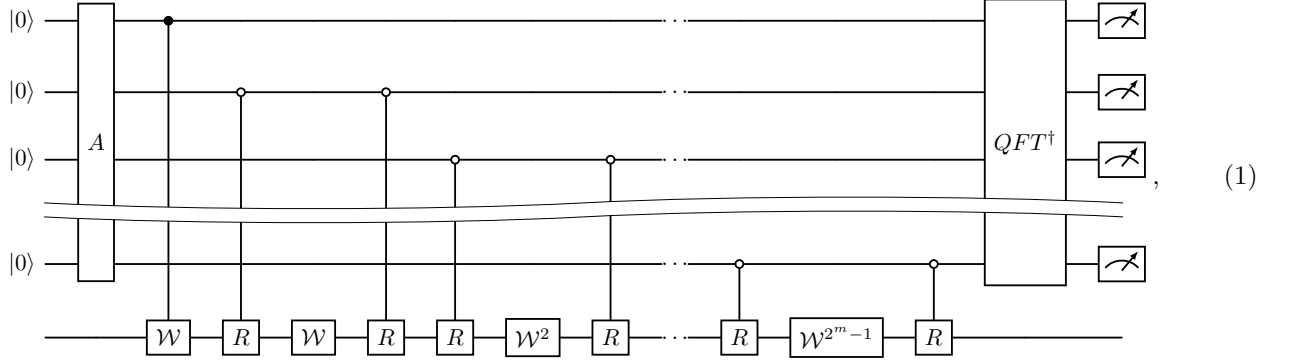


Supplementary Notes for: Resource-optimized fault-tolerant simulation of the Fermi-Hubbard model and high-temperature superconductor models

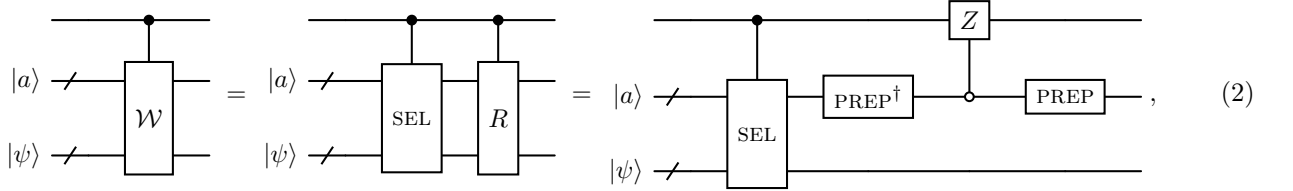
Supplementary Note 1 Resource estimation for Qubitization algorithms

A. Overview

We present a high-level circuit construction of the qubitization algorithm, used here, from [1], as well as our error analysis and resource estimation pipeline. Given a Hamiltonian H , the algorithm applies a quantum phase estimation (QPE) algorithm to the walk operator $\mathcal{W}$, according to the following circuit:



where A prepares a window state to improve the QPE success probability [2] and $QFT^\dagger$ is the inverse quantum Fourier transform. The (controlled) walk operator $\mathcal{W}$ and reflection operator R are realized with the circuit:



where given a Hamiltonian H expressed as a linear combination of unitaries (LCU) [3, 4]:

$$H = \sum_{i=1}^M c_i P_i, \quad c_i \in \mathbb{R}, \quad c_i \geq 0 \quad (3)$$

with P_i being unitary and self-inverse. SEL and PREP are used to block-encode [4] H as follows:

$$\text{PREP} |0\rangle = \sum_{i=1}^M \sqrt{\frac{c_i}{\lambda}} |i\rangle, \quad \lambda = \sum_{i=1}^M c_i, \quad (4)$$

$$\text{SEL} = \sum_{i=1}^M |i\rangle\langle i| \otimes P_i \quad (5)$$

$$\Rightarrow (\langle 0| \otimes I) \text{PREP}^\dagger \cdot \text{SEL} \cdot \text{PREP}(|0\rangle \otimes I) = \frac{H}{\lambda}. \quad (6)$$

As noted in [1], since both A and $QFT^\dagger$ have gate complexity $\tilde{O}(m)$ [5], where m is the number of phase qubits in the QPE, whereas $\mathcal{W}$ and R have query complexity $O(2^m)$, the resource costs of A and $QFT^\dagger$ are completely neglected. Hence, they are ignored in the following discussion and analysis.

As in [1], we measure the simulation error through the root-mean-square error of the estimated phase $\Delta\phi$, which is related to the error budget in energy ΔE via

$$\Delta E = \lambda \Delta \cos(\phi) \leq \lambda \Delta \phi. \quad (7)$$

There are two sources of error: the Holevo variance [6, 7] of the phase estimation output $\epsilon_{QPE} = \frac{\pi}{2^{m+1}}$ [1], and the circuit synthesis error of the walk operator ϵ_W , i.e.,

$$\left(\frac{\Delta E}{\lambda}\right)^2 \leq (\Delta\phi)^2 = \epsilon_{QPE}^2 + \epsilon_W^2. \quad (8)$$

We split the total error budget according to

$$\epsilon_{QPE}^2 = x \left(\frac{\Delta E}{\lambda}\right)^2, \quad \epsilon_W^2 = (1-x) \left(\frac{\Delta E}{\lambda}\right)^2, \quad (9)$$

where $0 < x < 1$. Then, we can compute the sufficient number of phase qubits as

$$\frac{\pi}{2^{m+1}} = \frac{\sqrt{x}\Delta E}{\lambda} \implies m = \left\lceil \log_2 \left(\frac{\pi\lambda}{2\sqrt{x}\Delta E} \right) \right\rceil < \log_2 \left(\frac{\pi\lambda}{\sqrt{x}\Delta E} \right). \quad (10)$$

Then, according to the circuit in (1), we need at most

$$2^m < \frac{\pi\lambda}{\sqrt{x}\Delta E} \quad (11)$$

queries to the walk operator to estimate the energy to within error ΔE . Therefore, one can estimate the gate count of the algorithm by multiplying this query count with the gate count of the walk operator, i.e., one SEL and two PREP, where the comparatively much smaller cost of the multiply controlled Z in (2) is neglected [1].

In what follows, we describe the low-level circuit implementation of the SEL and PREP oracles, and derive the resource estimates for simulating the Fermi-Hubbard model, one-band cuprate model, and two-band pnictide model. We adopt the Clifford + T + Toffoli gate set, as in [8], and assume that 2 T gates can be performed using 1 Toffoli gates, i.e., 2 T = 1 Toffoli, with catalyst-assisted circuits from [9].

B. Fermi-Hubbard model

The Fermi-Hubbard model can be expressed as

$$H = t \sum_{\langle i,j \rangle, \sigma} a_{i,\sigma}^\dagger a_{j,\sigma} + u \sum_{i, \sigma \neq \sigma'} \left(n_{i,\sigma} - \frac{1}{2} \right) \left(n_{i,\sigma'} - \frac{1}{2} \right), \quad (12)$$

where the notation $\langle i, j \rangle$ denotes nearest neighbors, and $\sigma \in \{\uparrow, \downarrow\}$ labels the spin. Upon Jordan-Wigner (JW) transformation [10],

$$H = \frac{t}{2} \sum_{\langle i,j \rangle, \sigma} (X_{i,\sigma} \vec{Z} X_{j,\sigma} + Y_{i,\sigma} \vec{Z} Y_{j,\sigma}) + \frac{u}{4} \sum_{i, \sigma \neq \sigma'} Z_{i,\sigma} Z_{i,\sigma'}, \quad (13)$$

where $\vec{Z}$ denotes the product of Z operators that could appear in $d > 1$ dimensions due to JW transformation. In the following analysis, we will assume a default JW ordering that follows a serpentine path, where fermions at the same site with opposite spins are neighbors, i.e., spin is the fastest changing variable when traversing the JW path. We consider the Fermi-Hubbard model on an $L \times L$ lattice with periodic boundary conditions.

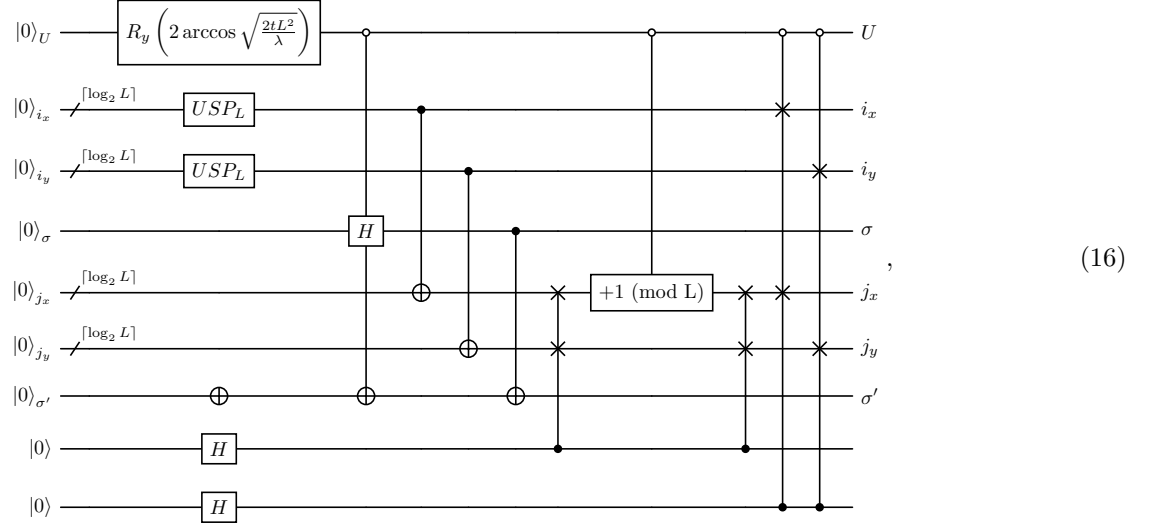
We first provide circuit implementations that are almost the same as those in [1], with minor differences arising from the $-1/2$ shift in the number operators. The PREP oracle is defined by

$$\begin{aligned} & \text{PREP } |0\rangle \\ &= \sum_{i_x, i_y=0}^{L-1} \left[\sqrt{\frac{u}{4\lambda}} |1\rangle_U |i_x\rangle |i_y\rangle |0\rangle_\sigma |i_x\rangle |i_y\rangle |1\rangle_{\sigma'} \right. \\ & \quad \left. + \sqrt{\frac{t}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |i_x\rangle |i_y\rangle |\sigma\rangle \left(|i_x+1\rangle |i_y\rangle + |i_x\rangle |i_y+1\rangle + |i_x-1\rangle |i_y\rangle + |i_x\rangle |i_y-1\rangle \right) |\sigma\rangle \right], \end{aligned} \quad (14)$$

where i_x and i_y are components of i in the x - and y -directions, and

$$\lambda = 4L^2 t + \frac{uL^2}{4}. \quad (15)$$

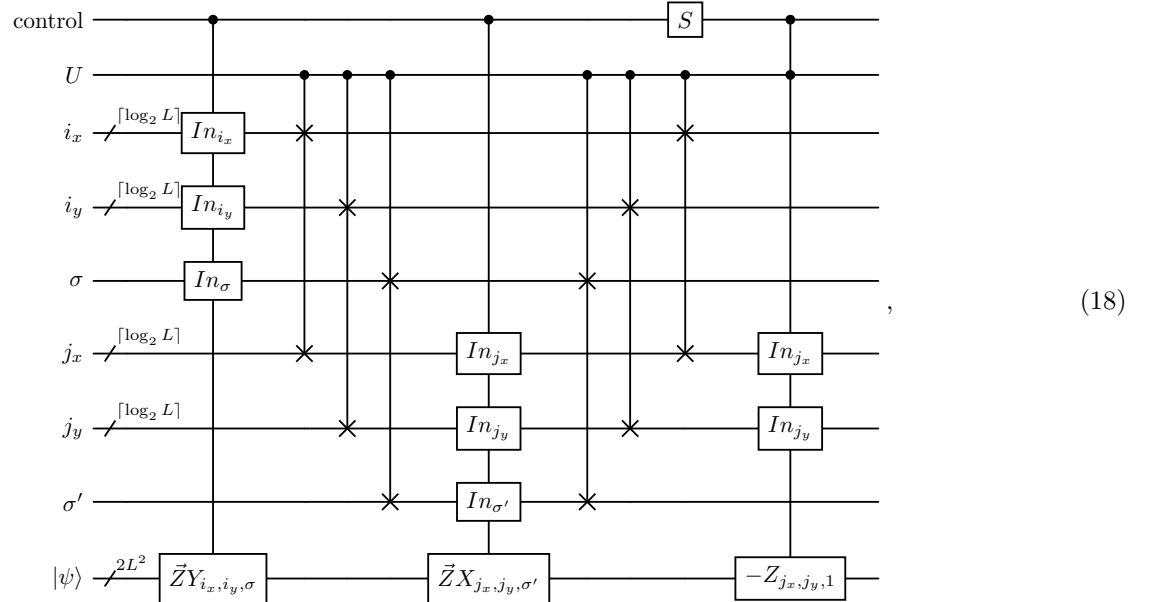
We implement this oracle using the circuit:



The SEL oracle satisfies the equation:

$$\text{SEL} |U, i_x, i_y, \sigma, j_x, j_y, \sigma'\rangle |\psi\rangle = |U, i_x, i_y, \sigma, j_x, j_y, \sigma'\rangle \otimes \begin{cases} Z_{i,\sigma} Z_{j,\sigma'} |\psi\rangle & U \wedge (\mathbf{i} = \mathbf{j}) \wedge (\sigma = 0) \wedge (\sigma' = 1) \\ X_{i,\sigma} \vec{Z} X_{j,\sigma} |\psi\rangle & \neg U \wedge (\mathbf{i} < \mathbf{j}) \wedge (\sigma = \sigma') \\ Y_{i,\sigma} \vec{Z} Y_{j,\sigma} |\psi\rangle & \neg U \wedge (\mathbf{i} > \mathbf{j}) \wedge (\sigma = \sigma') \\ \text{Undefined} & \text{otherwise,} \end{cases} \quad (17)$$

