$ is the corresponding number operator, t is the hopping amplitude, and U is the repulsive onsite potential. As in the case for spin-1/2 J_1-J_2 Heisenberg model, we mainly consider the lattice of $M\times M$ sites, and thus the total number of qubits required to simulate the system is $N = 2M^2$. While we take t to be unity in this work, we explicitly keep it among expressions when it is more informative.

For concreteness, we assume to employ the Jordan-Wigner transformation to encode fermionic degrees of freedom into quantum computer:

$$c^{(\dagger)}_{p,s} = \text{sgn}(s)\prod_{r<p}\prod_{s=\uparrow,\downarrow}\sigma^{+(-)}_{r,s} \tag{62}$$

where $p = p_x + p_y M$ and sgn(s) takes ±1 according to whether the spin is up or down. This leads us to the expression of Hamiltonian as follows,

$$H = -\frac{t}{2}\sum_{\langle p,q\rangle,s}(\sigma^X_{p,s}\sigma^{\vec{Z}}\sigma^X_{q,s} + \sigma^Y_{p,s}\sigma^{\vec{Z}}\sigma^Y_{q,s}) + \frac{U}{4}\sum_p \sigma^Z_{p,\uparrow}\sigma^Z_{p,\downarrow} - \frac{U}{4}\sum_{p,s}\sigma^Z_{p,s} + \frac{UN}{8}I, \tag{63}$$

where we have introduced $\sigma^{\vec{Z}}$ to represent the Z-string generated by Jordan-Wigner transformation. While one may use this definition, we may also employ an alternative definition with shift in the potential term as $\sum_p n_{p,\uparrow}n_{p,\downarrow} \to \sum_p n_{p,\uparrow}n_{p,\downarrow} - \frac{U}{2}\sum_{p,s}n_{p,s} + \frac{U}{4}I$, which results in the simplified form as

$$H = -\frac{t}{2}\sum_{\langle p,q\rangle}(\sigma^X_{p,s}\sigma^{\vec{Z}}\sigma^X_{q,s} + \sigma^Y_{p,s}\sigma^{\vec{Z}}\sigma^Y_{q,s}) + \frac{U}{4}\sum_p \sigma^Z_{p,\uparrow}\sigma^Z_{p,\downarrow}. \tag{64}$$

While this shift does not significantly change the T-count of oracles, it slightly modifies the L1 norm of the Hamiltonian which affects the total gate count, as discussed below.

In similar to the case with 2d J_1-J_2 Heisenberg model, we first introduce the sequential construction of `UNIT-PREP`. Our construction simply aims to generate

$$\begin{aligned}\texttt{UNIT-PREP}\,|0\rangle \mapsto \sum_{p_x,p_y}\Bigg[&\sqrt{\frac{U}{4\lambda}}\,|p,\uparrow\rangle\,|\text{onsite interaction}\rangle_{\alpha,\mu}\\ &+\sum_{s=\uparrow,\downarrow}\;\sum_{\mu=e_x,e_y,e_{-x},e_{-y}}\sqrt{\frac{t}{2\lambda}}\,|p,s\rangle\,|\text{hopping towards }\mu\rangle_{\alpha,\mu}\Bigg],\end{aligned}\tag{65}$$

which is realized by circuit shown in Supplementary Fig. 17(a). Note that the ancillary quantum states such as $|\text{onsite interaction}\rangle$ functions as "flag" to tell the `SELECT` oracle which interaction term must be operated. Here, the L1 norm is explicitly given as $\lambda = \lambda_t + \lambda_U$ where the contribution from the hopping term is $\lambda_t = t\cdot(\#\text{edges})$ and that from the interaction term is $\lambda_U = \frac{1}{8}UN$. We comment that this value is slightly different from $\lambda_U = \frac{1}{2}UN$ as employed in Ref. [52], which takes the constant term into account. However, the constant term can be eliminated from the QPE algorithm since we do not need to estimate its value. This results in a slightly smaller λ so that the repetition count of `SELECT` oracle is suppressed.

By carefully examining the coefficients and the commutation relation, we find out that `UNIT-PREP` presented in Supplementary Fig. 17(b) provides the the product-wise construction via the following operation:

$$\begin{aligned}\texttt{UNIT-PREP}\,|0\rangle \mapsto \sum_{p_x,p_y}\Bigg[&\sqrt{\frac{U}{4\lambda}}\,|p_x,p_y,\uparrow\rangle\,|p_x,p_y,\downarrow\rangle\,|1\rangle_\alpha\\ &+\sum_{s=\uparrow,\downarrow}\sqrt{\frac{t}{2\lambda}}\Big(|p_x,p_y,s\rangle\,|p_x+1,p_y,s\rangle\,|0\rangle_\alpha + |p_x,p_y,s\rangle\,|p_x,p_y+1,s\rangle\,|0\rangle_\alpha\Big)\\ &+\sum_{s=\uparrow,\downarrow}\sqrt{\frac{t}{2\lambda}}\Big(|p_x,p_y,s\rangle\,|p_x-1,p_y,s\rangle\,|0\rangle_\alpha + |p_x,p_y,s\rangle\,|p_x,p_y-1,s\rangle\,|0\rangle_\alpha\Big)\Bigg],\end{aligned}\tag{66}$$

where α specifies the type of interaction term.

Now we proceed to the operation of `SELECT`. We remark that the sequential construction is straightforward, and can be done by following the procedure explained as in Supplementary Fig. 15(a). In contrast, the product-wise construction of `SELECT` is rather nontrivial; given a signal state as presented in Eq. (66), the operation of `SELECT` shall satisfy

$$\texttt{SELECT}\,|p_x,p_y,s_{\mathbf{p}},q_x,q_y,s_{\mathbf{q}},\alpha\rangle\,|\psi\rangle \tag{67}$$

$$= |p_x,p_y,s_{\mathbf{p}},q_x,q_y,s_{\mathbf{q}},\alpha\rangle \otimes \begin{cases}\sigma^Z_{p,\uparrow}\sigma^Z_{q,\downarrow}\,|\psi\rangle & \text{if } p=q,\,(s_p,s_q)=(\uparrow,\downarrow),\alpha=1\\ \sigma^X_{p,s_p}(\prod_{p<r<q,s_r}\sigma^Z_{r,s_r})\sigma^X_{q,s_q}\,|\psi\rangle & \text{if } p<q,\,s_p=s_q,\,|\mathbf{p}-\mathbf{q}|=1,\alpha=0\\ \sigma^Y_{q,s_q}(\prod_{q<r<p,s_r}\sigma^Z_{r,s_r})\sigma^Y_{p,s_p}\,|\psi\rangle & \text{if } q<p,\,s_p=s_q,\,|\mathbf{p}-\mathbf{q}|=1,\alpha=0\end{cases} \tag{68}$$

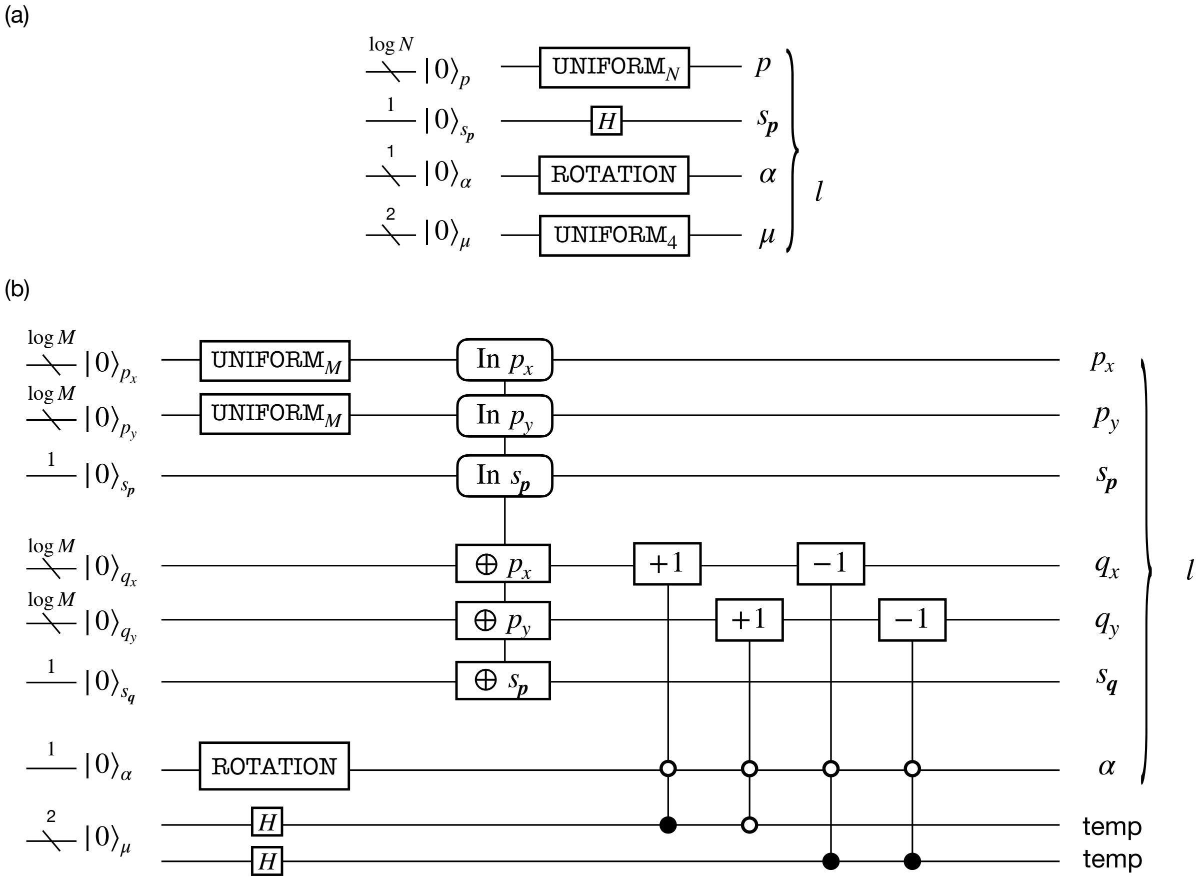
where the operation on subspace outside the signal state is unrestricted as long as $\texttt{SELECT}^2 = I$ is satisfied. Note that comparison between p and q is based on the integer $p = p_x + p_y M$ and $q = q_x + q_y M$ while $\|\mathbf{p}-\mathbf{q}\|$ denotes the Manhattan distance on the square lattice. Product-wise construction of `SELECT` oracle for the Fermi-Hubbard model has been suggested by Babbush et al., which pointed out that the construction can be done by structure shown in Supplementary Fig. 18 with T-count of $10N$ [52], as opposed to the one by the sequential type of $18N$. Note that, the T-count of sequential type without shifting the potential term is $22N$.