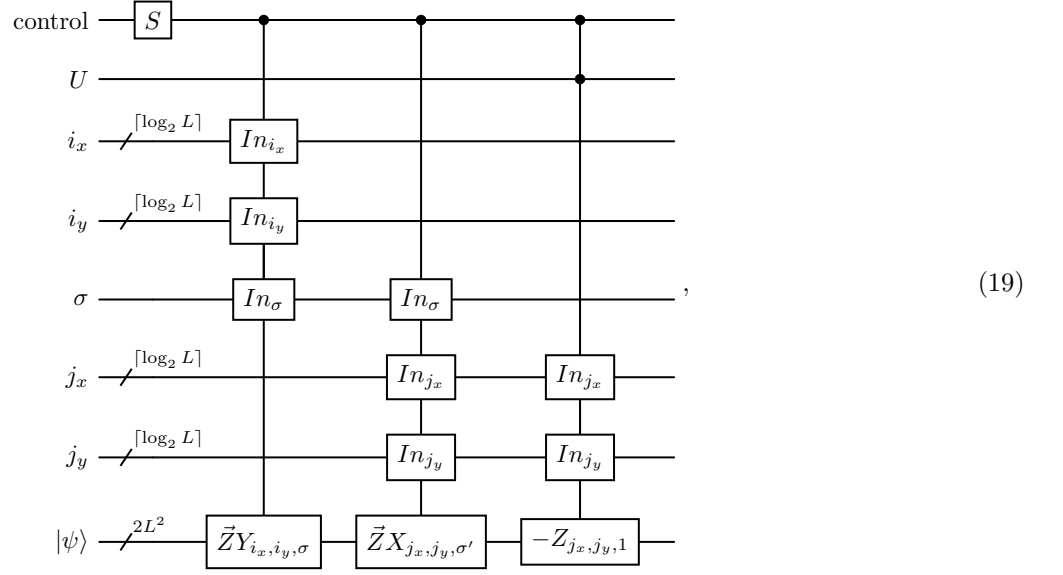
where, when $\mathbf{i}$ and $\mathbf{j}$ are compared, they are treated as two-digit integers with radix L , i.e., $\mathbf{i} = i_x + i_y L$. We implement controlled SEL using the circuit:



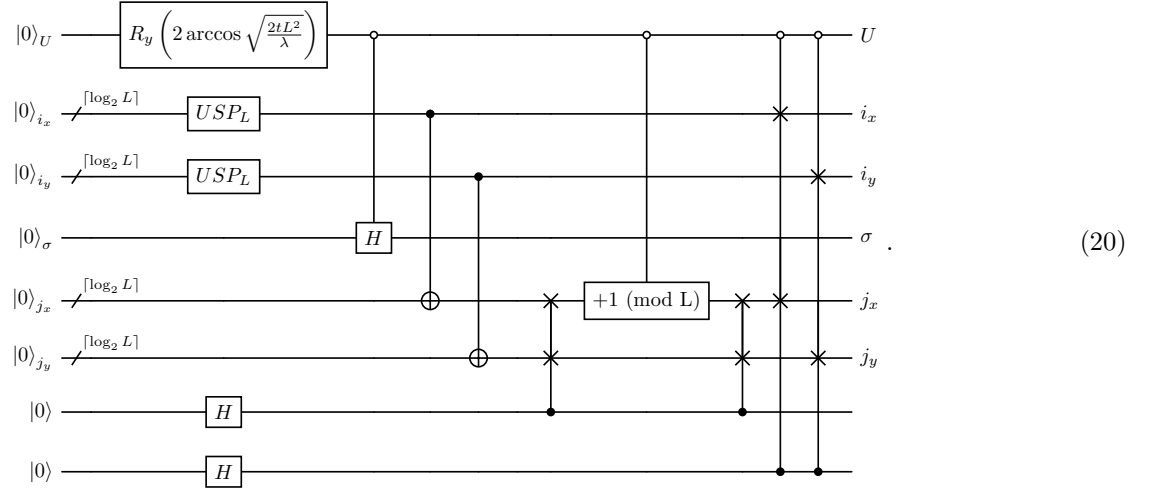
where we have used the ranged operation and unary iteration operation shown in figure 7 and 9 of [1]; note that the ‘In’ boxes denote the address portion of a QROM gadget.

Now we show that the controlled swap gates in (18) and the σ' qubit, as well as the gates acting on it, are superfluous. We note that this optimization is used in [11] independent from this work. First, we notice that the controlled swap gates between registers i_x and j_x , and those between registers i_y and j_y are unnecessary because $\mathbf{i} = \mathbf{j}$ when $U = 1$. The controlled swap gates between qubits σ and σ' can also be removed, and σ replaces σ' as part of the address of the middle QROM. This works because (i) when $U = 0$, $\sigma = \sigma'$, and (ii) when $U = 1$, σ and σ' are swapped. This

way, one can see that the σ' qubit is redundant and the gates acting on it in (16) can also be removed. As a result, we obtain the optimized SEL circuit:



and the optimized PREP circuit:

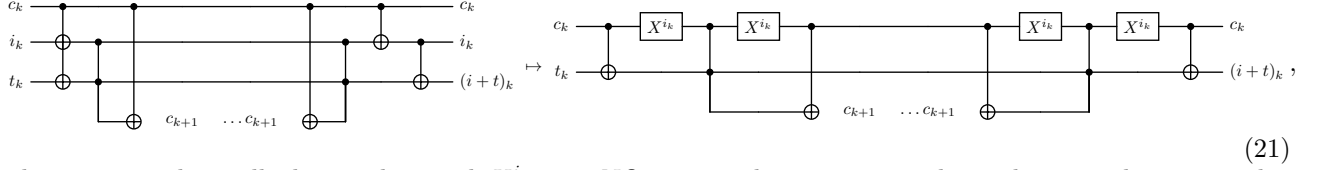


We detail below the non-Clifford count and ancilla count of various circuit components in SEL and PREP.

- A singly controlled QROM of N items costs $N - 1$ Toffoli gates and $\lceil \log_2(N) \rceil$ ancilla qubits [1].
- Each arbitrary-angle single-qubit rotation gate about the $x/y/z$ axis can be synthesized using the repeat-until-success (RUS) method from [12] with a mean T count of $0.53 \log_2(1/\delta) + 4.86$ and 1 ancilla qubit, where δ is the error budget of the rotation gate.
- Each n -controlled swap gate, i.e., $C^n \text{Swap}$ is synthesized by turning the middle CNOT gate of the standard decomposition of a swap gate into 3 CNOTs into an n -controlled NOT, i.e., $C^n X$ gate.
- Each n -controlled Hadamard gate, i.e., $C^n H = S^\dagger H T^\dagger (C^n X) T H S$, where the single qubit rotation gates act on the target qubit of $C^n X$.
- Using the method from [13], adding n control qubits to a controlled unitary costs n Toffoli gates and n clean ancilla qubits in $|0\rangle$. So, a $C^n X$ gate costs $n - 1$ Toffoli gates and $n - 2$ ancilla qubits.
- Uniform state preparation (USP): We aim to prepare $\sum_{l=0}^{2^k m - 1} c_l |l\rangle$, where $|c_l|^2 = \frac{1}{2^{k m}}$. If $m = 1$, only Hadamard gates are required and no ancilla qubits are needed. For $m > 1$, the amplitude-amplification method from [1]

or [14] costs $2\lceil\log_2(m)\rceil - 2$ Toffoli gates, $2 R_z$ gates, and $\lceil\log_2(m)\rceil$ ancilla qubits for $m > 1$. Converting the R_z count to the expected T count using the RUS circuits from [12] and then to Toffoli count according to $1 \text{ Toffoli} = 2 \text{ T}$ [9], this method is expected to cost $2\lceil\log_2(m)\rceil + 2(0.53 \log_2(1/\epsilon) + 4.86) - 2$ Toffoli gates, where ϵ is the error budget per R_z .

- Addition/Subtraction with classical numbers and between addends of different lengths: We consider an n -bit adder from [15], and modify its building-block according to



when i_k is a classically known bit, and X^{i_k} is a NOT gate when $i_k = 1$ and an identity otherwise. This modification works because the first CNOT between c_k and i_k copies $c_k(\oplus 1)$ into i_k when $i_k = 0$ (1), and hence, the subsequent Toffoli gate can be controlled by $c_k(\oplus 1)$ if $i_k = 0$ (1) instead of i_k . There are two corollaries. First, when adding a classical state, i.e., a computational basis state, to a quantum state, the classical state does not need to be stored in qubits. Second, when adding two quantum states of different lengths, the shorter one does not need to be padded by 0's.

- Controlled addition with classical numbers: Instead of using a controlled adder, which costs $2n - 1$ Toffoli gates [15], we initialize the classical number in a quantum register using CNOTs before adding it into an quantum addend using an uncontrolled adder, before undoing the CNOTs. As such, a controlled incrementer costs $n - 2$ Toffoli gates and n ancilla qubits, where 1 ancilla qubit stores the classical addend 1 and n is the length of the quantum addend.

The non-Clifford gate counts per PREP are 1 R_y gate, 2 T gates, and $5\lceil\log_2(L)\rceil - 1$ Toffoli gates, as well as the non-Clifford gates required by the two USP_L 's. The non-Clifford gate count per SEL is $5L^2 - 2$ Toffoli gates. Since the ancilla qubit count of a SEL is larger than that of a PREP, the ancilla qubit count of a walk operator, i.e., size of $|a\rangle$ in (2), is given by that of a SEL, which is $6\lceil\log_2(L)\rceil + 4$. Therefore, combining with (10), the total qubit count is

$$\left\lceil \log_2 \left(\frac{\pi \lambda L^6}{2\sqrt{x}\Delta E} \right) \right\rceil + 2L^2 + 3. \quad (22)$$

We multiply the gate count per walk operator, i.e., one SEL and two PREP, by the query count in (11) to obtain the total respective numbers of $R_{y/z}$, T, and Toffoli gates:

$$N_R = \begin{cases} \frac{2\pi\lambda}{\sqrt{x}\Delta E}, & \text{if } L \text{ is a binary power.} \\ \frac{6\pi\lambda}{\sqrt{x}\Delta E}, & \text{otherwise.} \end{cases} \quad (23)$$

$$N_T = \frac{4\pi\lambda}{\sqrt{x}\Delta E}, \quad (24)$$

$$N_{\text{Toff}} = \begin{cases} \frac{\pi\lambda}{\sqrt{x}\Delta E} (5L^2 + 10\lceil\log_2(L)\rceil - 4), & \text{If } L \text{ is a binary power.} \\ \frac{\pi\lambda}{\sqrt{x}\Delta E} (5L^2 + 10\lceil\log_2(L)\rceil + 4\lceil\log_2(m)\rceil - 4), & \text{otherwise,} \end{cases} \quad (25)$$

where m is the smallest integer such that $L = 2^k m$. Then, the $R_{y/z}$ gates are synthesized with Clifford + T gates using RUS circuits [12], which implies

$$N_T = \begin{cases} \frac{\pi\lambda}{\sqrt{x}\Delta E} \left(4 + 2 \cdot (0.53 \log_2 \left(\frac{2\pi\lambda^2}{\sqrt{x(1-x)(\Delta E)^2}} \right) + 4.86) \right), & \text{if } L \text{ is a binary power.} \\ \frac{\pi\lambda}{\sqrt{x}\Delta E} \left(4 + 6 \cdot (0.53 \log_2 \left(\frac{6\pi\lambda^2}{\sqrt{x(1-x)(\Delta E)^2}} \right) + 4.86) \right), & \text{otherwise.} \end{cases} \quad (26)$$

where we used (9), i.e., $\epsilon_W = \frac{\sqrt{1-x}\Delta E}{\lambda}$. Assuming that $1 \text{ Toffoli} = 2 \text{ T}$ [9], we numerically minimize the total Toffoli count, i.e., $N_{\text{Toff}} + N_T/2$, with respect to x , and find $x \approx 0.99$ to be near optimal, i.e., yielding the lowest Toffoli count. The total Toffoli and qubit counts for $u = 8$ and $t = 1$ are summarized in Supplementary Table 1.

L	# Toffoli	# qubits
4	4.33e5	58
6	9.75e5	102
8	1.36e6	160
10	2.19e6	234
12	3.03e6	324
14	3.99e6	429
16	4.96e6	550
18	6.38e6	687
20	7.81e6	840
22	9.36e6	1009
24	1.11e7	1194
26	1.29e7	1395
28	1.50e7	1611
30	1.71e7	1844
32	1.93e7	2092

Supplementary Table 1: Resource estimates of qubitization simulations of the $L \times L$ Fermi-Hubbard model with $u = 8$, $t = 1$, and $\Delta E = 0.51\%L^2$.

C. Single-orbital Cuprate model

The cuprate model considered here extends the Fermi-Hubbard model by including second- and third-nearest-neighbor hopping terms, leading to the following Hamiltonian:

$$H = t \sum_{\langle i,j \rangle, \sigma} a_{i,\sigma}^\dagger a_{j,\sigma} + t' \sum_{\langle\langle i,j \rangle\rangle, \sigma} a_{i,\sigma}^\dagger a_{j,\sigma} + t'' \sum_{\langle\langle\langle i,j \rangle\rangle\rangle, \sigma} a_{i,\sigma}^\dagger a_{j,\sigma} + u \sum_{i,\sigma \neq \sigma'} \left(n_{i,\sigma} - \frac{1}{2} \right) \left(n_{i,\sigma'} - \frac{1}{2} \right), \quad (27)$$

where t' and t'' are the hopping strength between second- and third-nearest neighbors, respectively. Post-JW transformation, the Hamiltonian becomes

$$H = \frac{t}{2} \sum_{\langle i,j \rangle, \sigma} (X_{i,\sigma} \vec{Z} X_{j,\sigma} + Y_{i,\sigma} \vec{Z} Y_{j,\sigma}) + \frac{t'}{2} \sum_{\langle\langle i,j \rangle\rangle, \sigma} (X_{i,\sigma} \vec{Z} X_{j,\sigma} + Y_{i,\sigma} \vec{Z} Y_{j,\sigma}) + \frac{t''}{2} \sum_{\langle\langle\langle i,j \rangle\rangle\rangle, \sigma} (X_{i,\sigma} \vec{Z} X_{j,\sigma} + Y_{i,\sigma} \vec{Z} Y_{j,\sigma}) + \frac{u}{4} \sum_{i,\sigma \neq \sigma'} Z_{i,\sigma} Z_{i,\sigma'}, \quad (28)$$

where $\vec{Z}$ denotes the product of Z operators that could appear in $d > 1$ dimensions due to JW transformation

The PREP oracle performs the transformation:

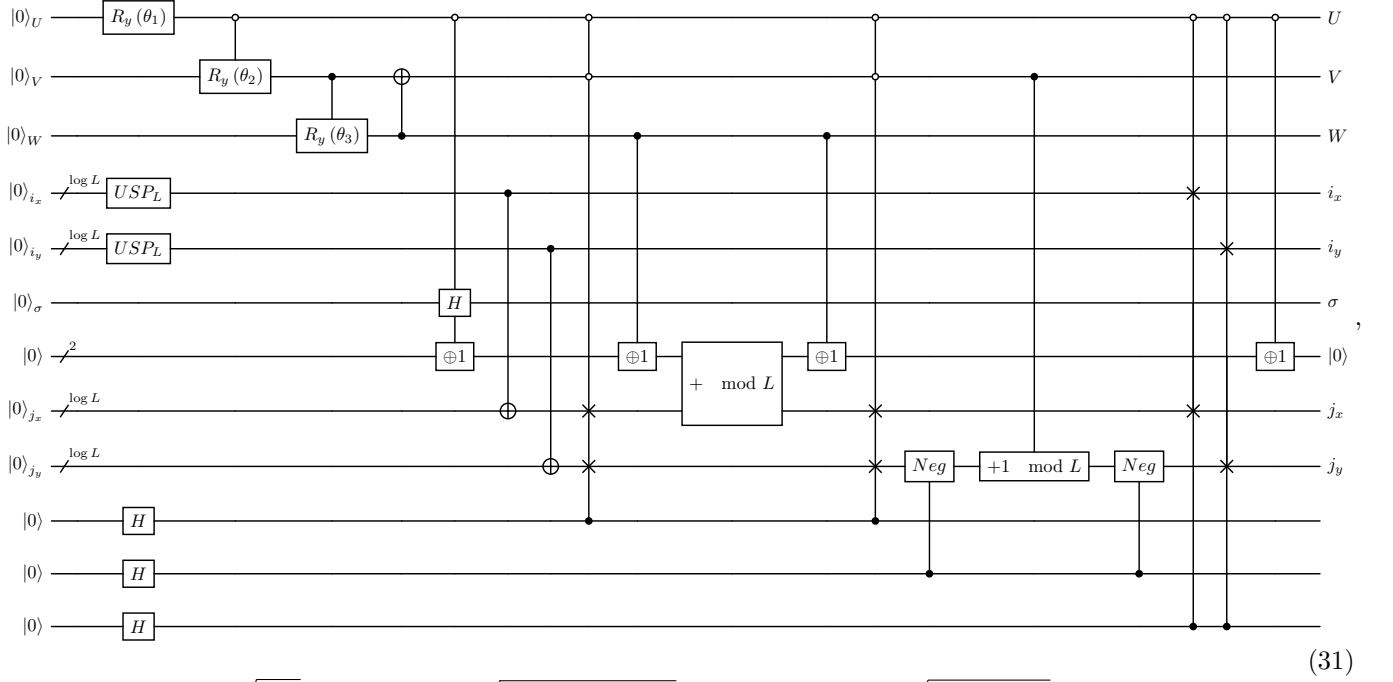
PREP $|0\rangle$

$$= \sum_{i_x, i_y=0}^{L-1} \left[\sqrt{\frac{u}{4\lambda}} |1\rangle_U |0\rangle_V |0\rangle_W |i_x\rangle |i_y\rangle |0\rangle_\sigma |i_x\rangle |i_y\rangle |1\rangle_{\sigma'} \right. \\ + \sqrt{\frac{t}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |0\rangle_V |0\rangle_W |i_x\rangle |i_y\rangle |\sigma\rangle \left(|i_x+1\rangle |i_y\rangle + |i_x\rangle |i_y+1\rangle + |i_x-1\rangle |i_y\rangle + |i_x\rangle |i_y-1\rangle \right) |\sigma\rangle \\ + \sqrt{\frac{t'}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |1\rangle_V |0\rangle_W |i_x\rangle |i_y\rangle |\sigma\rangle \left(|i_x+1\rangle |i_y+1\rangle + |i_x+1\rangle |i_y-1\rangle + |i_x-1\rangle |i_y+1\rangle + |i_x-1\rangle |i_y-1\rangle \right) |\sigma\rangle \\ \left. + \sqrt{\frac{t''}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |0\rangle_V |1\rangle_W |i_x\rangle |i_y\rangle |\sigma\rangle \left(|i_x+2\rangle |i_y\rangle + |i_x\rangle |i_y+2\rangle + |i_x-2\rangle |i_y\rangle + |i_x\rangle |i_y-2\rangle \right) |\sigma\rangle \right], \quad (29)$$

where

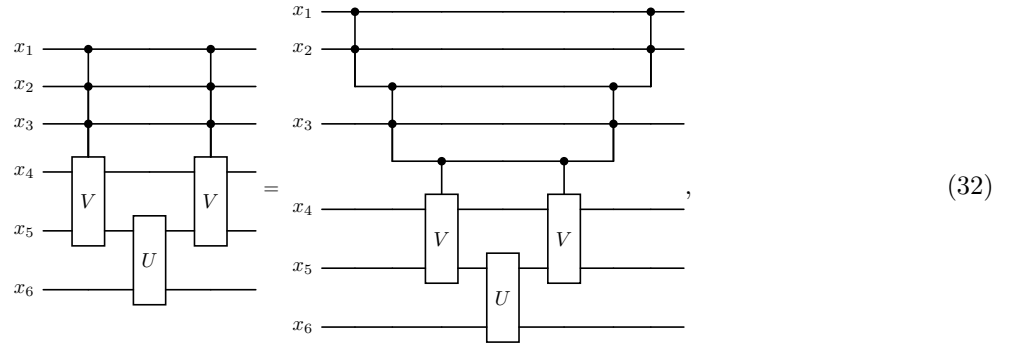
$$\lambda = 4L^2(t + t' + t'') + \frac{uL^2}{4}. \quad (30)$$

Its circuit implementation is:



where $\theta_1 = 2 \arcsin \sqrt{\frac{L^2 u}{2\lambda}}$, $\theta_2 = 2 \arccos \sqrt{(1 + \frac{t}{t'} + \frac{t}{t''})^{-1}}$, and $\theta_3 = 2 \arccos \sqrt{(1 + \frac{t'}{t''})^{-1}}$. We provide some clarifications and explanation about the components in this circuit that are absent in (20).

- The first four gates acting on registers U, V, W initialize them in the correct bit patterns and amplitudes to be distributed by the other gates, as per (29).
- The controlled R_y gates cost two uncontrolled R_y gates and two CNOT gates [16].
- The first two $C^3\text{Swap}$ gates are compiled using the identity



where V is taken to be a swap gate.

- The first controlled $\oplus 1$ operation is a CNOT that initializes an ancilla register to $|01\rangle = |1\rangle$ to be added to register j_x for generating the second and third lines in (29). The second controlled $\oplus 1$ operation consists of two CNOTs that turn $|01\rangle$ to $|10\rangle = |2\rangle$ to be added to register j_x for generating the fourth line in (29). The third and fourth controlled $\oplus 1$ operations uncompute the second and first ones, respectively.
- The pair of controlled Neg operations are CNOTs that turn the adder conjugated by them into a subtractor [17].

The SEL implementation is the same as that in (19).

The non-Clifford gate counts per PREP are 5 R_y gates, 2 T gates, and $6\lceil \log_2(L) \rceil + 2$ Toffoli gates, as well as the non-Clifford gates required by the USP_L 's. The ancilla qubit count is the same as that of a SEL, so the total qubit

count is

$$\left\lceil \log_2 \left(\frac{\pi \lambda L^6}{2\sqrt{x}\Delta E} \right) \right\rceil + 2L^2 + 3. \quad (33)$$

The T and Toffoli counts are

$$N_T = \begin{cases} \frac{\pi \lambda}{\sqrt{x}\Delta E} \left(4 + 10 \cdot (0.53 \log_2 \left(\frac{10\pi \lambda^2}{\sqrt{x(1-x)}(\Delta E)^2} \right) + 4.86) \right), & \text{if } L \text{ is a binary power.} \\ \frac{\pi \lambda}{\sqrt{x}\Delta E} \left(4 + 14 \cdot (0.53 \log_2 \left(\frac{14\pi \lambda^2}{\sqrt{x(1-x)}(\Delta E)^2} \right) + 4.86) \right), & \text{otherwise.} \end{cases} \quad (34)$$

$$N_{Tof} = \begin{cases} \frac{\pi \lambda}{\sqrt{x}\Delta E} (5L^2 + 12\lceil \log_2(L) \rceil + 2), & \text{if } L \text{ is a binary power.} \\ \frac{\pi \lambda}{\sqrt{x}\Delta E} (5L^2 + 12\lceil \log_2(L) \rceil + 4\lceil \log_2(m) \rceil + 2), & \text{otherwise.} \end{cases} \quad (35)$$

As in the Fermi-Hubbard simulations, we numerically minimize the total Toffoli count, i.e., $N_{Tof} + N_T/2$, with respect to x , and find $x \approx 0.99$ to be near optimal. We set $u = 8$, $t = 1$, $t' = 0.3$, and $t'' = 0.2$, which are in range of the parameter settings used in condensed matter literature [18–22]. We further choose $\Delta E = 0.51\%L^2$, assuming that this model is as hard to simulate classically as the Fermi-Hubbard model. Though, we believe it is likely harder and so, a larger ΔE will likely suffice for a quantum advantage; we leave for future work to determine this ΔE value. The total Toffoli and qubit counts are summarized in Supplementary Table 2.

L	# Toffoli	# qubits
4	1.04e6	59
6	1.78e6	102
8	2.29e6	161
10	3.41e6	235
12	4.53e6	324
14	5.81e6	430
16	7.10e6	551
18	9.01e6	688
20	1.09e7	841
22	1.30e7	1010
24	1.53e7	1194
26	1.78e7	1395
28	2.05e7	1612
30	2.33e7	1844
32	2.62e7	2093

Supplementary Table 2: Resource estimates of qubitization simulations of the $L \times L$ cuprate model with $u = 8$, $t = 1$, $t' = 0.3$, and $t'' = 0.2$, and $\Delta E = 0.51\%L^2$.

D. Two-orbital Pnictide model

We consider the two-orbital Pnictide model from [23] defined by

$$H = t_1 \left[\sum_{\langle \mathbf{i}, \mathbf{j} \rangle_{e_y}, \sigma} a_{\mathbf{i}, \sigma, x}^\dagger a_{\mathbf{i}, \sigma, x} + \sum_{\langle \mathbf{i}, \mathbf{j} \rangle_{e_x}, \sigma} a_{\mathbf{i}, \sigma, y}^\dagger a_{\mathbf{i}, \sigma, y} \right] + t_2 \left[\sum_{\langle \mathbf{i}, \mathbf{j} \rangle_{e_x}, \sigma} a_{\mathbf{i}, \sigma, x}^\dagger a_{\mathbf{i}, \sigma, x} + \sum_{\langle \mathbf{i}, \mathbf{j} \rangle_{e_y}, \sigma} a_{\mathbf{i}, \sigma, y}^\dagger a_{\mathbf{i}, \sigma, y} \right] \\ + t_3 \sum_{\langle \langle \mathbf{i}, \mathbf{j} \rangle \rangle, \sigma, d} a_{\mathbf{i}, \sigma, d}^\dagger a_{\mathbf{i}, \sigma, d} + t_4 \left[\sum_{\langle \langle \mathbf{i}, \mathbf{j} \rangle \rangle_{e_x + e_y}, \sigma, d \neq d'} a_{\mathbf{i}, \sigma, d}^\dagger a_{\mathbf{i}, \sigma, d'} - \sum_{\langle \langle \mathbf{i}, \mathbf{j} \rangle \rangle_{e_x - e_y}, \sigma, d \neq d'} a_{\mathbf{i}, \sigma, d}^\dagger a_{\mathbf{i}, \sigma, d'} \right] \\ + u \sum_{\mathbf{i}, \sigma \neq \sigma', d} \left(n_{\mathbf{i}, \sigma, d} - \frac{1}{2} \right) \left(n_{\mathbf{i}, \sigma', d} - \frac{1}{2} \right) + v \sum_{\mathbf{i}, \sigma, \sigma', d \neq d'} \left(n_{\mathbf{i}, \sigma, d} - \frac{1}{2} \right) \left(n_{\mathbf{i}, \sigma', d'} - \frac{1}{2} \right), \quad (36)$$