3. *Spin-1 antiferromagnetic Heisenberg chain*

Recall that the Hamiltonian of spin-1 antiferromagnetic Heisenberg chain reads

$$H = \sum_{p=1}^{N_{\text{site}}}\sum_{\alpha\in\{X,Y,Z\}} S^\alpha_p S^\alpha_{p+1}, \tag{69}$$

where S^α_p is the spin-1 operator acting on the p-th site and N_{site} denotes the number of sites in the system. To render the problem compatible with quantum computers that take qubits as a elementary unit of quantum registers, we



Supplementary Figure 17. Circuit structure of `UNIT-PREP` for 2d Fermion Hubbard model on square lattice. (a) Sequential construction requires T-count of $8\log N + 4\log(1/\delta_{\rm SS}) + O(1)$ with $\log N + O(1)$ ancilla qubits, while (b) product-wise construction consumes T-count of $16\log N + 6\Gamma\log(1/\delta_{SS}) + O(1)$ and $3\log N + O(1)$ ancilla qubits. The construction of `SHIFT_INDEX` is slightly more efficient than the one presented in Ref. [52]; the required number of T-gates is reduced from $32\log N$ to $16\log N$. However, the dominant factor is the `SELECT` circuit that consumes T-count of $O(N)$ and hence the difference is negligible in large-scale limit.

shall encode the system into enlarged Hilbert space spanned solely by qubits. In this regard, we represent each spin operator S_p^α as a sum of Pauli operators as $S_p^\alpha = \frac{1}{2}\sum_{\nu=1}^{2S}\sigma_{p,\nu}^\alpha$ and rewrite the Hamiltonian as

$$H = \frac{1}{4}\sum_{p=1}^{N_{\rm site}}\sum_{\nu,\tau=1}^{2S}\sum_{\alpha\in\{X,Y,Z\}}\sigma_{p,\nu}^\alpha\sigma_{p+1,\tau}^\alpha. \tag{70}$$

Note that we also assume that the trial state for quantum phase estimation is also represented using the qubit encoding.

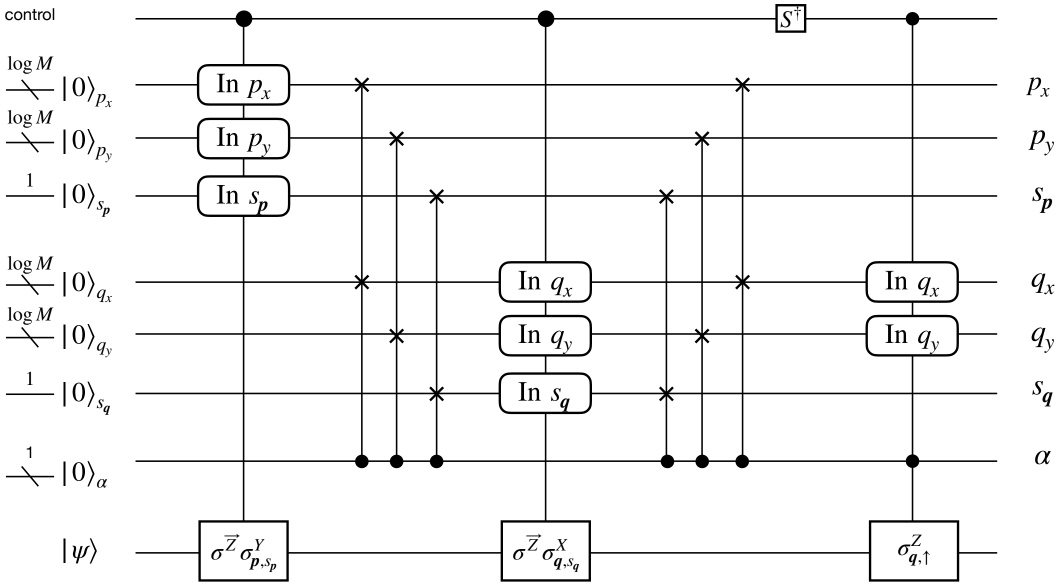
From the expression of Hamiltonian, we deduce that the `UNIT-PREP` circuit under periodic boundary condition can be composed as shown in Supplementary Fig. 19 to generate the following state:

$$\texttt{UNIT-PREP}\,|0\rangle^{\log N_{\rm site}+2} \mapsto \sum_{p,\alpha\in\{X,Y,Z\}}\sqrt{\frac{1}{\lambda}}\,|p\rangle\,|\alpha\rangle\,. \tag{71}$$

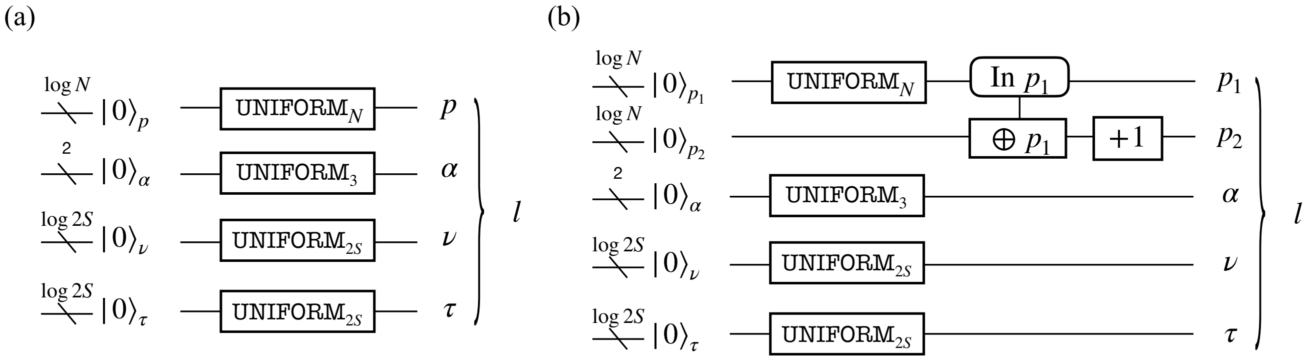
Correspondingly, the `SELECT` in the case of qubitization is constructed following the one presented in Supplementary Fig. 15(a) such that

$$\texttt{SELECT}\,|p,\alpha,\nu,\tau\rangle\,|\psi\rangle = |p,\alpha,\nu,\tau\rangle\,\sigma_{p,\nu}^\alpha\sigma_{p+1,\tau}^\alpha\,|\psi\rangle\,, \tag{72}$$

which can be realized by the general sequential implementation of `SELECT` by T-count of $48S^2N_{\rm site} - 4$. One may alternatively consider the product-wise implementation of `SELECT` as in Supplementary Fig. 15(b) by consuming T-count of $48SN_{\rm site} - 4$. However, we point out that there is no advantage in terms of T-count for $S = 1$ case, while



Supplementary Figure 18. Product-wise construction of SELECT for 2d Fermion Hubbard model on square lattice proposed in Ref. [52].



Supplementary Figure 19. Circuit structure of UNIT-PREP for spin-S Heisenberg chain. We display both (a) sequential construction that consumes T-counts of $8\log N + 8\Gamma\log(1/\delta_{\rm SS})$ and $\log N + 2\log S + O(1)$ ancillary qubits and (b) product-wise construction that consumes T-counts of $12\log N + 8\log(1/\delta_{\rm SS})$ and $3\log N + 2\log S + O(1)$ ancillary qubits. Note that it is redundant to include COPY_INDEX and SHIFT_INDEX for (b), since the site index is unambiguously related with the interaction terms.

it requires additional $\log N_{\rm site}$ ancillary qubits. Therefore, we perform our runtime estimation based on the former implementation.

VI. COMPLEXITY OF BASIC QUANTUM OPERATIONS

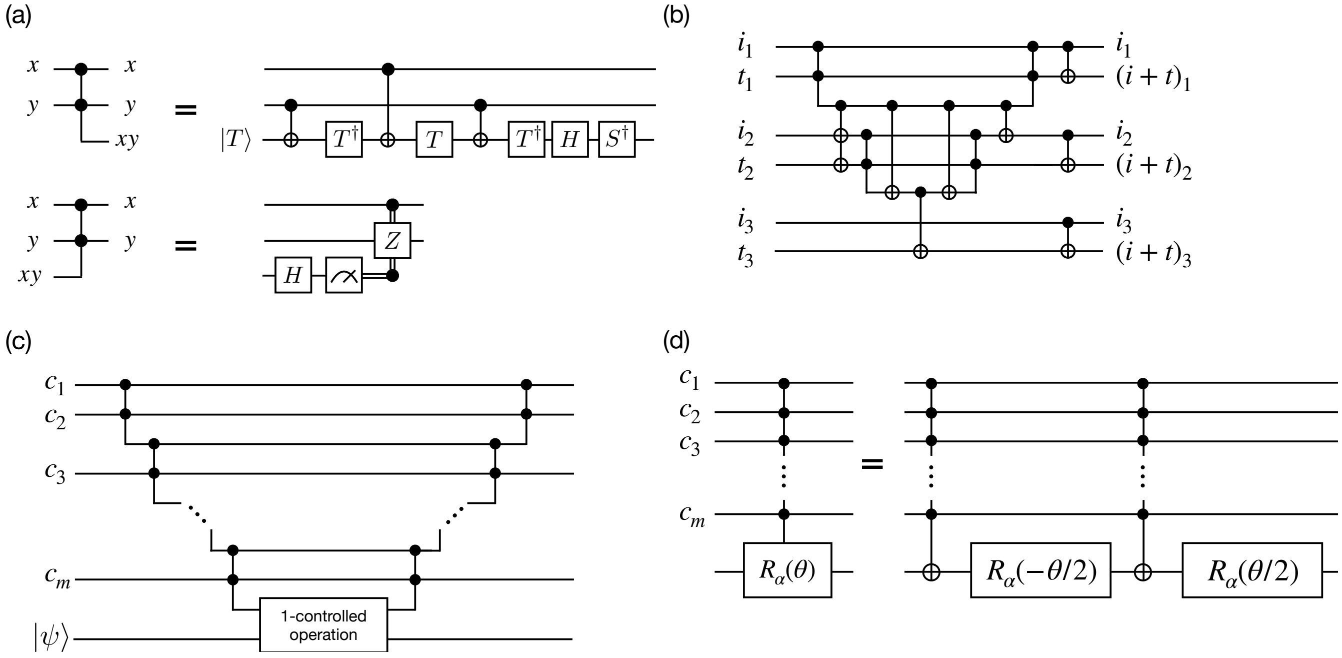
Here we provide the details on the basic quantum operations that are used for circuit construction and T-count analysis.

One of the most important operation that are commonly used in various basic gates presented in Supplementary Table IX is the logical AND operation, whose explicit Clifford+T construction is provided in Supplementary Fig. 20(a). Gidney showed that, by consuming ancillary qubits to store the temporal results of addition, we can implement n-qubit logical adder with T-count of $4n + O(1)$ and controlled adder with $8n + O(1)$ [58] (Supplementary Fig. 20(b)). We further push the use of temporary ancilla qubit to reduce the T-count for other controlled operations, such as m-controlled addition, SWAP, or rotation (See Supplementary Fig. 20(c), (d)). Also, we implement the UNIFORM operation following the construction scheme presented in Ref. [52]. Finally, as a variant of the adder, we introduce a

Operation	T-count
General Addition/Subtraction [58] (`SHIFT_INDEX`)	$4n-4$
Addition/Subtraction of $2^k L$	$4(n-k-1)-4$
Single-qubit rotation [59]	$\Gamma\lceil\log_2(1/\delta_{\mathrm{SS}})\rceil+\Xi$
$\mathtt{UNIFORM}_{2^k L}$ [52]	$8\log L+2\Gamma\log(1/\delta_{\mathrm{SS}})+2\Xi-4$
Arbitrary state synthesis [60]	$2^{n+1}-2$ arbitrary rotations

Controlled operation	T-count
m-controlled addition/subtraction	$4(m-1)+8(n-1)$
m-controlled addition/subtraction of $2^k L$	$4(m-1)+8(n-k-2)$
m-controlled NOT	$\begin{cases}0 & (m=0,1)\\ 4(m-1) & (m\geq 2)\end{cases}$
m-control SWAP	$\begin{cases}0 & (m=0)\\ 4m & (m\geq 1)\end{cases}$
m-controlled rotation	$8(m-1)+2\Gamma\log(1/\delta_{\mathrm{SS}})+2\Xi$
m-controlled $\mathtt{UNIFORM}_{2^k L}$	$4(m-1)+2k+10\log L+2\Gamma\log(1/\delta_{\mathrm{SS}})+2\Xi-4$

Supplementary Table IX. T-count required to perform basic operations that involves n qubits. Here, Γ and Ξ are constants whose mean values are numerically computed as 1.03 and 5.6, respectively [59], and δ_{SS} denotes the accuracy of rotation synthesis.