where $\langle \mathbf{i}, \mathbf{j} \rangle_{e_x/e_y}$ denotes nearest neighbors in the e_x/e_y direction, $\langle \langle \mathbf{i}, \mathbf{j} \rangle \rangle_{e_x \pm e_y}$ denotes the next-nearest neighbors in the $e_x \pm e_y$ direction, $d \in \{x, y\}$ labels the fermionic orbitals, and u and v are the strengths of the intra- and

inter-orbital on-site Coulomb interaction. We do not consider other weaker types of on-site interaction, such as Hund's rule coupling and pair hopping that are over an order of magnitude weaker than Coulomb interactions [23, 24]. After JW transformation, the Hamiltonian becomes

$$\begin{aligned}
H = \sum_{P \in \{X, Y\}} \left\{ \frac{t_1}{2} \left[\sum_{\langle i, j \rangle_{e_y, \sigma}} P_{i, \sigma, x} \vec{Z} P_{i, \sigma, x} \sum_{\langle i, j \rangle_{e_x, \sigma}} P_{i, \sigma, y} \vec{Z} P_{i, \sigma, y} \right] + \frac{t_2}{2} \left[\sum_{\langle i, j \rangle_{e_x, \sigma}} P_{i, \sigma, x} \vec{Z} P_{i, \sigma, x} + \sum_{\langle i, j \rangle_{e_y, \sigma}} P_{i, \sigma, y} \vec{Z} P_{i, \sigma, y} \right] \right. \\
+ \frac{t_3}{2} \sum_{\langle \langle i, j \rangle \rangle, \sigma, d} P_{i, \sigma, d} \vec{Z} P_{i, \sigma, d} + \frac{t_4}{2} \left[\sum_{\langle \langle i, j \rangle \rangle_{e_x + e_y, \sigma, d \neq d'}} P_{i, \sigma, d} \vec{Z} P_{i, \sigma, d'} - \sum_{\langle \langle i, j \rangle \rangle_{e_x - e_y, \sigma, d \neq d'}} P_{i, \sigma, d} \vec{Z} P_{i, \sigma, d'} \right] \Big\} \\
+ \frac{u}{4} \sum_{i, \sigma \neq \sigma', d} Z_{i, \sigma, d} Z_{i, \sigma', d} + \frac{v}{4} \sum_{i, \sigma, \sigma', d \neq d'} Z_{i, \sigma, d} Z_{i, \sigma', d'}. \quad (37)
\end{aligned}$$

Note that while fermions with opposite spins at the same site are adjacent to each other along the JW path, fermions with different d 's, i.e., orbitals, are not. For a given d -value, as stated above, we choose the JW path to be a serpentine path, where spin is the fastest-changing variable; we join the end of the path for $d = x$ (represented by $d = 0$ on a quantum computer) with the beginning of the path for $d = y$ (represented by $d = 1$ on a quantum computer) to obtain the full JW path. So, d is the slowest changing variable.

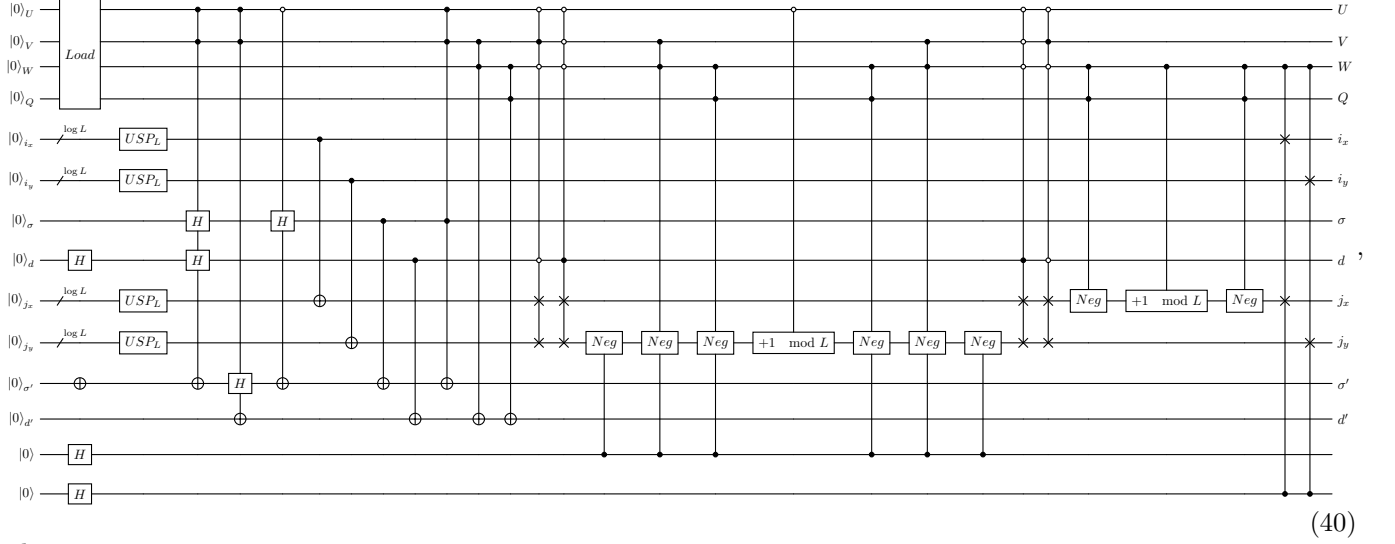
The PREP operation performs the transformation:

$$\begin{aligned}
& \text{PREP} |0\rangle \\
&= \sum_{i_x, i_y=0}^{L-1} \left\{ \sqrt{\frac{v}{4\lambda}} \sum_{\sigma, \sigma' \in \{\uparrow, \downarrow\}} |1\rangle_U |1\rangle_V |0\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle |\sigma\rangle |0\rangle_d |i_x\rangle |i_y\rangle |\sigma'\rangle |1\rangle_{d'} \right. \\
&+ \sqrt{\frac{u}{4\lambda}} \sum_{d \in \{x, y\}} |1\rangle_U |0\rangle_V |0\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle |0\rangle_\sigma |d\rangle |i_x\rangle |i_y\rangle |1\rangle_{\sigma'} |d\rangle \\
&+ \sqrt{\frac{t_1}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |0\rangle_V |0\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle (|0\rangle_d |\sigma\rangle |i_x\rangle |i_y + 1\rangle |0\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y\rangle |1\rangle_{d'} |\sigma\rangle) \\
&+ |0\rangle_d |\sigma\rangle |i_x\rangle |i_y - 1\rangle |0\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y\rangle |1\rangle_{d'} |\sigma\rangle) \\
&+ \sqrt{\frac{t_2}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} |0\rangle_U |1\rangle_V |0\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle (|0\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y\rangle |0\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x\rangle |i_y + 1\rangle |1\rangle_{d'} |\sigma\rangle) \\
&+ |0\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y\rangle |0\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x\rangle |i_y - 1\rangle |1\rangle_{d'} |\sigma\rangle) \\
&+ \sqrt{\frac{t_3}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{d \in \{x, y\}} |0\rangle_U |0\rangle_V |1\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle |d\rangle |\sigma\rangle (|i_x + 1\rangle |i_y + 1\rangle + |i_x + 1\rangle |i_y - 1\rangle) \\
&+ |i_x - 1\rangle |i_y - 1\rangle + |i_x - 1\rangle |i_y + 1\rangle) |d\rangle |\sigma\rangle) \\
&+ \sqrt{\frac{t_4}{2\lambda}} \sum_{\sigma \in \{\uparrow, \downarrow\}} \left[|0\rangle_U |1\rangle_V |1\rangle_W |0\rangle_Q |i_x\rangle |i_y\rangle (|0\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y + 1\rangle |1\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y + 1\rangle |0\rangle_{d'} |\sigma\rangle) \right. \\
&+ |0\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y - 1\rangle |1\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y - 1\rangle |0\rangle_{d'} |\sigma\rangle) \\
&+ |0\rangle_U |0\rangle_V |1\rangle_W |1\rangle_Q |i_x\rangle |i_y\rangle (|0\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y - 1\rangle |1\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x + 1\rangle |i_y - 1\rangle |0\rangle_{d'} |\sigma\rangle + \\
&\left. |0\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y + 1\rangle |1\rangle_{d'} |\sigma\rangle + |1\rangle_d |\sigma\rangle |i_x - 1\rangle |i_y + 1\rangle |0\rangle_{d'} |\sigma\rangle) \right] \Big\}, \quad (38)
\end{aligned}$$

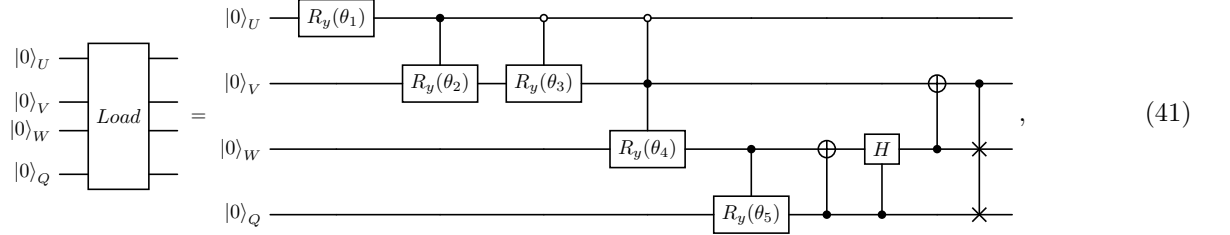
where

$$\lambda = L^2 \left[4(t_1 + t_2) + 8(t_3 + t_4) + \frac{u}{2} + v \right]. \quad (39)$$

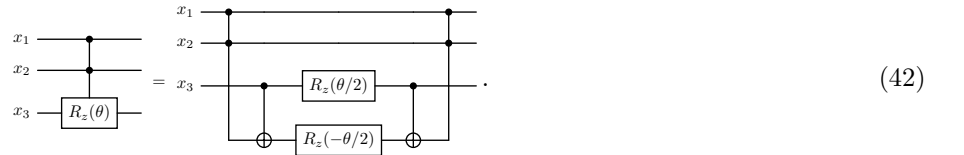
Its circuit implementation is:



where



where $\theta_1 = 2 \arcsin \sqrt{\frac{L^2(u+v)}{2\lambda}}$, $\theta_2 = 2 \arcsin \sqrt{(1 + \frac{v}{u})^{-1}}$, $\theta_3 = 2 \arccos \sqrt{(1 + \frac{t_1}{t_2} + \frac{t_1}{2t_3} + \frac{t_1}{2t_4})^{-1}}$, $\theta_4 = 2 \arccos \sqrt{(1 + \frac{t_2}{2t_3} + \frac{t_2}{2t_4})^{-1}}$, and $\theta_5 = 2 \arccos \sqrt{(1 + \frac{t_3}{t_4})^{-1}}$. Note that we apply (32) to compile the multi-controlled swap gates and negation operations that conjugate the controlled incrementers. For other multi-controlled gates, we store the logical conjunctions in clean ancilla qubits, whenever they can be used in future operations; these logical conjunctions are uncomputed when they are no longer needed for future operations, using only Clifford gates and measurements [15]. For example, $U \wedge V$ is computed using a Toffoli into a clean ancilla qubit; it is then XOR'ed into the target qubits in the decompositions of the Hadamard and NOT gates controlled by $|U\rangle$ and $|V\rangle$, and its logical AND with $|\sigma\rangle$ is computed into $|\sigma'\rangle$ for the NOT gate controlled by $|U\rangle$, $|V\rangle$, and $|\sigma\rangle$. Moreover, we compile the multi-controlled rotation gate using the identity [25]

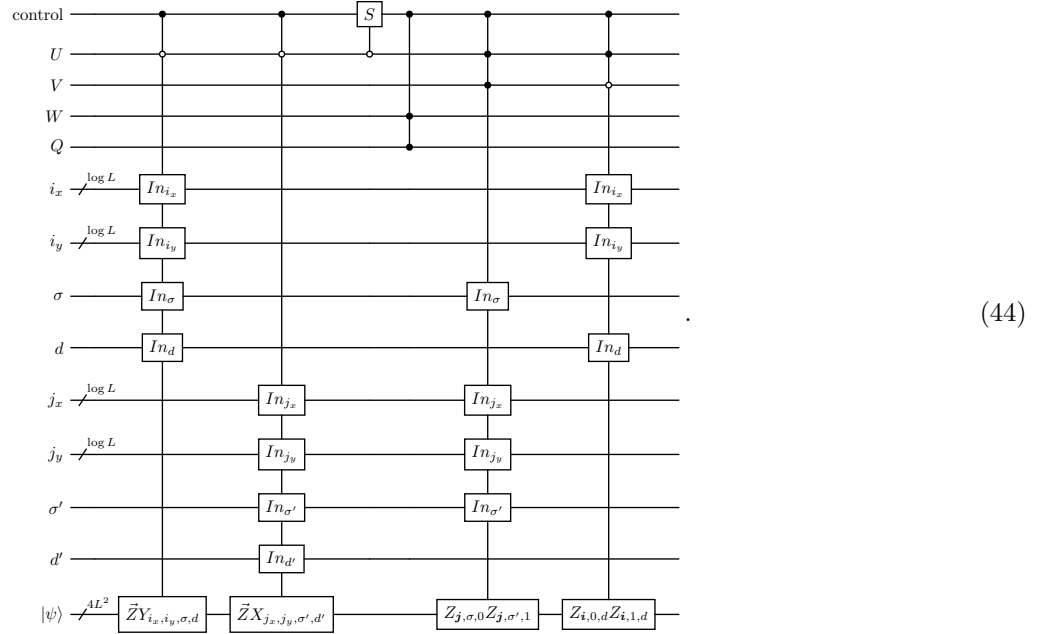


The SEL operation is defined by

$$\text{SEL} |U, V, W, Q, i_x, i_y, \sigma, d, j_x, j_y, \sigma', d'\rangle |\psi\rangle$$

$$= |U, V, W, Q, i_x, i_y, \sigma, d, j_x, j_y, \sigma', d'\rangle \otimes \begin{cases} Z_{i,\sigma,d} Z_{j,\sigma',d'} |\psi\rangle & U \wedge (i = j) \\ X_{i,\sigma,d} \bar{Z} X_{j,\sigma',d'} |\psi\rangle & \neg U \wedge \neg(W \wedge Q) \wedge (M(d, i) < M(d', j)) \wedge (\sigma = \sigma') \\ Y_{i,\sigma,d} \bar{Z} Y_{j,\sigma',d'} |\psi\rangle & \neg U \wedge \neg(W \wedge Q) \wedge (M(d, i) > M(d', j)) \wedge (\sigma = \sigma') \\ -X_{i,\sigma,d} \bar{Z} X_{j,\sigma',d'} |\psi\rangle & W \wedge Q \wedge (M(d, i) < M(d', j)) \wedge (\sigma = \sigma') \\ -Y_{i,\sigma,d} \bar{Z} Y_{j,\sigma',d'} |\psi\rangle & W \wedge Q \wedge (M(d, i) > M(d', j)) \wedge (\sigma = \sigma') \\ \text{Undefined} & \text{otherwise,} \end{cases} \quad (43)$$

where $M(d, \mathbf{i}) = i_x + i_y L + dL^2$. Note that $\vec{Z}$ in the first case reduces to an identity when $d = d'$. The controlled SEL circuit is:



The non-Clifford gate counts per PREP are 9 R_y gates, 10 T gates, and $12 + 8\lceil \log_2 L \rceil$ Toffoli gates, as well as the non-Clifford gates required by the USP_L 's. The non-Clifford gate counts per SEL are $14L^2 + 3$ Toffoli gates and 2 T gates. The total qubit count is given by

$$\left\lceil \log_2 \left(\frac{\pi \lambda L^6}{2\sqrt{x} \Delta E} \right) \right\rceil + 4L^2 + 10. \quad (45)$$

The T and Toffoli counts are

$$N_T = \begin{cases} \frac{\pi \lambda}{\sqrt{x} \Delta E} \left(22 + 18 \cdot (0.53 \log_2 \left(\frac{18\pi \lambda^2}{\sqrt{x(1-x)}(\Delta E)^2} \right) + 4.86) \right), & \text{if } L \text{ is a binary power.} \\ \frac{\pi \lambda}{\sqrt{x} \Delta E} \left(22 + 22 \cdot (0.53 \log_2 \left(\frac{22\pi \lambda^2}{\sqrt{x(1-x)}(\Delta E)^2} \right) + 4.86) \right), & \text{otherwise.} \end{cases} \quad (46)$$

$$N_{Tof} = \begin{cases} \frac{\pi \lambda}{\sqrt{x} \Delta E} (14L^2 + 12\lceil \log_2(L) \rceil + 27), & \text{if } L \text{ is a binary power.} \\ \frac{\pi \lambda}{\sqrt{x} \Delta E} (14L^2 + 12\lceil \log_2(L) \rceil + 4\lceil \log_2(m) \rceil + 27), & \text{otherwise.} \end{cases} \quad (47)$$

Once again, we numerically minimize the total Toffoli count, i.e., $N_{Tof} + N_T/2$, with respect to x , and find $x \approx 0.99$ to be near optimal. We set $t_1 = 1$, $t_2 = 1.3$, $t_3 = 0.85$, and $t_4 = 0.85$, following [23]. We further choose $u = 8$ and $\Delta E = 0.51\% L^2$, assuming that $u/t_1 = 8$ is the point where this model is classically as hard to simulate as the Fermi-Hubbard model with $u/t = 8$. We believe this assumption is very lenient as the system size of this model is double that of the Fermi-Hubbard model. The total Toffoli and qubit counts are summarized in Supplementary Table 3.

Supplementary Note 2 Resource estimation for Trotter algorithms

A. Overview

We present a high-level overview of the Trotter-based energy-estimation algorithm, used here, from [26], along with our error analysis and resource estimation pipeline. Specifically, given a Hamiltonian H , this algorithm applies the quantum phase estimation algorithm from [27] to query the directionally-controlled time evolution, i.e., $|c\rangle |\psi\rangle \mapsto e^{-i(-1)^c H \tau} |c\rangle |\psi\rangle$, instead of the controlled time evolution; the number of queries N_q to reach a root-mean-squared

L	# Toffoli	# qubits
4	1.08e7	100
6	1.81e7	183
8	2.56e7	298
10	3.76e7	444
12	5.11e7	621
14	6.67e7	831
16	8.39e7	1072
18	1.05e8	1345
20	1.29e8	1650
22	1.54e8	1987
24	1.82e8	2355
26	2.12e8	2756
28	2.45e8	3189
30	2.79e8	3653
32	3.16e8	4150

Supplementary Table 3: Resource estimates of qubitization simulations of the $L \times L$ pnictide model with $u = 8$, $t_1 = 1$, $t_2 = 1.3$, $t_3 = 0.85$, and $t_4 = 0.85$, and $\Delta E = 0.51\%L^2$.

error of $\Delta\phi$ is [26]

$$N_q \approx \frac{0.76\pi}{\Delta\phi}, \quad (48)$$

where $\Delta\phi$ translates to an energy error ΔE_{PE} via

$$\Delta\phi = \Delta E_{PE} \cdot \tau. \quad (49)$$

The time-evolution is approximated using the symmetric, second-order Trotter formula [28]

$$e^{-iH\tau} \approx U(\tau, r) \equiv \left[\left(\prod_{l=1}^{K-1} e^{-iH_l \frac{\tau}{2r}} \right) e^{-iH_K \frac{\tau}{r}} \left(\prod_{l=K-1}^1 e^{-iH_l \frac{\tau}{2r}} \right) \right]^r, \quad (50)$$

where the Hamiltonian is decomposed into an ordered sum: $H = \sum_{i=1}^K H_i$ and H_i 's are Hermitian, and r is the number of Trotter steps. Note that r is set to 1 in [8, 26]. The directionally-controlled time evolution for the Hamiltonians considered in this work only incurs an additional Clifford overhead compared to the uncontrolled time-evolution [26].

In addition to ΔE_{PE} , there are two additional sources of errors: (i) Trotter approximation error ΔE_T and (ii) rotation synthesis error ΔE_R ; note that we have expressed the errors in the same unit as ΔE_{PE} . The total error budget is given by

$$\Delta E = \Delta E_{PE} + \Delta E_T + \Delta E_R, \text{ where} \quad (51)$$

$$\Delta E_{PE} = y\Delta E, \Delta E_T = (1-s)(1-y)\Delta E, \Delta E_R = s(1-y)\Delta E, \text{ with } 0 < s, y < 1. \quad (52)$$

ΔE_T is upper-bounded by [26]

$$\Delta E_T \tau \leq \|e^{-iH\tau} - U(\tau, r)\| \equiv \delta, \text{ if } \delta \ll 1, \quad (53)$$

where $\|\cdot\|$ is the spectral norm. We bound the Trotter error δ by computing the commutator bounds [29]:

$$\delta \leq \frac{\tau^3}{12r^2} \sum_{i=1}^K \left\| \left[\left[H_i, \sum_{j=i+1}^K H_j \right], \sum_{k=i+1}^K H_k \right] \right\| + \frac{\tau^3}{24r^2} \sum_{i=1}^K \left\| \left[\left[H_i, \sum_{j=i+1}^K H_j \right], H_i \right] \right\| \equiv \frac{\tau^3 W}{r^2}. \quad (54)$$

We compute r using (52)-(54):

$$\Delta E_T \tau = \frac{\tau^3 W}{r^2} \implies r = \left\lceil \sqrt{\frac{W\tau^2}{(1-s)(1-y)\Delta E}} \right\rceil. \quad (55)$$

Finally, the gate count of the algorithm is estimated by multiplying N_q with the gate count per $U(\tau, r)$. In practice, the Toffoli count is optimized over the parameters, s, y , and τ , while keeping the total error $\leq \Delta E$ and $\delta \ll 1$.

B. Fermi-Hubbard model

Trotter scheme and error: We adopt the Trotter scheme, i.e., the ordered decomposition of H , dubbed PLAQ, and its error analysis from [8], which we present below. We assume an $L \times L$ lattice with an even L and periodic boundary conditions, and denote it by $\Lambda = (\mathbb{Z}_L)^2 = \{(0,0), (0,1), (1,0), (1,1), \dots, (L-1, L-1)\}$. We also define a sub-lattice $\Lambda' = ((2\mathbb{Z})_L)^2 = \{(0,0), (0,2), (2,0), (2,2), \dots, (L-2, L-2)\}$. Furthermore, we let $p_1 = (0,0)$, $p_2 = (1,0)$, $p_3 = (1,1)$ and $p_4 = (0,1)$. We denote a hopping operator as

$$h_{i,j,\sigma} \equiv a_{i,\sigma}^\dagger a_{j,\sigma} + a_{j,\sigma}^\dagger a_{i,\sigma}. \quad (56)$$

Then, the nearest-neighbor (nn) hopping term of the Hamiltonian is decomposed into two terms:

$$H_{nn,1} = t \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{k=1}^4 h_{i+p_k, i+p_{k+1}, \sigma}, H_{nn,2} = t \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{k=1}^4 h_{i+p_k-p_3, i+p_{k+1}-p_3, \sigma}, \quad (57)$$

where $H_{nn,1}$ and $H_{nn,2}$ are equivalent up to a lattice translation by $(1,1)$. The on-site Coulomb term, given by

$$H_c = u \sum_{i \in \Lambda} \left(n_{i,\uparrow} - \frac{1}{2} \right) \left(n_{i,\downarrow} - \frac{1}{2} \right), \quad (58)$$

is implemented as one term. The order in which these three terms are evolved in the Trotter formula is defined by the ordered list:

$$\{H_i\}_{i=1}^3 = \{H_c, H_{nn,1}, H_{nn,2}\}. \quad (59)$$

As reported in [8], the Trotter error is bounded from above by:

$$\frac{\delta r^2}{\tau^3} \leq \frac{|ut^2 L^2|}{6} (\sqrt{5} + 8) + \frac{|u^2 t|}{24} \|H_{nn,1} + H_{nn,2}\| + \frac{3}{24} \|[[H_{nn,1}, H_{nn,2}], H_{nn,1}]\|, \quad (60)$$

where the last two terms can be efficiently computed classically because $H_{nn,1}$ and $H_{nn,2}$ are free-fermionic; we display their computed values, which we take from [8], in Supplementary Table 4.

L	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
$\ H_{nn,1} + H_{nn,2}\ /t$	24	56	100	160	230	320	410	520	650	780	930	1100	1300	1500	1700
$\ [[H_{nn,1}, H_{nn,2}], H_{nn,1}]\ /t^3$	0	110	190	300	440	630	810	1000	1300	1600	1800	2200	2500	2900	3300

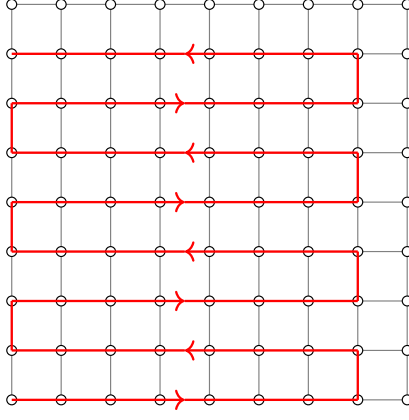
Supplementary Table 4: Numerically computed values of $\|H_{nn,1} + H_{nn,2}\|/t$ and $\|[[H_{nn,1}, H_{nn,2}], H_{nn,1}]\|/t^3$ at various values of L for an $L \times L$, periodic Fermi-Hubbard model, taken from [8].

Circuit implementation: Our circuit implementation of $e^{-iH_i\tau}$ is the same as that in [8] up to rotation synthesis, which we discuss below. We perform the same JW transformation as in the qubitization case.

On-site term $e^{-iH_c\theta}$: After JW transformation, $e^{-iH_c\theta}$ is implemented using L^2 $e^{-iZZ\theta}$ gates applied to pairs of qubits that represent fermions on the same site. Each $e^{-iZZ\theta}$ is equivalent to one $R_z(2\theta)$ gate conjugated by a pair of CNOT gates [30].

In [8], the L^2 same-angle $R_z(\theta)$ gates can be applied using Hamming-weight phasing (HWP) [15], a circuit optimization technique widely used in quantum simulations [8, 26, 31–33]. In HWP, the Hamming-weight of the to-be-rotated L^2 -qubit state is first computed into an ancilla register of size $\lfloor \log_2(L^2) \rfloor + 1$. Then, $\lfloor \log_2(L^2) \rfloor + 1$ R_z rotations $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor} R_z(2^i\theta)$ are applied to the ancilla register, before the Hamming weight is uncomputed. The Hamming weight of a binary register can be computed using $L^2 - w(L^2)$ half- and full-adders [34], where $w(L^2)$ computes the number of 1s in the binary representation of L^2 . A half- or full-adder can be implemented, up to Clifford gates, with a Toffoli gate, and can be uncomputed via a measurement-feed-forward Clifford circuit [15]. See Supplementary Figure 2 for the half- and full-adder circuits, and an example of an 8-bit Hamming-weight computation circuit. This implies that the computation and un-computation of the Hamming-weight function cost $L^2 - w(L^2)$ Toffoli gates.