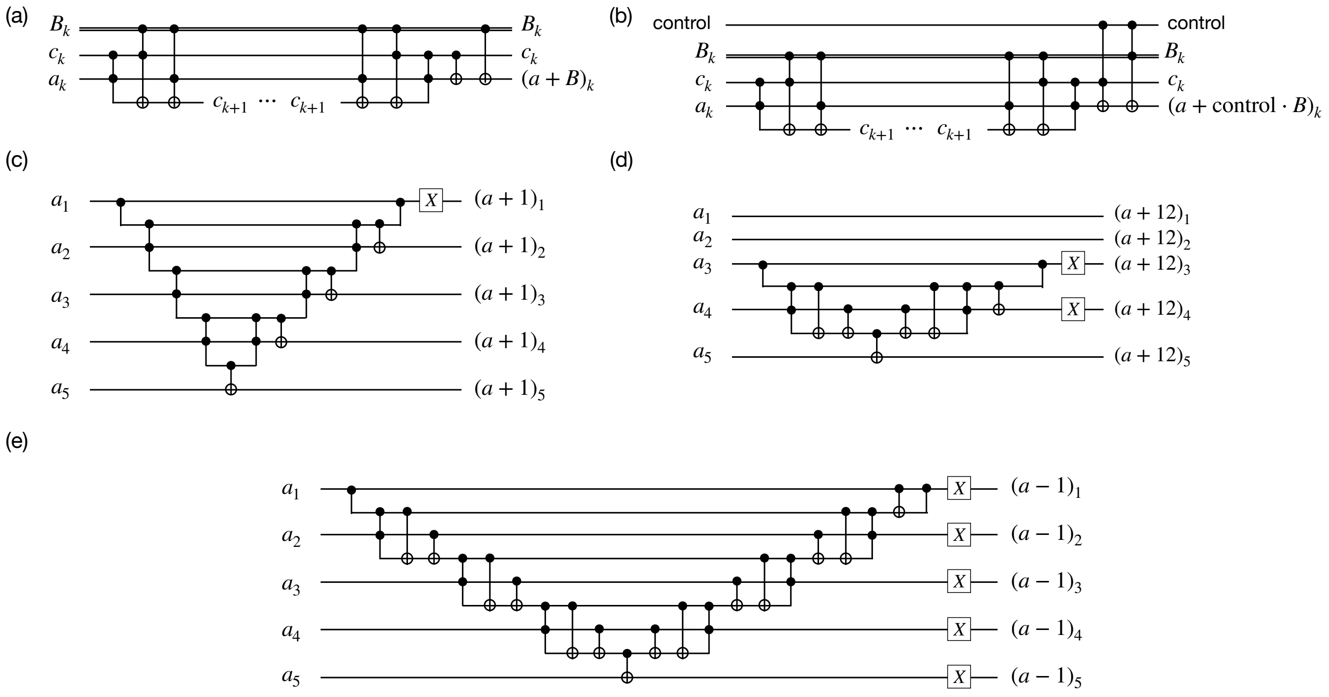


Supplementary Figure 20. Graphical representation of basic quantum operations. (a) Computation and uncomputation of logical AND operation. (b) 3-qubit logical adder. (c) m-controlled operation (d) m-controlled rotation expressed using m-controlled NOT and single-qubit rotation.

constant adder that performs some constant binary addition as shown in Supplementary Figs. 21.

VII. SZEGEDY QUANTUM WALK WITH IMPERFECT PREPARE

In Sec. IV D, we have seen that the Szegedy-type quantum walk operator enables us to explore the energy spectra of the block-encoded Hermitian operator. In particular, the oracular unitaries can be implemented efficiently if the Hamiltonian exhibit spatially periodic structure with $N_{\boldsymbol{\alpha}}=O(1)$ types of interaction coefficients. However, this



Supplementary Figure 21. Graphical representation of constant adder. Each panel represents (a) building block of constant adder, (b) building block of controlled constant adder, (c) 5-bit constant adder $a + 1 \pmod{2^5}$, (d) 5-bit constant adder of $a + 12 \pmod{2^5}$, (e) 5-bit constant subtracter of $a - 1 \pmod{2^5}$.

argument strongly depends on the boundary condition; when the model is defined with open boundary condition or cylindrical boundary condition, the interaction configurations differ at the boundary sites, which affects the complexity of the PREPARE oracle in particular. Although the T-gate complexity of PREPARE remains to be $\log(N/\epsilon)$ and therefore does not dominate the total cost even when we incorporate boundary effect, it is desirable to deal with minimal effort.

Here, we present a method that use homogeneous implementation of PREPARE and still extract simulation results for lattice systems defined under boundary conditions that are not necessarily periodic along all directions. To be concrete, we find that the error in the operation can be readily absorbed by constant shift in the Hamiltonian.

Let us first consider a perfect and also a slightly erroneous PREPARE operation that takes the ancillary system to the following states:

$$\texttt{PREPARE}_0 : |0\rangle^{\otimes a} \mapsto |\mathcal{L}_0\rangle = \sum_{l=1}^{L} \sqrt{\frac{w_l}{\lambda}} \, |l\rangle \,, \tag{73}$$

$$\texttt{PREPARE} : |0\rangle^{\otimes a} \mapsto |\mathcal{L}\rangle = \beta_1 \, |\mathcal{L}_0\rangle + \beta_2 \, |\perp\rangle \,, \tag{74}$$

where $|\perp\rangle$ is orthogonal to any $|l\rangle$. Our goal is to perform unbiased simulation of Szegedy quantum walk so that the obtained results correctly extracts the eigenspectra of the Hamiltonian. One naive solution is to "correct" the state from $|\mathcal{L}_0\rangle$ to $|\mathcal{L}\rangle$ via, e.g., amplitude amplification. Meanwhile, by making use of the specific structure of SELECT considered in this work, we can eliminate the error with negligible cost. To be specific, we assume that the SELECT operates nontrivially only in the subspace $\Pi\mathcal{H}\Pi$ where $\Pi = (\sum_{l=1}^{L} I^{\otimes n} \otimes |l\rangle \, \langle l|)$ and $\mathcal{H}$ is the total Hilbert space (confirm that this is satisfied in our construction presented in Sec. IV D.). Under such an assumption, we introduce a constant shift γ to the Hamiltonian to absorb the error. The overlaps are explicitly written as

$$(\langle \mathcal{L}_0| \otimes I)\texttt{SELECT}(|\mathcal{L}_0\rangle \otimes I) = \frac{H_0}{\lambda} + \gamma(-I), \tag{75}$$

$$(\langle \perp| \otimes I)\texttt{SELECT}(|\perp\rangle \otimes I) = I, \tag{76}$$

$$(\langle \perp| \otimes I)\texttt{SELECT}(|\mathcal{L}_0\rangle \otimes I) = 0, \tag{77}$$

which leads to

$$(\langle \mathcal{L}| \otimes I)\texttt{SELECT}(|\mathcal{L}\rangle \otimes I) = \frac{|\beta_1|^2 H_0}{\lambda} + (-\gamma|\beta_1|^2 + |\beta_2|^2)I. \tag{78}$$

Therefore, if we pick $\gamma = |\beta_2|^2/|\beta_1|^2$, we can simulate the rescaled version of Hamiltonian.

For concreteness, let us consider a case where we want to simulate a lattice system with cylindrical boundary condition using PREPARE that is intended for periodic boundary condition. We argue that the overhead due to the rescaling is negligible. This can be understood from the fact that the T-count for estimating the eigenenergy (at arbitrary precision) using the Taylorization or qubitization algorithm is proportional to λ. As can be seen from Eq. (78), the L1 norm of the coefficient is amplified under the erroneous signal state $|\mathcal{L}\rangle$ as $\lambda \to \lambda/|\beta_1|^2$. Since the ratio of coefficients is given as $|\beta_2/\beta_1|^2 = O((N^{\frac{d-1}{d}}/N)^2) = O(N^{-2/d})$ in d-dimensional lattice systems with local interactions, we straightforwardly obtain $|\beta_1|^2 \approx 1$, and thus does not affect the overall cost up to the leading order. Throughout this work, for T-count and runtime analysis we assume to employ PREPARE oracle designed to implement a signal state for periodic boundary condition.

VIII. RUNTIME ANALYSIS OF QUBITIZATION-BASED QUANTUM PHASE ESTIMATION

A. Overview

In this section, we explain the details of the runtime analysis for the quantum phase estimation that is based on the qubitization algorithm. As we have shortly mentioned in Sec. IV D, the runtime of quantum circuits represented with the Clifford+T formalism is typically evaluated with the number of T-gates, known as T-count, in the compiled quantum circuits. This is because each application of T-gates requires a time-consuming procedure consisting of magic-state injection, distillation, and teleportation in typical quantum error-correcting codes stabilized by Pauli operators. This is why we have compared the Hamiltonian simulation subroutines in Table 1 in the main text. While the T-count can capture the time-scaling of quantum algorithms when magic-state preparations are the most time-consuming factors, this estimation loses several vital factors in time analysis, and the actual execution time can vary a few orders of magnitudes depending on the strategy of compilation and executions. Here, we review several standard strategies and show how we calculated the execution times.

Throughout this paper, we assume the following situations: 1) Physical qubits are allocated on the vertices of 2d grids, which we call a qubit plane. 2) Logical qubits are encoded with surface codes [61–64]. 3) Logical operations are constructed based on twist-based lattice surgery [63, 65]. 4) The method introduced by Litinski [66] is applied to the construction of magic-state distillation, and purpose-specific strategies are employed for the compilation of SELECT and PREPARE oracles. Note that these assumptions may change or become unnecessary in the future by using devices with more flexible connectivity or employing other logical-operation strategies such as defect-pair braiding [64] and twist-free approach [67]. Nevertheless, we focus on the above scenario since they are standard assumptions in relevant studies on the runtime analysis [45, 68], and we leave the other scenario as future work.

B. Surface code and lattice surgery

In fault-tolerant quantum computing with surface codes, each logical qubit is allocated as a $d \times d$ block on the qubit plane, where d is the distance of surface codes. The error rates of logical qubits decrease exponentially with respect to d, as far as the error rates of physical qubits are smaller than the value called threshold. The required d is determined from physical error rates of given devices, logical-error suppression rate in error-correcting codes, and required logical error rates. During the computation, we repeat a set of Pauli measurements called stabilizer measurements, according to the generator set of a stabilizer group. A period for all the stabilizer Pauli measurements is called a code cycle. The recovery Pauli operations are estimated with a delay due to classical processing and stored in the unit logic called Pauli frame [63, 69].