We optimize HWP by implementing $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor} R_z(2^i\theta)$ using a generalized phase-gradient operation, shown in Supplementary Figure 3, instead of direct synthesis; we dub this gate-optimized HWP *catalyzed HWP*. While baseline HWP requires $L^2 - w(L^2)$, where $w(L^2)$ computes the number of 1's in the binary representation of L^2 , Toffoli gates and $\lfloor \log_2(L^2) \rfloor + 1$ R_z rotations, catalyzed HWP requires $L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1$ Toffoli gates and only one R_z ,



Supplementary Figure 1: A serpentine path for Jordan-Wigner transformation. The top-most and right-most sites are identified with the bottom-most and left-most sites due to periodic boundary conditions.

as well as a reusable catalyst state $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor} R_z(2^i \theta) |+\dots+\rangle$ whose one-time synthesis costs are amortized over the course of the simulation.

NN hopping term $e^{-iH_{nn,i}\theta}$: The implementation of $e^{-iH_{nn,1}\theta}$ and $e^{-iH_{nn,2}\theta}$ are very similar. Both $H_{nn,1}$ and $H_{nn,2}$ consist of hopping operators that act on the sites surrounding non-overlapping plaquettes. Inheriting the labeling choice made in [8], we choose to label pairwise interacting fermions around a plaquette such that their fermionic labels, defined by a fermion-to-qubit transformation, differ by 1 mod 4. However, no explicit circuit construction was provided in [8]. We accomplish this by first performing a JW transformation with a serpentine ordering shown in Supplementary Figure 1. Then, we apply an appropriate fermionic swap network, which we will describe shortly, to achieve the desired fermionic ordering.

Before getting to the fermionic swap network, we must first define a fermionic swap:

$$f_{i,j}^{swap} a_j^\dagger (f_{i,j}^{swap})^\dagger = a_i^\dagger, \quad f_{i,j}^{swap} a_j (f_{i,j}^{swap})^\dagger = a_i. \quad (61)$$

Assuming a JW transformation, a nearest-neighbor (nn) fermionic swap $f_{i,i+1}^{swap}$ can be implemented as a swap followed by a CZ to correct the phase. An operation that shifts the i th fermion to the $(i+k)$ th site can be composed using a series of nn fermionic swap, i.e.,

$$f_{i \rightarrow i+k} \equiv f_{i+k-1,i+k}^{swap} \dots f_{i+1,i+2}^{swap} f_{i,i+1}^{swap}, \quad f_{i \rightarrow i+k} a_i (f_{i \rightarrow i+k})^\dagger = a_{i+k}. \quad (62)$$

Then, we can swap the i th and $(i+k)$ th fermions by first shifting the i th fermion to the $(i+k)$ th site and then, the $(i+k-1)$ th fermion to the i th site, i.e.,

$$f_{i,i+k}^{swap} = f_{i+k-1 \rightarrow i} f_{i \rightarrow i+k} \quad (63)$$

which costs $2k-1$ nn fermionic swaps.

We begin with the swap network for $H_{nn,1}$. For a given spin and for each row of plaquettes, we label the plaquettes 1 through $p = \frac{L}{2}$, where L is the number of sites along a dimension, from left to right. We then pair up the plaquettes $(1,2), (3,4), \dots, (p-1,p)$ if p is even; if p is odd, then the last pair will be $(p-2,p-1)$ and the p th plaquette is left unpaired. Finally, for each plaquette pair, we swap the bottom left fermion of the left plaquette with the top right fermion of the right plaquette, and the bottom right fermion of the left plaquette with the top left fermion of the right plaquette. This costs $O(L^3)$ nn fermionic swaps because there are $O(L^2)$ plaquette pairs and $O(L)$ nn fermionic swaps are performed on each pair. This guarantees that the desired ordering of fermions around every plaquette. As for $H_{nn,2}$, we first map it to $H_{nn,1}$ by shifting the sites in the left-most column to the right by $L-1$ sites, and those in the bottom-most row upwards by $L-1$ sites; then, we apply the swap network for $H_{nn,1}$. Note that shifting a site upwards by a site requires $O(L^2)$ nn fermionic swaps. So, localizing the plaquettes in $H_{nn,2}$ to the desired fermionic ordering also costs $O(L^3)$ nn fermionic swaps.

Then, each of $e^{-iH_{nn,1}\theta}$ and $e^{-iH_{nn,2}\theta}$ will be transformed into $L^2/2$, where a factor of 2 accounts for the two spin sector, plaquette evolutions of the form $e^{i\theta K_i}$ with [8]

$$K_i = 2(b_i^\dagger b_i - c_i^\dagger c_i) \quad (64)$$

$$b_i = (a_{i+1} + a_{i+2} + a_{i+3} + a_{i+4})/2 \quad (65)$$

$$c_i = (a_{i+1} - a_{i+2} + a_{i+3} - a_{i+4})/2. \quad (66)$$

These unitaries act on disjoint plaquettes, indicated by the index i . Hereafter, we assume WLOG that $i = 0$. Let $F_{j,k}$ be a two-site fermionic Fourier transform:

$$F_{j,k} a_j F_{j,k}^\dagger = \frac{(a_j + a_k)}{\sqrt{2}}, \quad F_{j,k} a_k F_{j,k}^\dagger = \frac{(a_j - a_k)}{\sqrt{2}}. \quad (67)$$

Then, we define

$$V = f_{2,3}^{swap} F_{1,2} F_{3,4} f_{1,2}^{swap} F_{2,3}, \quad (68)$$

Using the fact that

$$V a_2 V^\dagger = b, \quad V a_3 V^\dagger = c, \quad (69)$$

we can realize a plaquette evolution as

$$\begin{aligned} e^{i\theta K} &= V e^{i2\theta a_2^\dagger a_2} e^{i2\theta a_3^\dagger a_3} V^\dagger \\ &= f_{2,3}^{swap} F_{1,2} F_{3,4} f_{1,2}^{swap} F_{2,3} e^{i2\theta a_2^\dagger a_2} e^{i2\theta a_3^\dagger a_3} F_{2,3} f_{1,2}^{swap} F_{3,4} F_{1,2} f_{2,3}^{swap}. \end{aligned} \quad (70)$$

Assuming a JW transformation, $F_{2,3} e^{i2\theta a_2^\dagger a_2} e^{i2\theta a_3^\dagger a_3} F_{2,3} = e^{i\theta X_2 X_3} e^{i\theta Y_2 Y_3}$ [8], which can be mapped to a single layer of two same-angle $R_z(2\theta)$ gates acting on qubits 2 and 3 using Clifford gates [25]. Each F operation requires two T gates. Thus, each $e^{i\theta K}$ costs eight T gates and two R_z gates. As a result, each of $e^{-i\theta H_{nn,1}}$ and $e^{-i\theta H_{nn,2}}$ can be implemented using $4L^2$ T gates and a layer of L^2 same-angle R_z gates, which we realize using catalyzed HWP once again.

Resource estimation: The total gate count is given by the gate count per approximated evolution $U(\tau, r)$ multiplied by the number of queries N_q to it. To construct $U(\tau, r)$, $e^{-i\theta H_c}$ is applied $r + 1$ times, $e^{-i\theta H_{nn,1}}$ is applied $2r$ times, and $e^{-i\theta H_{nn,2}}$ is applied r times, where adjacent evolutions of the same term are merged; note that each term is implemented using catalyzed HWP, the costs of which are delineated in the main text (see “Space-time trade-off in Hamming-weight phasing”), up to Clifford gates and two-site fermionic Fourier transforms. Each $e^{-i\theta H_c}$ costs 1 R_z gate and $L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1$ Toffoli gates, and each $e^{-i\theta H_{nn,1}}$ or $e^{-i\theta H_{nn,2}}$ costs 1 R_z gate, $4L^2$ T gates, and $L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1$ Toffoli gates, not including the synthesis of the catalyst states. The required catalyst states are $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor + 1} R_z(2^i \frac{u\tau}{2r}) |+\dots+\rangle$ for $e^{-i\frac{u\tau}{4r} H_c}$ and $e^{-i\frac{u\tau}{2r} H_c}$ that arises from merging two adjacent $e^{-i\frac{u\tau}{4r} H_c}$ ’s, and $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor + 1} R_z(2^i \frac{t\tau}{r}) |+\dots+\rangle$ for $e^{-i\frac{t\tau}{2r} H_{nn,1}}$, $e^{-i\frac{t\tau}{2r} H_{nn,2}}$, and $e^{-i\frac{t\tau}{r} H_{nn,2}}$ that arises from merging two adjacent $e^{-i\frac{t\tau}{2r} H_{nn,2}}$ ’s; these states cost $2\lfloor \log_2(L^2) \rfloor + 4$ R_z gates.

The Toffoli count is

$$N_{Tof} = (4r + 1)(L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1). \quad (71)$$

To get the T count, we first need to obtain the number of T gates needed to synthesize the R_z gates. To do that, we split the rotation synthesis error ΔE_R into two parts such that

$$\Delta E_R = s(1 - y)\Delta E \rightarrow (x + z)(1 - y)\Delta E, \quad (72)$$

where we have let $s = x + z$. Then, we assign one part to synthesize the R_z for the catalyst states, which leads to a T count of

$$N_{T1} = (2\lfloor \log_2(L^2) \rfloor + 3) \left(0.53 \log_2 \left(\frac{2\lfloor \log_2(L^2) \rfloor + 4}{z(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right), \quad (73)$$

where we have used the repeat-until-success rotation synthesis from [12]; the remaining part is assigned to synthesize the remaining R_z gates, which require

$$N_{T2} = (4r + 1) \left(0.53 \log_2 \left(\frac{4r + 1}{x(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right) \quad (74)$$

T gates. Including the directly applied T gates, we obtain a T count of

$$N_T = 12rL^2 + N_{T1} + N_{T2}. \quad (75)$$

Assuming $1 \text{ Toffoli} = 2 \text{ T}$ [9], the total Toffoli count is given by

$$N_{Tot} = N_q \left(N_{Tot} + \frac{N_T}{2} \right), \quad (76)$$

where we remind the readers that $N_q \approx \frac{0.76\pi}{y\tau\Delta E}$.

As in the qubitization algorithm, we set $\Delta E = 0.51\%L^2$. Then, we numerically minimize N_{Tot} over the parameters x, y, z and τ , under the constraint:

$$\tau < \left(\frac{\sqrt{2}}{W} \right)^{1/3}, \quad (77)$$

such that $\delta < \sqrt{2}/r^2$, and we observe that $r \gtrsim 10$; thus, the constraint $\delta \ll 1$ in (53) is satisfied. We find typically that $x \sim 0.01$, $\tau \sim 0.02$, $y \sim 0.6$, and $z \sim 0.001$ are optimal.

We additionally estimate the resource requirements for three other scenarios: (1) we use baseline HWP instead of catalyzed HWP, and we at most effect the same-angle R_z gates in batches of $L^2/2$ using (2) catalyzed HWP and (3) baseline HWP. The batched approaches limit the number of ancilla qubits required. For all scenarios, we set $u = 8$, $t = 1$, and $\Delta E = 0.51\%L^2$. All Toffoli count and qubit count, including ancilla qubits for QPE, RUS rotation synthesis and HWP, and catalyst qubits when necessary, are shown in Supplementary Table 5.

L	W	Catalyzed HWP		Baseline HWP		Batched, catalyzed HWP		Batched, baseline HWP	
		# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits
4	2.82e2	9.80e5	66	1.52e6	49	1.23e6	55	2.07e6	41
6	6.54e2	8.92e5	128	1.19e6	108	9.32e5	107	1.48e6	90
8	1.16e3	8.40e5	216	1.08e6	193	8.79e5	181	1.24e6	161
10	1.83e3	8.23e5	322	9.64e5	299	8.81e5	269	1.10e6	249
12	2.63e3	8.16e5	458	9.20e5	432	8.86e5	383	1.05e6	360
14	3.61e3	8.07e5	613	8.83e5	587	8.68e5	512	9.81e5	489
16	4.69e3	8.04e5	798	8.83e5	769	8.63e5	667	9.45e5	641
18	5.82e3	8.02e5	1000	8.62e5	971	8.51e5	835	9.25e5	809
20	7.21e3	8.00e5	1228	8.51e5	1199	8.54e5	1025	9.05e5	999
22	8.71e3	8.00e5	1478	8.44e5	1449	8.40e5	1233	8.89e5	1207
24	1.04e4	7.94e5	1760	8.39e5	1728	8.48e5	1469	8.87e5	1440
26	1.22e4	8.01e5	2058	8.42e5	2026	8.46e5	1717	8.85e5	1688
28	1.42e4	7.96e5	2383	8.36e5	2351	8.49e5	1988	8.77e5	1959
30	1.63e4	7.98e5	2730	8.37e5	2698	8.41e5	2277	8.78e5	2248
32	1.86e4	8.01e5	3108	8.42e5	3073	8.45e5	2593	8.86e5	2561

Supplementary Table 5: Resource estimates of Trotter simulations and the Trotter error bounds W of the $L \times L$ Fermi-Hubbard model with $u = 8$, $t = 1$, and $\Delta E = 0.51\%L^2$. Note that (i) catalyzed HWP always yields the lowest Toffoli count, (ii) batched, baseline HWP always yields the lowest qubit count, and (iii) batched, catalyzed HWP always yields similar Toffoli counts to baseline HWP, while using fewer qubits, except for when $L = 4$.

C. Single-orbital Cuprate model

Trotter scheme and error: As with the Fermi-Hubbard case, we consider an $L \times L$ lattice with periodic boundary conditions, i.e., $\Lambda = (\mathbb{Z}_L)^2$, except that, in this case, we consider only values of L that are multiples of four. We defined a new sub-lattice, $\Lambda' = ((4\mathbb{Z})_L)^2 = \{(0,0), (0,4), (4,0), (4,4), \dots, (L-4, L-4)\}$. With the coordinates p_k defined as before, we now express the Hermitian summands in the decomposition of the Hamiltonian. First, the nn terms are:

$$H_{nn,1} = t \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{l,m=0}^1 \sum_{k=1}^4 h_{i+p_k+(2l,2m), i+p_{k+1}+(2l,2m), \sigma}, \quad (78)$$

and $H_{nn,2}$ is obtained by shifting $H_{nn,1}$ by $(-1, -1)$ on the lattice. Second, the nnn terms are:

$$H_{nnn,1} = t' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{l,m=0}^1 \sum_{k=1}^2 h_{i+p_k+(2l,2m), i+p_{k+2}+(2l,2m), \sigma}, \quad (79)$$

$$H_{nnn,2} = t' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{l,m=0}^1 \sum_{k=1}^2 h_{i+p_k-(1+2l,1+2m), i+p_{k+2}-(1+2l,1+2m), \sigma}, \quad (80)$$

$$H_{nnn,3} = t' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{l,m=0}^1 \sum_{k=1}^2 h_{i+p_k+(1+2l,2m), i+p_{k+2}+(1+2l,2m), \sigma}, \quad (81)$$

$$H_{nnn,4} = t' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{l,m=0}^1 \sum_{k=1}^2 h_{i+p_k+(2l,2m-1), i+p_{k+2}+(2l,2m-1), \sigma}. \quad (82)$$

Third, the nnnn terms are

$$H_{nnnn,1} = t'' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{k=1}^4 h_{i+2p_k, i+2p_{k+1}, \sigma} + h_{i+2p_k+(0,1), i+2p_{k+1}+(0,1), \sigma} \\ + h_{i+2p_k+(1,0), i+2p_{k+1}+(1,0), \sigma} + h_{i+2p_k+(1,1), i+2p_{k+1}+(1,1), \sigma}, \quad (83)$$

$$H_{nnnn,2} = t'' \sum_{i \in \Lambda'} \sum_{\sigma} \sum_{k=1}^4 h_{i+2p_k+(2,2), i+2p_{k+1}+(2,2), \sigma} + h_{i+2p_k+(2,3), i+2p_{k+1}+(2,3), \sigma} \\ + h_{i+2p_k+(3,2), i+2p_{k+1}+(3,2), \sigma} + h_{i+2p_k+(3,3), i+2p_{k+1}+(3,3), \sigma}. \quad (84)$$

Last, the on-site Coulomb term is the same as in (58). We order these nine terms according to the ordered list:

$$\{H_i\}_{i=1}^9 = \{H_{nn,1}, H_{nn,2}, H_{nnn,1}, H_{nnn,2}, H_{nnn,3}, H_{nnn,4}, H_{nnnn,1}, H_{nnnn,2}, H_c\}. \quad (85)$$

We use the automated symbolic commutator evaluation software from [35, 36] to evaluate the Trotter error; the software is designed to automate the evaluation of commutator bounds of Fermi-Hubbard-like models. Applied to the Fermi-Hubbard model, it produces bounds with similarly sized constants as the analytical bounds from [8]. We briefly describe how the software works and refer readers to [35, 36] for further details. The software admits Hermitian operators of the form $\sum_{i \in \Lambda'} A_i$, where A_i are local operators composing of either hopping operators or number operators and Λ' is a sublattice of Λ on which A_i is defined. Each commutator will be of the form

$$\left[\sum_{i \in \Lambda'} A_i, \sum_{i' \in \Lambda'} B_{i'} \right] = \sum_{i \in \Lambda'} \left[A_i, \sum_{j \in \Lambda'} B_{i+j} \right], \quad (86)$$

where the equality is implied by the periodic boundary conditions and translational invariance of the two Hermitian operators [36]. Note that A_i or B_i could be a commutator. The software enumerates the lattice vectors j for which A_i and B_{i+j} overlap in support. Then, it evaluates the right hand side of (86) symbolically; if it evaluates to (1) quadratic/free-fermionic operators, their norms are computed exactly; if it evaluates to (2) non-quadratic operators, they are partitioned into clusters supported on no more than 14 fermionic modes, i.e., $i \in \Lambda'$, then diagonalized exactly, and finally summed up using triangle inequality. We report below the computed commutator error bound:

$$W = L^2(0.5562|t^3| + 3.5166|t^2t'| + 1.0147|t^2t''| + 1.2652|t^2u| + 5.9063|tt'^2| + 6.6727|tt't''| \\ + 2.7246|tt'u| + 1.4832|tt''^2| + 2.4294|tt''u| + 0.2018|tu^2| + 4.2510|t'^3| + 5.7898|t'^2t''| \\ + 2.2182|t'^2u| + 4.2787|t't''^2| + 2.8980|t't''u| + 0.3333|t'u^2| + 0.7688|t''^3| + 1.3761|t''^2u| + 0.2369|t''u^2|), \quad (87)$$

where we remind readers that $\delta \leq \tau^3 W / r^2$ as per (54).

Circuit implementation: The implementation of the on-site and nn hopping terms are the same way as in the Fermi-Hubbard model. In what follows, we describe how the nnn and nnnn hopping terms are implemented. We perform the same JW transformation as in the qubitization case.

NNN hopping term $e^{-iH_{nnn,i}\theta}$: First, we reorder the fermions in $H_{nnn,i}$ such that the labels of pairwise interacting fermions differ by 1 as follows: Each $H_{nnn,i}$ consists of hopping operators acting on diagonal pairs of fermions on

non-overlapping plaquettes. For each such plaquette, we swap the top right and bottom right fermions using the fermionic swap operation defined in (63); there are $O(L^2)$ plaquettes and each swap consists of $O(L)$ nn fermionic swaps, so $O(L^3)$ nn fermionic swaps are needed. Then, under JW transformation, all hopping operators will be of the form $e^{-i\theta(XX+YY)/2}$, which are mapped to a layer of same-angle $R_z(\theta)$ gates using Clifford gates. Finally, the fermionic swaps are uncomputed.

NNNN hopping term $e^{-iH_{nnnn,i}\theta}$: First, we map $H_{nnnn,i}$ to $H_{nn,i}$ using fermionic swaps as follows. We group the hopping operators in $H_{nnnn,i}$ such that each group acts on fermions on the corners of 3×3 squares. Then, we divide the lattice into 4×4 squares in a way that each square consists of four such 3×3 squares; for example, an 8×8 lattice can be divided into four such 4×4 squares. Next, for each 4×4 square, we swap the middle two sites along every side, and swap diagonally the corners of the central 2×2 plaquette. So, in total, we have applied six non-local fermionic swaps, each consisting of $O(L)$ nn fermionic swaps. There are $O(L^2)$ such 4×4 squares, which means $O(L^3)$ nn fermionic swaps are needed to map $H_{nnnn,i}$ to $H_{nn,i}$. Next, we apply the circuit implementation of the nn hopping terms, before uncomputing the fermionic swaps.

Resource estimation: The total gate count is given by the gate count per approximated evolution $U(\tau, r)$ multiplied by the number of queries N_q to it. To construct $U(\tau, r)$, $e^{-i\theta H_c}$ is applied r times, $e^{-i\theta H_{nn,i}}$'s are applied $7r + 1$ times, and $e^{-i\theta H_{nnnn,i}}$'s are applied $8r$ times where adjacent evolutions of the same term are merged; note that each term is implemented using catalyzed HWP, the costs of which are delineated in the main text (see ‘‘Space-time trade-off in Hamming-weight phasing’’), up to Clifford gates and two-site fermionic Fourier transforms. Each $e^{-i\theta H_c}$ costs 1 R_z gate and $L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1$ Toffoli gates. Each $e^{-i\theta H_{nn,i}}$ costs 1 R_z gate, $4L^2$ T gates and $L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1$ Toffoli gates, and $e^{-i\theta H_{nnnn,i}}$ costs 1 R_z gate and $2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1$ Toffoli gates, not including the synthesis costs of the catalyst states. The required catalyst states are $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor} R_z(2^i \frac{u\tau}{r}) |+\dots+\rangle$ for $e^{-i\frac{u\tau}{2r} H_c}$, $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor + 1} R_z(2^i \frac{t\tau}{r}) |+\dots+\rangle$ for $e^{-i\frac{t\tau}{2r} H_{nn,i}}$ and $e^{-i\frac{t\tau}{r} H_{nn,1}}$, $\bigotimes_{i=0}^{\lfloor \log_2(2L^2) \rfloor} R_z(2^i \frac{t'\tau}{r}) |+\dots+\rangle$ for $e^{-i\frac{t'\tau}{2r} H_{nnnn,i}}$, and $\bigotimes_{i=0}^{\lfloor \log_2(L^2) \rfloor} R_z(2^i \frac{t''\tau}{r}) |+\dots+\rangle$ for $e^{-i\frac{t''\tau}{2r} H_{nnnn,i}}$. These states cost $3\lfloor \log_2(L^2) \rfloor + \lfloor \log_2(2L^2) \rfloor + 5$ R_z gates. The Toffoli count is

$$N_{Tof} = (8r + 1)(L^2 + \lfloor \log_2(L^2) \rfloor - w(L^2) + 1) + 8r(2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1). \quad (88)$$

To get the T count, we first need to obtain the number of T gates needed to synthesize the R_z gates. To do that, we split the rotation synthesis error ΔE_R into two parts such that

$$\Delta E_R = s(1 - y)\Delta E \rightarrow (x + z)(1 - y)\Delta E, \quad (89)$$

where we have let $s = x + z$. Then, we assign one part to synthesize the R_z for the catalyst states, which leads to a T count of

$$N_{T1} = (3\lfloor \log_2(L^2) \rfloor + \lfloor \log_2(2L^2) \rfloor + 5) \left(0.53 \log_2 \left(\frac{3\lfloor \log_2(L^2) \rfloor + \lfloor \log_2(2L^2) \rfloor + 5}{z(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right), \quad (90)$$

where we have used the repeat-until-success rotation synthesis from [12]; the remaining part is assigned to synthesize the remaining R_z gates, which require

$$N_{T2} = (16r + 1) \left(0.53 \log_2 \left(\frac{16r + 1}{x(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right) \quad (91)$$

T gates. Including the directly applied T gates, we obtain a T count of

$$N_T = 4L^2(7r + 1) + N_{T1} + N_{T2}. \quad (92)$$

Assuming 1 Toffoli = 2 T, the total Toffoli count is given by

$$N_{Tot} = N_q \left(N_{Tof} + \frac{N_T}{2} \right), \quad (93)$$

where we remind the readers that $N_q \approx \frac{0.76\pi}{y\tau\Delta E}$.

As in the qubitization algorithm, we set $\Delta E = 0.51\%L^2$. Then, we numerically minimize N_{Tot} over the parameters x, y, z and τ , under the constraint:

$$\tau < \left(\frac{\sqrt{2}}{W} \right)^{1/3}, \quad (94)$$

such that $\delta < \sqrt{2}/r^2$, and we observe that $r \gtrsim 10$; thus, the constraint $\delta \ll 1$ in (53) is satisfied. We find typically that $x \sim 0.01$, $z \sim 0.05$, $y \sim 0.6$, and $0.006 \lesssim \tau \lesssim 0.1$ are optimal.

We additionally estimate the resource requirements for three other scenarios: (1) we use baseline HWP instead of catalyzed HWP, and we at most effect the same-angle R_z gates in batches of $L^2/2$ using (2) catalyzed HWP and (3) baseline HWP. The batched approaches limit the number of ancilla qubits required. For all scenarios, we set $u = 8$, $t = 1$, $t' = 0.3$, $t'' = 0.2$, and $\Delta E = 0.51\%L^2$, as in the qubitization case. All Toffoli count and qubit count, including ancilla qubits for QPE, RUS rotation synthesis and HWP, and catalyst qubits when necessary, are shown in Supplementary Table 6.

The Toffoli count of the catalyzed HWP implementation is roughly an order of magnitude higher than that of the FH simulation at the same lattice size, which can be explained as follows: For an $L \times L$ cuprate simulation, the Toffoli count, at leading order, is $24rL^2$ according to (88), where r scales as $\sqrt{W}$ according to (55) and the factor of $24L^2$ is due to the Trotter step circuit implementation. For an $L \times L$ FH simulation, the Toffoli count, at leading order, is $4rL^2$ according to (71). The values of $\sqrt{W}$ for the cuprate model, at the considered parameter setting, range from roughly 28 to 225 for $L = 4$ to 32, which are roughly 1.6 times higher than $\sqrt{W}$ for the FH model, at the considered parameter setting; along with the factor of 6 difference between the costs of a Trotter step between the two models, we obtain roughly the order of magnitude difference between the Toffoli counts of FH and cuprate simulations. Furthermore, we observe that the Trotter step implementation has a ~ 3 times higher impact on the Toffoli count than the Trotter error bound. A similar analysis holds for other less Toffoli-efficient Trotter step implementations, e.g., baseline HWP, which will increase the Trotter step implementation but not affect the Trotter error.

L	W	Catalyzed HWP		Baseline HWP		Batched, catalyzed HWP		Batched, baseline HWP	
		# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits
4	7.91e2	6.38e6	93	1.19e7	65	9.08e6	62	1.94e7	41
8	3.16e3	5.25e6	295	7.08e6	257	6.07e6	192	1.01e7	161
12	7.11e3	4.91e6	619	6.02e6	576	5.30e6	396	7.55e6	360
16	1.26e4	5.22e6	1073	5.65e6	1025	5.45e6	682	6.40e6	641
20	1.98e4	5.12e6	1647	5.40e6	1599	5.28e6	1040	6.00e6	999
24	2.85e4	5.11e6	2357	5.39e6	2304	5.23e6	1486	5.87e6	1440
28	3.87e4	5.09e6	3188	5.25e6	3135	5.18e6	2005	5.59e6	1959
32	5.06e4	5.07e6	4155	5.21e6	4097	5.18e6	2612	5.51e6	2561

Supplementary Table 6: Resource estimates of Trotter simulations and the Trotter error bounds W of the $L \times L$ cuprate model with $u = 8$, $t = 1$, $t' = 0.3$, $t'' = 0.2$, and $\Delta E = 0.51\%L^2$, as well as the Trotter error bounds W at this parameter setting. Note that (i) catalyzed HWP always yields the lowest Toffoli count, (ii) batched, baseline HWP always yields the lowest qubit count, and (iii) batched, catalyzed HWP always yields similar Toffoli counts to baseline HWP, while using fewer qubits.