All the basic logical operations must be achieved fault-tolerantly. The lattice surgery [65] is known to be the most promising strategy to efficiently achieve multi-qubit logical operations in fault-tolerant quantum computing. In this strategy, the logical Pauli measurements on multiple logical qubits are realized by connecting corresponding blocks with a proper circumference, and all the logical Clifford operations are performed via the logical measurements using lattice surgery and logical H and S gates. The time required for the multi-qubit logical Pauli measurements with the lattice surgery is independent of the number of the target logical qubits but linearly scales to the code distance d. Thus, it is convenient to define another unit, code beat, as the latency of d code cycles. We need 2 code beats for logical S and CNOT gates and 3 code beats for logical H gates. Thanks to the nature of CSS codes, the latency for destructive single-qubit Pauli-X, Z measurements and the preparation of the logical eigenstate of Pauli-X, Z are finished within a single code cycle. Since we can take their latency into account by increasing the code beat, we neglect their latency hereafter. Since lattice-surgery operations require ancillary logical-qubit cells that connect target logical qubits, the throughput of lattice-surgery operations depends on topological restrictions such as the allocation of logical qubits and routing of ancillary-qubit paths. See Ref. [63, 65, 68] for the detailed rules of the lattice surgery.

The non-Clifford gates, such as T and Toffoli gates, cannot be performed with the combinations of the Clifford operations, and are known to take a longer time compared with the other operations. These operations are achieved with the gate-teleportation technique that consumes quantum resources named the magic states. For instance, when one desires to operate a T-gate through its teleportation, we need a logical qubit prepared in a magic state $|A\rangle = 1/\sqrt{2}(|0\rangle + e^{i\pi/4}|1\rangle)$. While the noisy preparation of magic states can be achieved with the magic-state injection protocol, its infidelity is unacceptable for fault-tolerant operations. Therefore, we must create a magic state from a protocol called magic-state distillation, which allows the generation of magic states with arbitrarily small infidelity. A standard magic-state distillation protocol is 15-to-1 distillation. This protocol creates a single clean magic state from 15 noisy magic states to reduce the infidelity of magic states from p to $35p^3$, and succeeds with probability $1 - 15p$ [63]. The latency of a single trial of magic-state injection requires constant code cycles. Typically, we need two-level distillation for the required logical error rate, and each generation requires 15 code beats with the construction proposed in Ref. [66]. Note that large code distances are not required at the first distillation level since we only need to protect noisy magic states during distillation. Thus, the latency may be smaller. We typically assign a region of the qubit plane just for repeating magic-state preparation, which we call a magic-state factory. The teleportation of non-Clifford gates requires the feedback of a Clifford operation or the adaptive change of consequent Pauli measurement basis. Thus, successive gate teleportation will be blocked for a period of error-estimation delay, which is called a reaction time. The reaction time is determined by the throughput of the error-estimation algorithm for surface codes, physical error rates, chosen code distances, and its circuit implementation. It should be noted that, unlike Clifford operations, the most time-consuming part of non-Clifford gates is the magic-state preparation. This process can be performed in advance, and the preparation part can always be parallelized by assigning many magic-state factories. This makes basic optimization methods and bottleneck analysis for Clifford and non-Clifford gates different.

C. Time estimation without topological restriction

The runtime estimation of quantum algorithms is a key part of resource estimation and finding the crossover point of quantum advantage. In this section, we explain methods to estimate the runtime of quantum programs that are written in terms of Clifford+T formulation. As we detail below, the appropriate runtime estimation heavily relies on the hardware configuration and compilation settings. To estimate actual runtime with reasonable costs, there are two runtime-estimation strategies. One is heuristically estimating the runtime from several critical parameters such as T-count and depth, and the other is calculating the runtime with software simulation of fault-tolerant quantum computers. In the following, we introduce the first strategy in two different situations, the T-count limited and reaction limited estimations, as ones of the most commonly employed estimation strategies. We subsequently point out their defects and explain how to overcome them by software simulation while explicitly considering the topological restriction of logical qubits.

If quantum circuits consume magic states with a constant rate, we can roughly estimate the execution time by counting the number of T-gate. On the other hand, the consumption rate of magic states is not necessarily constant and the execution time may not be linear to the number of T-gates. Fortunately, there is a canonical form proposed by Litinski [66] that ensures an almost constant consumption rate of magic states. In this formalism, we convert the Clifford+T circuits to two stacks of layers; the first one is a sequence of $\pi/8$ Pauli rotations, and the latter is a sequence of Pauli measurements. The number of $\pi/8$ Pauli rotations is equal to the number of T-gates N_T. The execution of $\pi/8$ Pauli rotation consumes a single magic state, and is achieved with a single code beat and post-processing on the ancillary qubits. Based on Litinski's formalism, the estimation of the execution time with the Clifford+T formalism in a qubit-restricted situation is simple. Assuming that we have a small qubit plane and the number of the magic-state factory is very restricted, we can expect that the latter Pauli measurement part is negligible compared to the first $\pi/8$ rotation parts. Thus, the execution time can be estimated as the $\tau_T N_T$, where τ_T is the average time for generating a magic state and teleporting it. This is called a T-count limited estimation. In the most pessimistic scenario, we can only place a single factory on the qubit plane and the consumption rate of T-gate is much more frequent than the generation rate of a magic state factory. In this case, the execution time of quantum phase estimation can be obtained by multiplying the number of T-gates for `SELECT` and `PREPARE` oracles.

We can also consider the opposite scenario; we have a large qubit plane and can create arbitrary many magic-state factories. In this case, we can effectively have an on-demand magic-state supply. Thus, we can teleport multiple T-gates as far as they can be executed in parallel. Therefore, we assume that each simultaneous application is achieved with $\tau_{\text{group}} = \min(\tau_{\text{tel}}, \tau_{\text{react}})$, where τ_{tel} is the latency of lattice-surgery operations for the magic-state teleportation and τ_{react} is the reaction time. This quantity is directly related with the execution time as $D_T \tau_{\text{group}}$, where D_T

is the number of T-depth. When $\tau_{\mathrm{tel}} > \tau_{\mathrm{react}}$, we can hide the latency τ_{tel} by time-optimal construction [66, 70]. With ancillary qubits, we can convert the iterations of gate teleportation to parallel teleportation and a sequence of reactions. When we can utilize K times larger qubit space, we can force $\tau_{\mathrm{group}} \sim D_T \tau_{\mathrm{tel}}/K + D_T \tau_{\mathrm{react}}$. This situation is called a reaction-limited. An actual scenario would be the intermediate of the above two, and the actual execution time varies according to the size of the available qubit plane size.

In the above strategy, the T-count and depth of `SELECT` and `PREPARE` oracles are critical parameters. In the case of the ground-state energy estimation for condensed matter Hamiltonians, the coefficients of Hamiltonians are nearly uniform, and therefore the execution time is expected to be dominated by `SELECT` modules (See Sec. IV D). To implement `SELECT` operation with a small number of T-gates, we use a method proposed in Ref. [52]. The action of `SELECT` operations defined in Eq. (52) is divided into the product of small commutative unitary operations $\prod_{l=0}^{L-1} U_l$ where $U_l = |l\rangle \langle l| \otimes H_l + (I - |l\rangle \langle l|) \otimes I$. To construct U_l, i.e., a gate controlled by $\lceil \log L \rceil$-qubit input register $|l\rangle$, we use $(\lceil \log L \rceil - 1)$ ancillary qubits and map $|x\rangle |\psi\rangle |0\rangle$ to $|x\rangle |\psi\rangle |c_{x,l}\rangle$, where $c_{x,l} \in \{0,1\}^{(\lceil \log L \rceil - 1)}$ and the last bit of $c_{x,l}$ is 1 if $x = l$ and 0 if $x \neq l$. Then, we can achieve an action $|x\rangle |\psi\rangle |c_{x,l}\rangle \mapsto (U_l |x\rangle |\psi\rangle) |c_{x,l}\rangle = |x\rangle (H_l' |\psi\rangle) |c_{x,l}\rangle$ where $H_l' = H_l$ if $l = x$ and $H_l = I$ otherwise, using a Clifford operation $|0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes H_l$ where the control qubit corresponds to the last qubit of $|c_{x,l}\rangle$. To discuss the placement of logical qubits in the next subsection, we denote the register $|x\rangle$ and $|c_{x,l}\rangle$ as *control qubits*, and $|\psi\rangle$ as *system qubits*. Also, we denote the last qubit of $|c_{x,l}\rangle$ as *tail control qubit* since among the control qubits, only the tail control qubit interacts with system qubits in this iteration. The quantum circuit based on this iteration is shown in Supplementary Fig. 22 (a). At the first iteration, we construct $|x\rangle |0\rangle \mapsto |x\rangle |c_{x,0}\rangle$, of which the action is shown as `Initialize`. In the i-th iteration, we need to update the state as $|x\rangle |c_{x,i}\rangle \mapsto |x\rangle |c_{x,i+1}\rangle$, which is achieved by a unitary operation named $U_{\mathrm{inc},i}$. Finally, the ancillary qubits are disentangled as $|x\rangle |c_{x,L-1}\rangle \mapsto |x\rangle |0\rangle$ with `Finalize`. This method enables the implementation of `SELECT` operations with T-count of $4L - 4$.

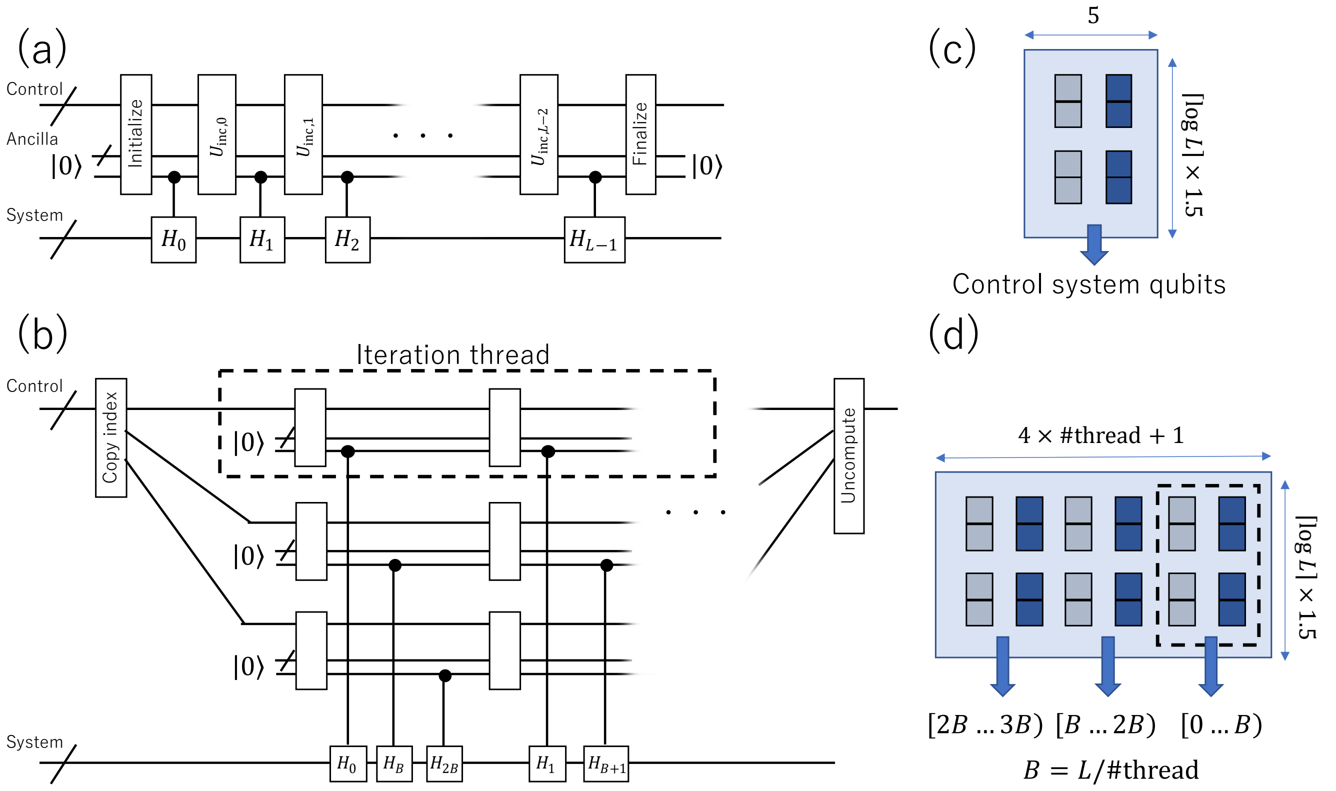
The above construction of `SELECT` operation is a serial sequence of Toffoli gates for `Initialize`, $U_{\mathrm{inc},i}$, and `Finalize`. Thus, its T-depth is almost equal to T-count. This means there is no room to reduce T-depth even when several magic-state factories are available. To enable a flexible performance tuning of `SELECT` operations according to available magic-state supply rates, we propose a technique to reduce the T-depth with a similar idea of loop-parallelization in classical computing (See Supplementary Fig. 22 (b)). Suppose that the target Hamiltonian consists of L terms of Pauli operators acting on n qubits. In this technique, we divide the task of `SELECT` operations to b parallelized executions using b times the number of control qubits as follows. First, we clone the input control registers to GHZ states $\sum_x \alpha_x |x\rangle \mapsto \sum_x \alpha_x |x\rangle^{\otimes b}$, which is denoted as `CopyIndex` operation, and we call each cloned control register as a *thread*. Then, we divide the L terms to b blocks, where the i-th block contains terms indexed from (iL/b) to $((i+1)L/b - 1)$, and `SELECT` operations is performed as b independent iterations by b threads. Finally, we uncompute the cloned registers. This construction effectively reduces the T-depth from $O(L)$ to $O(L/b + \log L)$ with an additional $O(b \log n)$ qubits. The brief construction of this parallelization is shown in Supplementary Fig. 22 (b). Note that as far as we keep b sufficiently small than n, the overhead of $O(b \log n)$ ancillary qubits would be negligible compared to the system size n. The detailed decomposition and parallelization efficiency will be presented in another paper.

Although the T-count limited or reaction limited estimation based on the above counting provides a rough estimation of runtime, it should be noted that they ignore the overhead by topological restriction of the qubit plane. Since each logical qubit is placed at a certain position on the qubit plane and we need to connect the target blocks for lattice surgery, we sometimes cannot execute operations due to the conflict of their paths. This means T-gates are not necessarily teleported in parallel even if they are located in parallel on the quantum circuit representation. To take them into account, we need to consider the allocation of logical qubits and their routing for lattice-surgery operations, which is discussed in the next section.

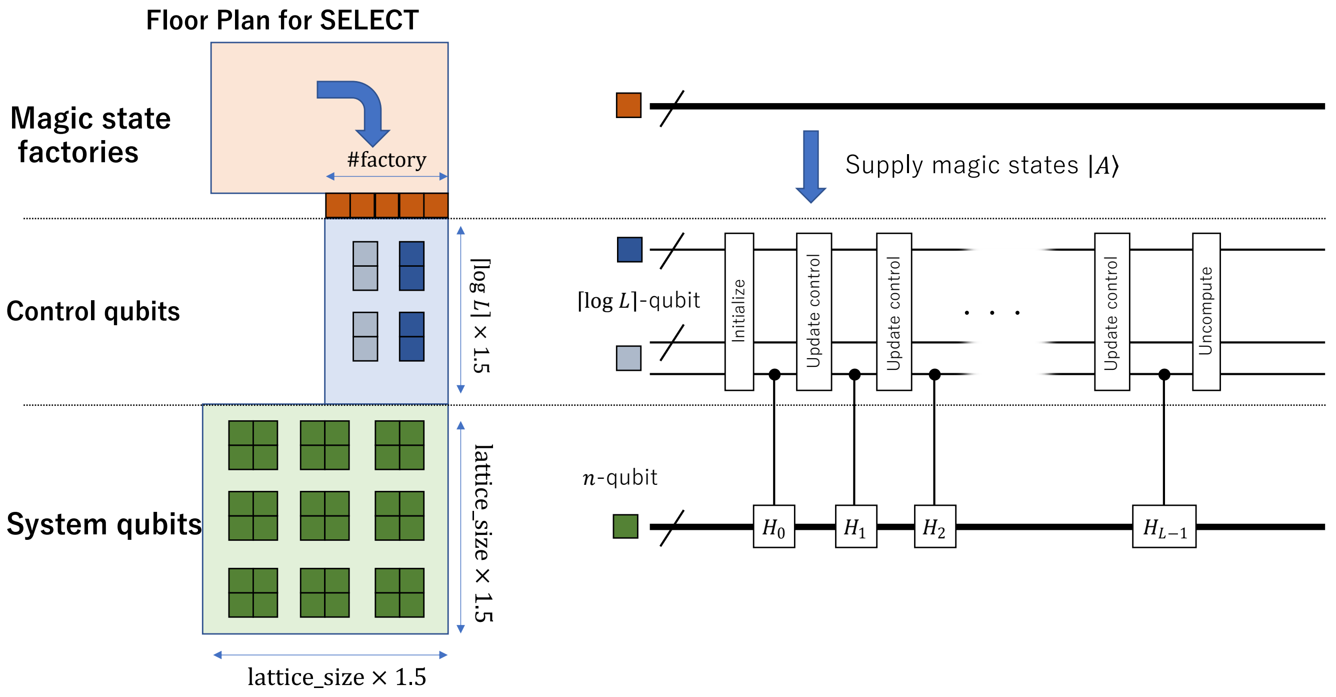
D. Placement of logical qubits and routing for lattice surgery

Once quantum circuits are translated to the sequence of intrinsic lattice-surgery operations, we need to place each logical qubit to a certain block on the qubit plane. While we should choose dense allocations to avoid the redundant usage of the qubit plane, too dense allocations may lead to path conflicts and the degradation of instruction throughput. In this paper, we heuristically designed the placement of logical qubits for condensed matter Hamiltonians for 2d spins by leveraging the feature of spatially local interactions. Our logical qubits mapping and correspondence to `SELECT` circuits is illustrated in Supplementary Fig. 23.

In our design, the qubit plane is divided into three parts; factory, control, and system. We assume that the design of magic state factories proposed by Litinski [66] is used and they are simply located in the factory parts. So, we explain the detailed logical-qubit placement of the system and control parts. The system part is used for locating system qubits in `SELECT` operation. For the 2d Heisenberg models, the position of system qubits representing a spin at a certain coordinate is placed at the same coordinate of the system part to leverage the locality. For the 2d



Supplementary Figure 22. (a,b) Quantum circuits for SELECT operations with sequential and parallelized constructions. (c,d) Qubit placement of SELECT operations with sequential and parallelized constructions.



Supplementary Figure 23. The floor plan of the qubit plane for the sequential construction of the SELECT operation.

Fermi-Hubbard models, we heuristically determined the positions of system qubits according to those of fermions. Since Clifford operations and lattice surgery need ancillary space around the logical-qubit cells, we created $O(\sqrt{n})$ corridors of ancillary space so that the vertical and horizontal edges of each logical-qubit cell touch them. Note that this is one of the most compact allocations while keeping Clifford operations available [63, 67]. We use this compact allocation for the system part because of the observation that system qubits are less frequently accessed compared to the control part, and the compact allocation of the system part reduces the required number of physical qubits without losing the parallelism of `SELECT` operations.

The control part is used for locating control qubits in `SELECT` operations, i.e., input register $|x\rangle$ and ancillary register $|c_{x,l}\rangle$. Since the operations on this part are dominated by Toffoli-gates acting on the neighboring bits in the registers, we placed logical qubits for them as arrays in the order of head bits to tail bits. An example allocation of the control part is shown in Supplementary Fig. 22(c), where an array of dark cells corresponds to the input register and that of light cells to the ancillary register.

After we determine the placement for each part, we need to map the location of the three parts on the qubit plane. As shown in Supplementary Fig. 23, we allocate three parts in the order of factory, control, and system parts. The factory part is placed behind the control part since control qubits consume magic states while system qubits do not. Also, since only the tail control qubits in the control part interact with the system part, we let the system part touch the side of the control part where the tail control qubits are located.

When we utilize the parallelization to the control part, we modify the qubit allocation of the control part as shown in Supplementary Fig. 22(d). If the Pauli terms are randomly assigned to each thread, tail control qubits of the threads try to touch system qubits at random positions, which induces the conflicts of Clifford operations and may become a bottleneck. This problem can be resolved by re-ordering the Pauli terms and appropriate block allocations. Since we now consider nearly uniform coefficients, the cost of `PREPARE` oracle is almost independent of the ordering of coefficients. Thus, we can change the order of Pauli terms to minimize the execution time for `SELECT` operations. Suppose the list of sorted indices in the i-th Pauli operator is $(j_1, ...j_k)$. Then, we sort the Pauli terms with the set in ascending order, separate them into b blocks, and assign them to each thread. When the thread number b satisfies $\sqrt{n} > b$, $\sqrt{n}/b + k$ corridors are relevant to each thread. With this assignment, there are no potential conflicts as far as $k < \sqrt{n}/b$. Thus, we can expect this assignment will prevent the potential conflict of lattice-surgery paths.

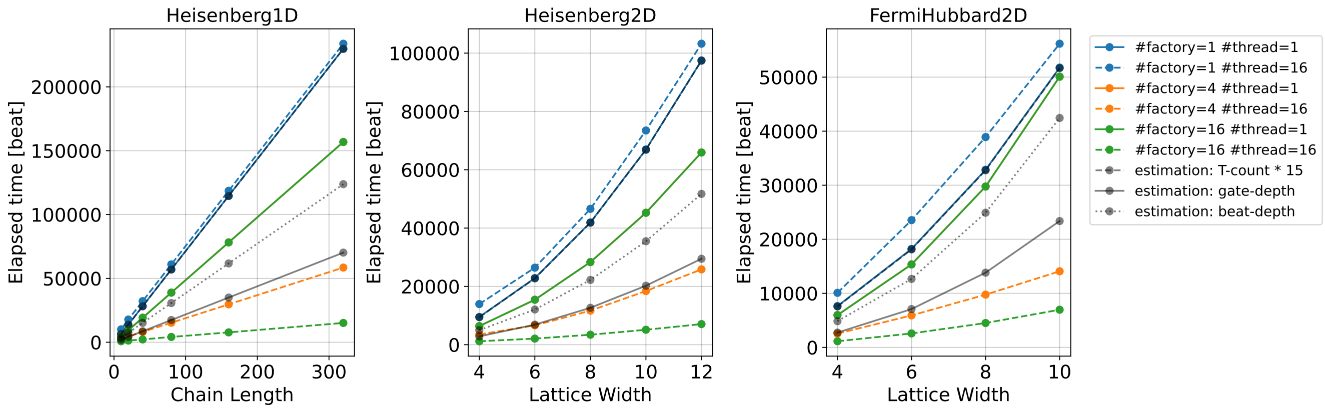
Finally, we schedule the order of instructions and determine ancillary cells for logical operations. We translate Clifford+T formalism into a 1D sequence of S, H, magic-state generation, one-body Pauli measurement, and two-body lattice-surgery operations. Each instruction becomes ready for execution when all the instructions that act on the target logical qubits earlier than the target instruction. Several instructions, such as feedback operations of T-gate teleportation, need to wait for relevant measurement outcomes to be error-corrected, i.e., there is a latency of reaction time. For logical S and H gates, logical qubits consume one of the neighboring and available ancillary cells. If there is no available ancillary cell, instructions wait for the next code beat. For lattice-surgery instruction, we find one of the shortest paths between target logical qubits with the breadth-first search algorithm and consume the ancillary cells on the path. If there is no path between logical qubits, instructions wait for the next code beat.

E. Parameter estimation and numerical evaluation

Here, we estimate the relevant parameters according to the recent device technologies. We define d, L, and m as a code distance, the number of terms in the target Hamiltonian, and the required digit of the ground energy, respectively. They are left as variables since they depend on the problem size. While a shorter code cycle is preferable to shorten the execution time, it is lower-bounded by the time for stabilizer measurements and error estimation algorithms. Currently, the latter is the dominant factor and is expected to be no shorter than 1 μs [71]. Thus, the code beat, which is defined by d code cycles, is d μs. The time for two-level magic-state distillation τ_T is $15d$ μs. The reaction time is typically assumed to be 10 μs. The T-count for the `SELECT` operation is $4L$ and the T-depth is $4L/b$, where b is a parallelization level.

Summarizing the above, we can estimate the total execution time using several variables depending on the problem. In the case of T-count limited scenario, the execution time for `SELECT` operation is about $15d$ [μs] $\times\ 4L = 60dL$ [μs]. In the case of T-depth limited case, the execution time for `SELECT` operation is 10 [μs] $\times\ 4L/b = 40L/b$ [μs], which is bd times smaller than the T-count limited case. Assuming the other parts, such as `PREPARE` , initial-state preparation, and quantum Fourier transform, are negligible compared to the `SELECT`, the whole execution time can be estimated by multiplying 2^m (or number of repetitions determined as $r = \pi\lambda/2\epsilon$ as discussed in Sec. IV D) to that for `SELECT`.

Finally, we verified our analysis with numerical calculation. We have synthesized the quantum circuits of `SELECT` operations with and without parallelization, allocated logical qubits as shown in the floor plan, determined the routing of lattice surgery with depth-first search algorithms, and calculated the required code beat for `SELECT` circuits with simulation. The observed coded beats to finish `SELECT` operations are plotted as a function of problem sizes and the



Supplementary Figure 24. The required code beats for executing `SELECT` operations for several Hamlitonian models.

type of Hamiltonians in Supplementary Fig.24. In the numerical simulation, we used the following approximations to simplify the evaluation:

- We assumed each magic-state factory generates a single magic state per 15 code beat. Note that this value may be improved by using more efficient magic-state distillation strategy, such as the construction proposed in Ref. [63, 66].
- When Clifford operations are conditioned by previous measurement outcomes, the operations are error-corrected and become available after 1 code beat.
- While logical S gates are required after T-gate teleportation with only probability of 0.5, we pessimistically assume that the logical S gate is always required.
- Due to the implementation of `PREPARE`, L Hamiltonian terms are not necessarily assigned to the computational basis from $|0\rangle$ to $|L-1\rangle$ in the control qubits. Nevertheless, we assumed it is enough to iterate from 0-th to $L-1$-th for simplicity. The conciliation between the cost of `SELECT` and `PREPARE` is future work.

According to the numerical results, when we do not perform parallelization and only have a single factory (solid blue line), the runtime is close to the product of the rate of magic-state generation and T-count (broken black line). When the number of factories increases to 4, the runtime saturates at about half of T-count limited estimation (solid green line). Compared to the depth of circuit (black dotted line), it is about 1.2 times larger. We expect that this discrepancy comes from the latency of reactions or path conflicts. When we employ parallelization, the runtime can be reduced to a smaller value while it requires a larger number of qubits and factories. With four threads and 16 factories, the runtime can be suppressed to 0.1 times smaller values. Thus, we can conclude that when there are several magic-state factories, the parallelization is effective in further reducing the computational time with a small overhead of required logical qubits. The required code beats for all the configurations are summarized in Table. X.

IX. ESTIMATION OF THE NUMBER OF PHYSICAL QUBITS, CODE DISTANCES, AND ACTUAL RUNTIME

Here, we explain how to estimate required number of physical qubits N_{ph} via determining the code distance d of the surface code. In Sec. IX A, we provide rough estimation on the physical qubit count to compare the space complexity between quantum algorithms used for the phase estimation. In Sec. IX B, we perform estimation that explicitly takes the mapping and connectivity of logical qubits into account, so that we can combine all the results and estimate the actual runtime required for the qubitization-based phase estimation algorithm.

A. Rough comparison between quantum algorithms

First we describe how to obtain a rough estimate on the physical qubit count which yields Supplementary Fig. 5 in the main text. We choose the code distance so that the logical error rate $p_{\log}$ is smaller than the number of logical

Supplementary Table X. The calculated runtimes for a single SELECT operations for several Hamiltonian models. The execution times are shown with the unit of code beats, and n_F means the number of magic-state factories. The runtime in terms of expected walltime is shown in Supplementary Table XII.

Hamiltonian	lattice size	#thread	runtime [beat]	runtime [beat]	runtime [beat]
			($n_F = 1$)	($n_F = 4$)	($n_F = 16$)
2d J_1-J_2 Heisenberg	4×4	1	9511	6389	6374
2d J_1-J_2 Heisenberg	4×4	16	13939	3514	1185
2d J_1-J_2 Heisenberg	6×6	1	22840	15433	15412
2d J_1-J_2 Heisenberg	6×6	16	26419	6634	2091
2d J_1-J_2 Heisenberg	8×8	1	41929	28348	28327
2d J_1-J_2 Heisenberg	8×8	16	46583	11678	3424
2d J_1-J_2 Heisenberg	10×10	1	66958	45261	45239
2d J_1-J_2 Heisenberg	10×10	16	73465	18400	5127
2d J_1-J_2 Heisenberg	12×12	1	97498	65939	65917
2d J_1-J_2 Heisenberg	12×12	16	103225	25840	7065
spin-1 Heisenberg chain	10	1	6559	4441	4407
spin-1 Heisenberg chain	10	16	10093	2547	780
spin-1 Heisenberg chain	20	1	13828	9391	9355
spin-1 Heisenberg chain	20	16	17779	4472	1308
spin-1 Heisenberg chain	40	1	28237	19227	19185
spin-1 Heisenberg chain	40	16	32179	8072	2172
spin-1 Heisenberg chain	80	1	57046	38887	38845
spin-1 Heisenberg chain	80	16	60983	15278	4157
spin-1 Heisenberg chain	160	1	114655	78208	78165
spin-1 Heisenberg chain	160	16	118585	29680	7728
spin-1 Heisenberg chain	320	1	229864	156848	156805
spin-1 Heisenberg chain	320	16	233789	58484	15116
2d Fermi-Hubbard	4×4	1	7637	5981	5969
2d Fermi-Hubbard	4×4	16	10093	2549	1134
2d Fermi-Hubbard	6×6	1	18215	15362	15341
2d Fermi-Hubbard	6×6	16	23539	5914	2579
2d Fermi-Hubbard	8×8	1	32864	29772	29751
2d Fermi-Hubbard	8×8	16	38903	9759	4507
2d Fermi-Hubbard	10×10	1	51764	50097	50076
2d Fermi-Hubbard	10×10	16	56183	14079	6966

operations $N_{\rm op}$:

$$p_{\log} < 1/N_{\rm op}, \tag{79}$$

where we estimate $p_{\log} = 0.1(p/p_{\rm th})^{(d+1)/2}$ with the ratio of physical error rate over error threshold $p/p_{\rm th} = 0.1$. Note that this choice has been commonly employed in existing runtime estimation literature such as Refs. [45, 66, 72]. In order to obtain rough estimate on the number of logical operations, we focus solely on the T gates. Based on the T-count that was also presented as Table I. in the main text, we calculate $N_{\rm op}$ as follows,

$$N_{\rm op} = N_{\log} \times (T\text{-count}) \times (\text{code beat per } T\text{-gate}), \tag{80}$$

where $N_{\log}$ is the number of logical qubits used for the computation, i.e., the number of system qubits and the ancilla. Note that in the next section we perform more detailed analysis with a slightly different definition of $N_{\rm ph}$. The code beat for each T-gate is assumed to be 15 [66]. By solving Eq. (79), we can determine the code distance d. Assuming that all the logical qubits are encoded with the framework of surface code with homogeneous code distance, the number of physical qubits can be obtained as

$$N_{\rm ph} = 2d^2 N_{\log}. \tag{81}$$

	($N_{\rm log}$, d, $N_{\rm ph}$) for 2d J_1-J_2 Heisenberg				
System size	qDRIFT	random Trotter	Taylorization	Qubitization (sequential)	Qubitization (product)
4×4	16, 29, 2.69e+04	16, 23, **1.69e+04**	96, 25, 1.20e+05	24, 21, 2.12e+04	38, 21, 3.35e+04
6×6	36, 31, 6.92e+04	36, 25, **4.50e+04**	146, 27, 2.13e+05	46, 23, 4.87e+04	64, 23, 6.77e+04
8×8	64, 33, 1.39e+05	64, 27, 9.33e+04	174, 27, 2.54e+05	74, 23, **7.83e+04**	92, 23, 9.73e+04
10×10	100, 35, 2.45e+05	100, 29, 1.68e+05	232, 29, 3.90e+05	112, 25, **1.40e+05**	134, 25, 1.68e+05
12×12	144, 35, 3.53e+05	144, 29, 2.42e+05	288, 29, 4.84e+05	156, 25, **1.95e+05**	178, 25, 2.22e+05
14×14	196, 37, 5.37e+05	196, 31, 3.77e+05	340, 31, 6.53e+05	208, 27, **3.03e+05**	230, 27, 3.35e+05
16×16	256, 37, 7.01e+05	256, 31, 4.92e+05	400, 31, 7.69e+05	268, 27, **3.91e+05**	290, 27, 4.23e+05
20×20	400, 39, 1.22e+06	400, 31, 7.69e+05	568, 31, 1.09e+06	414, 29, **6.96e+05**	440, 29, 7.40e+05

	($N_{\rm log}$, d, $N_{\rm ph}$) for 2d Fermi-Hubbard				
System size	qDRIFT	random Trotter	Taylorization	Qubitization (sequential)	Qubitization (product)
4×4	32, 29, 5.38e+04	32, 25, 4.00e+04	131, 25, 1.64e+05	41, 21, **3.62e+04**	49, 21, 4.32e+04
6×6	72, 31, 1.38e+05	72, 27, 1.05e+05	193, 27, 2.81e+05	83, 23, **8.78e+04**	95, 23, 1.01e+05
8×8	128, 33, 2.79e+05	128, 29, 2.15e+05	249, 27, 3.63e+05	139, 25, 1.74e+05	151, 23, **1.60e+05**
10×10	200, 35, 4.90e+05	200, 31, 3.84e+05	356, 29, 5.99e+05	213, 25, **2.66e+05**	229, 25, 2.86e+05
12×12	288, 35, 7.06e+05	288, 31, 5.54e+05	444, 29, 7.47e+05	301, 27, 4.39e+05	317, 25, **3.96e+05**
14×14	392, 37, 1.07e+06	392, 31, 7.53e+05	548, 31, 1.05e+06	405, 27, **5.90e+05**	421, 27, 6.14e+05
16×16	512, 37, 1.40e+06	512, 33, 1.12e+06	668, 31, 1.28e+06	525, 27, **7.65e+05**	541, 27, 7.89e+05
20×20	800, 39, 2.43e+06	800, 33, 1.74e+06	980, 33, 2.13e+06	815, 29, **1.37e+06**	835, 29, 1.40e+06

Supplementary Table XI. Estimation on quantum resource to execute the phase estimation algorithm under various quantum subroutines. Here we show $N_{\rm log}$, d, and $N_{\rm ph}$, which indicate the logical qubit count, code distance for the surface code, and the physical qubit count given by $N_{\rm ph} = 2d^2 N_{\rm log}$. Note that, as opposed to the estimation in Sec. IX B, here we exclude the logical qubits used for the magic state factory, each of which consumes 176 logical qubits in assuming standard level-2 distillation.

As shown in Supplementary Table XI, although the post-Trotter methods requires ancilla qubits and hence consume larger $N_{\rm log}$, the significant reduction in the T-count in the qubitization algorithm results in fewer $N_{\rm ph}$ compared to Trotter-based methods. See also Supplementary Fig. 5 in the main text for visualization of the spacetime cost for the lattice size of 10×10.

B. Detailed estimation with actual runtime

Next we proceed to describe how to perform more detailed estimation on $N_{\rm ph}$ together with the actual runtime of the algorithm. We assume that the qubits are allocated as depicted in the floor plan of Supplementary Fig. 23 which consists of system qubits, control qubits, and magic state factory. By assuming that all logical qubits are encoded by framework of surface code with homogeneous code distance of d, the number of physical qubits $N_{\rm ph}$ is provided as

$$N_{\rm ph} = \left(1.5^2 N_S + 1.5(4b+1)\log L + n_F A_F\right) \times (2d^2), \tag{82}$$

where N_S is number of system qubits, b is the number of parallelization threads, n_F is the number of magic state factory, and A_F is the floor area per single magic state factory. Here, we follow the construction of Ref. [66] and employ $A_F = 176$. Meanwhile, as was also noted in Sec. VIII B, we may reduce the code distance of logical qubits that are used for the first-level magic state distillation (e.g. two-fold), which results in significant reduction of physical qubits especially when number of magic state factories n_F is large.

In order to determine the code distance, now we estimate from the repetition count r of qubitization oracle by neglecting the contribution from the subdominant `PREPARE` oracle as

$$N_{\rm op} = (\#\text{logical qubits involved in } \texttt{SELECT}) \times (\#\text{code cycle per } \texttt{SELECT}) \times r, \tag{83}$$

where r is related with the total energy error ϵ via $\delta_{\rm PEA} = 0.9\epsilon/\lambda$ as $r = \lceil \pi/(2\delta_{\rm PEA}) \rceil$. By solving Eq. (79) using these quantities, we find that it is sufficient to take d around 21 or 25 around the quantum-classical crosspoint.

The total runtime can be estimated as

$$t \sim t_{\mathrm{beat}} \times N_{\mathrm{beat}} \times r, \tag{84}$$

where $t_{\mathrm{beat}} = t_{\mathrm{cycle}} \times d$ is the required time per code beat that is determined from the time per code cycle of surface code $t_{\mathrm{cycle}} = 1\mu s$, and N_{beat} is the number of code beat per `SELECT` oracle obtained in Sec. VIII. The overall runtime estimation results and quantum resource are summarized in Supplementary Table XII, which is displayed in comparison with classical algorithms in Supplementary Fig. 3 in the main text. Here, we clearly observe that quantum advantage is achieved in runtime of hours, with use of physical qubits of $O(10^5)$.

Our results imply that code distance 21 is the minimum requirement to demonstrate the quantum advantage under the assumption of $p/p_{\mathrm{th}} = 0.1$ and target of $\epsilon = 0.01$ (See Supplementary Fig. 7 in the main text for discussion with various hardware/algorithmic requirements.). In other words, we need a classical control unit of FTQC that can estimate/correct (non-logical) errors of $d = 21$ logical qubits with 10^{-3} physical error rate and 1 μs code cycle. However, according to Ref. [73], the current state-of-the-art implementation of lattice-surgery-compatible error decoders allows a code distance up to 11 with these assumptions. Therefore, achieving physical error rates ten-times smaller than the code threshold and increasing the number of qubits are not enough to demonstrate quantum computational advantage in the field of condensed matter physics.

We expect several possible directions to overcome this difficulty. First, as we have discussed in Supplementary Fig. 7 in the main text, achieving smaller physical error rates can reduce the required code distance and number of physical qubits. The execution times are also reduced since it is proportional to code distances. Second strategy is to improve encoding strategies and error-estimation algorithms for quantum error correction, which effectively increase the code threshold or allowed code distances. Third, increasing physical qubits allows more allocations of magic-state factories, which enables further parallelization of `SELECT` modules and reduction of the required code distances. Fourth, there is room for improving the workspace utilization for lattice surgery. We can use prepared logical Bell pairs [68] for more efficient scheduling, re-use dirty logical qubits [74], or find more suitable Hamiltonian models to demonstrate advantage. Finally, we can utilize longer code cycles to allow larger code distances. However, it should be noted that this approach also increases the runtime, and more resources are demanded to demonstrate an advantage. A possible direction is considering heterogeneous architecture, i.e., using superconducting qubits for the frequent-access region and ion qubits for low-access regions, but this requires careful consideration of the communication bandwidth between different qubit species. We expect that one of these approaches cannot solely resolve the problems, and the co-design of the application, algorithm, system, and hardware is demanded in future. Our resource estimation is the first step towards such advanced designs.

[*] nyoshioka@ap.t.u-tokyo.ac.jp
[†] t-okubo@phys.s.u-tokyo.ac.jp
[‡] yasunari.suzuki.gz@hco.ntt.co.jp
[§] mizukami.wataru.qiqb@osaka-u.ac.jp

[1] Guang-Ming Zhang, Hui Hu, and Lu Yu, "Valence-bond spin-liquid state in two-dimensional frustrated spin-1/2 heisenberg antiferromagnets," Phys. Rev. Lett. **91**, 067201 (2003).

[2] Hong-Chen Jiang, Hong Yao, and Leon Balents, "Spin liquid ground state of the spin-$\frac{1}{2}$ square J_1-J_2 heisenberg model," Phys. Rev. B **86**, 024424 (2012).

[3] Wen-Jun Hu, Federico Becca, Alberto Parola, and Sandro Sorella, "Direct evidence for a gapless Z_2 spin liquid by frustrating néel antiferromagnetism," Phys. Rev. B **88**, 060402 (2013).

[4] Ling Wang and Anders W. Sandvik, "Critical level crossings and gapless spin liquid in the square-lattice spin-1/2 $J_1 - J_2$ heisenberg antiferromagnet," Phys. Rev. Lett. **121**, 107202 (2018).

[5] Yusuke Nomura and Masatoshi Imada, "Dirac-type nodal spin liquid revealed by refined quantum many-body solver using neural-network wave function, correlation ratio, and level spectroscopy," Phys. Rev. X **11**, 031034 (2021).

[6] F. D. M. Haldane, "Nonlinear field theory of large-spin heisenberg antiferromagnets: Semiclassically quantized solitons of the one-dimensional easy-axis néel state," Phys. Rev. Lett. **50**, 1153–1156 (1983).

[7] F.D.M. Haldane, "Continuum dynamics of the 1-d heisenberg antiferromagnet: Identification with the o(3) nonlinear sigma model," Physics Letters A **93**, 464–468 (1983).

[8] M B Hastings, "An area law for one-dimensional quantum systems," Journal of Statistical Mechanics: Theory and Experiment **2007**, P0

[9] Steven R. White, "Density matrix formulation for quantum renormalization groups," Phys. Rev. Lett. **69**, 2863–2866 (1992).

[10] Stellan Östlund and Stefan Rommer, "Thermodynamic limit of density matrix renormalization," Phys. Rev. Lett. **75**, 3537–3540 (1995).

[11] J Dukelsky, M. A Martín-Delgado, T Nishino, and G Sierra, "Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains," Europhys. Lett. **43**, 457–462 (1998).

Hamiltonian	lattice size	$(n_F, \#thread)$	code distance d	$N_{\rm ph}$	Repetition count r	Runtime [sec]
2d J_1-J_2 Heisenberg	4×4	(1, 1)	19	1.96e+05	5.24e+03	9.46e+02
2d J_1-J_2 Heisenberg	4×4	(16, 16)	19	2.62e+06	5.24e+03	1.18e+02
2d J_1-J_2 Heisenberg	6×6	(1, 1)	21	2.86e+05	1.26e+04	6.03e+03
2d J_1-J_2 Heisenberg	6×6	(16, 16)	21	3.33e+06	1.26e+04	5.52e+02
2d J_1-J_2 Heisenberg	8×8	(1, 1)	23	4.18e+05	2.30e+04	2.22e+04
2d J_1-J_2 Heisenberg	8×8	(16, 16)	21	3.47e+06	2.30e+04	1.66e+03
2d J_1-J_2 Heisenberg	10×10	(1, 1)	23	5.12e+05	3.67e+04	5.64e+04
2d J_1-J_2 Heisenberg	10×10	(16, 16)	23	4.35e+06	3.67e+04	4.32e+03
2d J_1-J_2 Heisenberg	12×12	(1, 1)	25	7.28e+05	5.34e+04	1.30e+05
2d J_1-J_2 Heisenberg	12×12	(16, 16)	23	4.46e+06	5.34e+04	8.68e+03
spin-1 Heisenberg chain	10	(1, 1)	19	4.68e+05	5.45e+03	6.79e+02
spin-1 Heisenberg chain	10	(16, 16)	17	2.61e+07	5.45e+03	7.22e+01
spin-1 Heisenberg chain	20	(1, 1)	21	9.49e+05	1.07e+04	3.10e+03
spin-1 Heisenberg chain	20	(16, 16)	19	6.27e+07	1.07e+04	2.65e+02
spin-1 Heisenberg chain	40	(1, 1)	23	2.04e+06	2.12e+04	1.37e+04
spin-1 Heisenberg chain	40	(16, 16)	19	1.23e+08	2.12e+04	8.73e+02
spin-1 Heisenberg chain	80	(1, 1)	23	3.82e+06	4.21e+04	5.52e+04
spin-1 Heisenberg chain	80	(16, 16)	21	2.97e+08	4.21e+04	3.68e+03
spin-1 Heisenberg chain	160	(1, 1)	25	8.72e+06	8.40e+04	2.41e+05
spin-1 Heisenberg chain	160	(16, 16)	23	7.08e+08	8.40e+04	1.49e+04
spin-1 Heisenberg chain	320	(1, 1)	27	2.00e+07	1.68e+05	1.04e+06
spin-1 Heisenberg chain	320	(16, 16)	25	1.67e+09	1.68e+05	6.34e+04
2d Fermi-Hubbard	4×4	(1, 1)	21	2.72e+05	1.26e+04	2.02e+03
2d Fermi-Hubbard	4×4	(16, 16)	21	3.24e+06	1.26e+04	2.99e+02
2d Fermi-Hubbard	6×6	(1, 1)	23	4.29e+05	2.93e+04	1.23e+04
2d Fermi-Hubbard	6×6	(16, 16)	21	3.40e+06	2.93e+04	1.59e+03
2d Fermi-Hubbard	8×8	(1, 1)	23	5.70e+05	5.31e+04	4.01e+04
2d Fermi-Hubbard	8×8	(16, 16)	23	4.32e+06	5.31e+04	5.50e+03
2d Fermi-Hubbard	10×10	(1, 1)	25	8.86e+05	8.38e+04	1.05e+05
2d Fermi-Hubbard	10×10	(16, 16)	25	5.42e+06	8.38e+04	1.46e+04

Supplementary Table XII. The estimated number of physical qubits, required code distance, and actual runtime for our target Hamiltonians. Here, the runtime is based on the T-depth required to estimate the ground state energy with target precision $\epsilon = 0.01$.

[12] J. Eisert, M. Cramer, and M. B. Plenio, "Colloquium: Area laws for the entanglement entropy," Rev. Mod. Phys. **82**, 277–306 (2010).

[13] Ulrich Schollwöck, "The density-matrix renormalization group in the age of matrix product states," Annals of Physics **326**, 96–192 (2011), january 2011 Special Issue.

[14] J. P. F. LeBlanc, Andrey E. Antipov, Federico Becca, Ireneusz W. Bulik, Garnet Kin-Lic Chan, Chia-Min Chung, Youjin Deng, Michel Ferrero, Thomas M. Henderson, Carlos A. Jiménez-Hoyos, E. Kozik, Xuan-Wen Liu, Andrew J. Millis, N. V. Prokof'ev, Mingpu Qin, Gustavo E. Scuseria, Hao Shi, B. V. Svistunov, Luca F. Tocchio, I. S. Tupitsyn, Steven R. White, Shiwei Zhang, Bo-Xiao Zheng, Zhenyue Zhu, and Emanuel Gull (Simons Collaboration on the Many-Electron Problem), "Solutions of the two-dimensional hubbard model: Benchmarks and results from a wide range of numerical algorithms," Phys. Rev. X **5**, 041041 (2015).

[15] Ling Wang and Anders W. Sandvik, "Critical level crossings and gapless spin liquid in the square-lattice spin-1/2 $J_1 - J_2$ heisenberg antiferromagnet," Phys. Rev. Lett. **121**, 107202 (2018).

[16] Tomotoshi Nishino, Yasuhiro Hieida, Kouichi Okunishi, Nobuya Maeshima, Yasuhiro Akutsu, and Andrej Gendiar, "Two-Dimensional Tensor Product Variational Formulation," Prog. Theor. Phys. **105**, 409–417 (2001).

[17] Frank Verstraete and J Ignacio Cirac, "Renormalization algorithms for quantum-many body systems in two and higher dimensions," arXiv preprint cond-mat/0407066 (2004).

[18] H. C. Jiang, Z. Y. Weng, and T. Xiang, "Accurate determination of tensor network state of quantum lattice models in two dimensions," Phys. Rev. Lett. **101**, 090603 (2008).

[19] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, "Classical simulation of infinite-size quantum lattice systems in two spatial dimensions," Phys. Rev. Lett. **101**, 250602 (2008).

[20] Philippe Corboz, "Variational optimization with infinite projected entangled-pair states," Phys. Rev. B **94**, 035133 (2016).

[21] Laurens Vanderstraeten, Jutho Haegeman, Philippe Corboz, and Frank Verstraete, "Gradient methods for variational optimization of projected entangled-pair states," Phys. Rev. B **94**, 155123 (2016).

[22] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang, "Differentiable programming tensor networks," Phys. Rev. X **9**, 031041 (2019).

[23] Matthew Fishman, Steven R. White, and E. Miles Stoudenmire, "The ITensor Software Library for Tensor Network Calculations," SciPost Phys. Codebases , 4 (2022).

[24] U. Schollwöck, "The density-matrix renormalization group," Rev. Mod. Phys. **77**, 259–315 (2005).

[25] Steven R. White and David A. Huse, "Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic s=1 heisenberg chain," Phys. Rev. B **48**, 3844–3852 (1993).

[26] Örs Legeza and Gábor Fáth, "Accuracy of the density-matrix renormalization-group method," Phys. Rev. B **53**, 14349–14358 (1996).

[27] E. M. Stoudenmire and Steven R. White, "Real-space parallel density matrix renormalization group," Phys. Rev. B **87**, 155137 (2013).

[28] Johnnie Gray, "quimb: a python library for quantum information and many-body calculations," Journal of Open Source Software **3**, 819 (2018).

[29] Norbert Schuch, Michael M Wolf, Frank Verstraete, and J Ignacio Cirac, "Computational complexity of projected entangled pair states," Physical review letters **98**, 140506 (2007).

[30] Hiroshi Ueda and Koichi Kusakabe, "Determination of boundary scattering, magnon-magnon scattering, and the haldane gap in heisenberg spin chains," Phys. Rev. B **84**, 054446 (2011).

[31] F. D. M. Haldane, "Nonlinear field theory of large-spin heisenberg antiferromagnets: Semiclassically quantized solitons of the one-dimensional easy-axis néel state," Phys. Rev. Lett. **50**, 1153–1156 (1983).

[32] Yimin Ge, András Molnár, and J Ignacio Cirac, "Rapid adiabatic preparation of injective projected entangled pair states and gibbs states," Phys. Rev. Lett. **116**, 080503 (2016).

[33] J.R. Johansson, P.D. Nation, and Franco Nori, "Qutip 2: A python framework for the dynamics of open quantum systems," Computer Physics Communications **184**, 1234–1240 (2013).

[34] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," Nature Methods **17**, 261–272 (2020).

[35] Jutho Haegeman, J. Ignacio Cirac, Tobias J. Osborne, Iztok Pižorn, Henri Verschelde, and Frank Verstraete, "Time-dependent variational principle for quantum lattices," Phys. Rev. Lett. **107**, 070601 (2011).

[36] The tensor network library ITensor currently does not support SU(2) symmetry for MPS, while U(1) is available.

[37] A Yu Kitaev, "Quantum measurements and the abelian stabilizer problem," arXiv preprint quant-ph/9511026 (1995).

[38] Daniel S. Abrams and Seth Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," Phys. Rev. Lett. **83**, 5162–5165 (1999).

[39] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer, "Elucidating reaction mechanisms on quantum computers," PNAS **114**, 7555–7560 (2017).

[40] Masuo Suzuki, "General theory of fractal path integrals with applications to many ‐ body theories and statistical physics," J. Math. Phys. **32**, 400–407 (1991).

[41] Earl Campbell, "Random compiler for fast hamiltonian simulation," Phys. Rev. Lett. **123**, 070503 (2019).

[42] Earl Campbell, "Shorter gate sequences for quantum computing by mixing unitaries," Phys. Rev. A **95**, 042306 (2017).

[43] Matthew B. Hastings, "Turning gate synthesis errors into incoherent errors," Quantum Info. Comput. **17**, 488–494 (2017).

[44] Andrew M. Childs, Aaron Ostrander, and Yuan Su, "Faster quantum simulation by randomization," Quantum **3**, 182 (2019).

[45] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush, "Even more efficient quantum computations of chemistry through tensor hypercontraction," PRX Quantum **2**, 030305 (2021).

[46] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Exponential improvement in precision for simulating sparse hamiltonians," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 283–292.

[47] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Simulating hamiltonian dynamics with a truncated taylor series," Phys. Rev. Lett. **114**, 090502 (2015).

[48] Ryan Babbush, Dominic W Berry, Ian D Kivlichan, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik, "Exponentially more precise quantum simulation of fermions in second quantization," New Journal of Physics **18**, 033032 (2016).

[49] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush, "Improved techniques for preparing eigenstates of fermionic hamiltonians," npj Quantum Information **4**, 22 (2018).

[50] David Poulin, Alexei Kitaev, Damian S. Steiger, Matthew B. Hastings, and Matthias Troyer, "Quantum algorithm for spectral measurement with a lower gate count," Phys. Rev. Lett. **121**, 010501 (2018).

[51] Guang Hao Low and Isaac L Chuang, "Hamiltonian simulation by qubitization," Quantum **3**, 163 (2019).

[52] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven, "Encoding electronic spectra in quantum circuits with linear t complexity," Phys. Rev. X **8**, 041015 (2018).

[53] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang, "Grand unification of quantum algorithms," PRX Quantum **2**, 040203 (2021).

[54] Wojciech Górecki, Rafał Demkowicz-Dobrzański, Howard M. Wiseman, and Dominic W. Berry, "π-corrected heisenberg limit," Phys. Rev. Lett. **124**, 030501 (2020).

[55] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S. Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer, "Quantum computing enhanced computational catalysis," Phys. Rev. Research **3**, 033055 (2021).

[56] Pablo A. M. Casares, Roberto Campos, and M. A. Martin-Delgado, "TFermion: A non-Clifford gate cost assessment library of quantum phase estimation algorithms for quantum chemistry," Quantum **6**, 768 (2022).

[57] Yuki Koizumi, Yasunari Suzuki, Nobuyuki Yoshioka, and Yuuki Tokunaga, in preparation.

[58] Craig Gidney, "Halving the cost of quantum addition," Quantum **2**, 74 (2018).

[59] Vadym Kliuchnikov, Kristin Lauter, Romy Minko, Adam Paetznick, and Christophe Petit, "Shorter quantum circuits via single-qubit gate approximation," Quantum **7**, 1208 (2023).

[60] V.V. Shende, S.S. Bullock, and I.L. Markov, "Synthesis of quantum-logic circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**, 1000–1010 (2006).

[61] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill, "Topological quantum memory," Int. J. Theor. Phys. **43**, 4452–4505 (2002).

[62] Sergey B Bravyi and A Yu Kitaev, "Quantum codes on a lattice with boundary," arXiv preprint quant-ph/9811052 (1998).

[63] Austin G Fowler and Craig Gidney, "Low overhead quantum computation using lattice surgery," arXiv preprint arXiv:1808.06709 (2018).

[64] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Physical Review A **86**, 032324 (2012).

[65] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter, "Surface code quantum computing by lattice surgery," New Journal of Physics **14**, 123011 (2012).

[66] Daniel Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," Quantum **3**, 128 (2019).

[67] Christopher Chamberland and Earl T. Campbell, "Universal quantum computing with twist-free and temporally encoded lattice surgery," PRX Quantum **3** (2022), 10.1103/prxquantum.3.010331.

[68] Michael Beverland, Vadym Kliuchnikov, and Eddie Schoute, "Surface code compilation via edge-disjoint paths," PRX Quantum **3**, 020342 (2022).

[69] Leon Riesebos, Xiang Fu, Savvas Varsamopoulos, Carmen G Almudever, and Koen Bertels, "Pauli frames for quantum computer architectures," in *Proceedings of the 54th Annual Design Automation Conference 2017* (2017) pp. 1–6.

[70] Austin G Fowler, "Time-optimal quantum computation," arXiv preprint arXiv:1210.4626 (2012).

[71] Adam Holmes, Mohammad Reza Jokar, Ghasem Pasandi, Yongshan Ding, Massoud Pedram, and Frederic T Chong, "NISQ+: Boosting quantum computing power by approximating quantum error correction," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)* (IEEE, 2020) pp. 556–569.

[72] Joshua J. Goings, Alec White, Joonho Lee, Christofer S. Tautermann, Matthias Degroote, Craig Gidney, Toru Shiozaki, Ryan Babbush, and Nicholas C. Rubin, "Reliably assessing the electronic structure of cytochrome p450 on today's classical computers and tomorrow's quantum computers," PNAS **119**, e2203533119 (2022).

[73] Yosuke Ueno, Masaaki Kondo, Masamitsu Tanaka, Yasunari Suzuki, and Yutaka Tabuchi, "QULATIS: A quantum error correction methodology toward lattice surgery," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, 2022).

[74] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer, "Trading t-gates for dirty qubits in state preparation and unitary synthesis," arXiv preprint arXiv:1812.00954 (2018).