D. Two-orbital Pnictide model

Trotter scheme and error: Once again, we consider an $L \times L$ lattice with periodic boundary conditions and even values of L . We will use the software from [35, 36] to evaluate the Trotter error bounds. However, the software does not explicitly handle multi-orbital models. As a workaround, we embed a $(\mathbb{Z}/L)^2$ lattice with two orbitals per site onto a $\Lambda = (\mathbb{Z}/(2L))^2$ lattice, where the sites with only even and odd coordinates represent the two different orbitals, respectively, while the sites with mixed-parity coordinates are empty. Note that this lattice embedding is only to facilitate the evaluation of the commutator error bound; the JW transformation is still performed over the original lattice. We define a sub-lattice $\Lambda' = ((4\mathbb{Z})/2L)^2$, and label $p_1 = (0, 0)$, $p_2 = (1, 0)$, $p_3 = (1, 1)$ and $p_4 = (0, 1)$. We

decompose the Hamiltonian into fourteen terms, as follows:

$$H_{nn,1} = t_1 \sum_{i \in \Lambda'} \sum_{\sigma} (h_{i+p_1, i+2p_4, \sigma} + h_{i+2p_2, i+2p_3, \sigma} + h_{i+p_3, i+p_3+2p_2, \sigma} + h_{i+p_3+2p_4, i+3p_3, \sigma}), \quad (95)$$

$$H_{nn,3} = t_2 \sum_{i \in \Lambda'} \sum_{\sigma} (h_{i+p_1, i+2p_2, \sigma} + h_{i+2p_4, i+2p_3, \sigma} + h_{i+p_3, i+p_3+2p_4, \sigma} + h_{i+p_3+2p_2, i+3p_3, \sigma}), \quad (96)$$

$$H_{nnn,1} = t_3 \sum_{i \in \Lambda'} \sum_{\sigma} (h_{i+p_1, i+2p_3, \sigma} + h_{i+2p_4, i+2p_2, \sigma} + h_{i+p_3, i+3p_3, \sigma} + h_{i+p_3+2p_4, i+p_3+2p_2, \sigma}), \quad (97)$$

$$H_{nnn,5} = t_4 \sum_{i \in \Lambda'} \sum_{\sigma} (h_{i+p_1, i+3p_3, \sigma} + h_{i+p_3, i+2p_3, \sigma} - h_{i+2p_4, i+p_3+2p_2, \sigma} - h_{i+p_3+2p_4, i+2p_2, \sigma}), \quad (98)$$

$$H_{c,1} = u \sum_{i \in \Lambda'} \sum_{k=1}^4 \left(n_{i+2p_k, \uparrow} - \frac{1}{2} \right) \left(n_{i+2p_k, \downarrow} - \frac{1}{2} \right) + \left(n_{i+2p_k+p_3, \uparrow} - \frac{1}{2} \right) \left(n_{i+2p_k+p_3, \downarrow} - \frac{1}{2} \right), \quad (99)$$

$$H_{c,2} = v \sum_{i \in \Lambda'} \sum_{\sigma, \sigma'} \sum_{k=1}^4 \left(n_{i+2p_k, \sigma} - \frac{1}{2} \right) \left(n_{i+2p_k+p_3, \sigma'} - \frac{1}{2} \right) \quad (100)$$

$H_{nn,2}$ is obtained from $H_{nn,1}$ by shifting all its lattice indices by $(2, 2)$. $H_{nn,4}$ is obtained from $H_{nn,3}$ by shifting all its lattice indices by $(2, 2)$. $H_{nnn,2}$, $H_{nnn,3}$, and $H_{nnn,4}$ are obtained from $H_{nnn,1}$ by shifting all its lattice indices by $(0, 2)$, $(2, 0)$, and $(2, 2)$, respectively. $H_{nnn,6}$, $H_{nnn,7}$, and $H_{nnn,8}$ are obtained from $H_{nnn,5}$ by shifting all its lattice indices by $(0, 2)$, $(2, 0)$, and $(2, 2)$, respectively.

We order these terms according to the following ordered list:

$$\{H_i\}_{i=1}^{14} = \{H_{nn,1}, H_{nn,2}, H_{nn,3}, H_{nn,4}, H_{nnn,1}, H_{nnn,2}, \dots, H_{nnn,8}, H_{c,1}, H_{c,2}\}. \quad (101)$$

We report below the computed commutator error bound:

$$\begin{aligned} W = & L^2(0.25|t_1^3| + 1.3333|t_1^2 t_3| + 1.4524|t_1^2 t_4| + 0.3333|t_1^2 u| + 0.7233|t_1^2 v| + 0.6667|t_1 t_2 t_3| + 1.9374|t_1 t_2 t_4| \\ & + 0.3398|t_1 t_2 u| + 1.037|t_1 t_2 v| + 2.6667|t_1 t_3^2| + 5.6918|t_1 t_3 t_4| + 1.015|t_1 t_3 u| + 2.6200|t_1 t_3 v| \\ & + 4.2562|t_1 t_4^2| + 1.1301|t_1 t_4 u| + 2.8315|t_1 t_4 v| + 0.0833|t_1 u^2| + 0.2506|t_1 uv| + 3.8354|t_1 v^2| \\ & + 0.25|t_2^3| + 1.3333|t_2^2 t_3| + 1.4524|t_2^2 t_4| + 0.3333|t_2^2 u| + 0.7363|t_2^2 v| + 2.6667|t_2 t_3^2| + 5.6918|t_2 t_3 t_4| \\ & + 1.0151|t_2 t_3 u| + 2.5783|t_2 t_3 v| + 4.2562|t_2 t_4^2| + 1.1279|t_2 t_4 u| + 2.7618|t_2 t_4 v| + 0.0833|t_2 u^2| \\ & + 0.2506|t_2 uv| + 0.2397|t_2 v^2| + 2.8333|t_3^3| + 8.0|t_3^2 t_4| + 1.3333|t_3^2 u| + 2.7211|t_3^2 v| + 8.0|t_3 t_4^2| \\ & + 2.3333|t_3 t_4 u| + 4.3035|t_3 t_4 v| + 0.1667|t_3 u^2| + 0.4714|t_3 uv| + 0.4714|t_3 v^2| + 2.8333|t_4^3| \\ & + 1.3333|t_4^2 u| + 2.7135|t_4^2 v| + 0.1667|t_4 u^2| + 0.4714|t_4 uv| + 0.4714|t_4 v^2|). \end{aligned} \quad (102)$$

Circuit implementation: We perform the same JW transformation as in the qubitization case.

On-site terms $e^{-iH_{c,i}\theta}$: The intra-orbital on-site term $e^{-iH_{c,1}\theta}$ is implemented in the same way as the on-site term in the Fermi-Hubbard model. We implement inter-orbital on-site terms $e^{-iH_{c,2}\theta}$ in two batches. The first batch consists of terms acting on different spins, and the second batch consists of terms acting on the same spins. Both batches contain only ZZ operators, whose evolution is then implemented in the same way as the intra-orbital terms, i.e., catalyzed HWP plus Cliffords.

NN hopping terms $e^{-iH_{nn,i}\theta}$: The implementation for nn hopping terms from [8] cannot be applied here because the nn hopping operators, which act on a given orbital $d \in \{x, y\}$, around a plaquette have different strengths, i.e., t_1 and t_2 . Thus, we follow a strategy similar to the one used to implement nnn hopping terms in the cuprate model: apply fermionic swaps to reduce $H_{nn,i}$ into a sum of fermion pairs with labels differing by 1. Given a $H_{nn,i}$, we consider the four hopping operators of the same strength that act on a plaquette. We divide the four hopping operators into two pairs, which are distinguished by the direction in which the edges, whose ends are occupied by the fermions acted on by the operators, point in; they either point along the JW path or not; if not, then we pick an arbitrary fermion on the plaquette and swap it with the one diagonally across, using $O(L)$ nn fermionic swaps. There are $O(L^2)$ plaquettes, so we need $O(L^3)$ nn fermionic swaps in total. This maps $e^{-iH_{nn,i}\theta}$ into a product of $e^{-i(XX+YY)\theta}$ acting on distinct nn fermion pairs along the JW path; we then transform the $e^{-i(XX+YY)\theta}$ operators into a layer of same-angle R_z gates, which we implement using catalyzed HWP.

NNN hopping terms $e^{-iH_{nnn,i}\theta}$: Once again, adopt the same strategy as in the implementation of nnn hopping terms in the cuprate model. We consider a $H_{nnn,i}$ for a fixed i and a plaquette. We focus first on the two hopping

operators acting on diagonal pairs of the same type of orbital. Then, we use fermionic swaps, each consisting of $O(L^3)$ nn fermionic swaps, to ‘unwind’ the diagonal hopping operators into nn hopping operators along the JW path, as we have done for the cuprate nnn hopping terms. We proceed to focus on the two hopping operators acting on diagonal pairs of different types of orbital. The difference here is that the fermionic swaps, which are used to ‘unwind’ the diagonal hopping operators into nn hopping operators along the JW path, will each consist of $O(L^4)$ nn fermionic swaps. This is because the JW path is partitioned into two halves, one for each type of orbital and of length $O(L^2)$, and thus, swapping any pair of fermions of different orbital types within a plaquette will require $O(L^2)$ nn fermionic swaps. Dealing with all $O(L^2)$ plaquettes will then require $O(L^4)$ nn fermionic swaps. Once each $e^{-iH_{nnn,i}\theta}$ is boiled down to nn hopping operators, i.e., $e^{-i(XX+YY)\theta}$, along the JW path, we use Clifford gates to map them to a layer of same-angle $R_z(2\theta)$ gates that are implemented using catalyzed HWP.

Resource estimation: The total gate count is given by the gate count per approximated evolution $U(\tau, r)$ multiplied by the number of queries N_q to it. To construct $U(\tau, r)$, $e^{-i\theta H_{c,1}}$ and $e^{-i\theta H_{c,2}}$ are applied r times, $e^{-i\theta H_{nn,i}}$ ’s are applied $7r + 1$ times, and $e^{-i\theta H_{nnn,i}}$ ’s are applied $8r$ times, where adjacent evolutions of the same term are merged; note that each term is implemented using catalyzed HWP, the costs of which are delineated in the main text (see “Space-time trade-off in Hamming-weight phasing”), up to Clifford gates. Each $e^{-i\theta H_{c,1}}$ costs 1 R_z gate and $2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1$ Toffoli gates. Each $e^{-i\theta H_{c,2}}$ costs 2 R_z gates and $2(2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1)$ Toffoli gates. Each $e^{-i\theta H_{nn,i}}$ costs 1 R_z gate and $4L^2 + \lfloor \log_2(4L^2) \rfloor - w(4L^2) + 1$ Toffoli gates, and $e^{-i\theta H_{nnn,i}}$ costs 1 R_z gate and $2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1$ Toffoli gates, not including the synthesis costs of the catalyst states. The required catalyst states are $\bigotimes_{i=0}^{\lfloor \log_2(2L^2) \rfloor} R_z(2^i \frac{u\tau}{2r}) |+\dots\rangle$ for $e^{-i\frac{u\tau}{4r} H_{c,1}}$, $\bigotimes_{i=0}^{\lfloor \log_2(2L^2) \rfloor} R_z(2^i \frac{v\tau}{2r}) |+\dots\rangle$ for $e^{-i\frac{v\tau}{4r} H_{c,2}}$, $\bigotimes_{i=0}^{\lfloor \log_2(4L^2) \rfloor + 1} R_z(2^i \frac{t_1\tau}{r}) |+\dots\rangle$ for $e^{-i\frac{t_1\tau}{2r} H_{nn,1}}$ and $e^{-i\frac{t_1\tau}{r} H_{nn,1}}$, $\bigotimes_{i=0}^{\lfloor \log_2(4L^2) \rfloor} R_z(2^i \frac{t_2\tau}{r}) |+\dots\rangle$ for $e^{-i\frac{t_2\tau}{2r} H_{nn,2}}$, $\bigotimes_{i=0}^{\lfloor \log_2(2L^2) \rfloor} R_z(2^i \frac{t_3\tau}{r}) |+\dots\rangle$ for $e^{-i\frac{t_3\tau}{2r} H_{nnn,3}}$, and $\bigotimes_{i=0}^{\lfloor \log_2(2L^2) \rfloor} R_z(2^i \frac{t_4\tau}{r}) |+\dots\rangle$ for $e^{-i\frac{t_4\tau}{2r} H_{nnn,4}}$. These states cost $2\lfloor \log_2(4L^2) \rfloor + 4\lfloor \log_2(2L^2) \rfloor + 7$ R_z gates.

The Toffoli count is

$$N_{Tot} = (7r + 1)(4L^2 + \lfloor \log_2(4L^2) \rfloor - w(4L^2) + 1) + 11r(2L^2 + \lfloor \log_2(2L^2) \rfloor - w(2L^2) + 1). \quad (103)$$

To get the T count, we first need to obtain the number of T gates needed to synthesize the R_z gates. To do that, we split the rotation synthesis error ΔE_R into two parts such that

$$\Delta E_R = s(1 - y)\Delta E \rightarrow (x + z)(1 - y)\Delta E, \quad (104)$$

where we have let $s = x + z$. Then, we assign one part to synthesize the R_z for the catalyst states, which leads to a T count of

$$N_{T1} = (2\lfloor \log_2(4L^2) \rfloor + 4\lfloor \log_2(2L^2) \rfloor + 7) \left(0.53 \log_2 \left(\frac{2\lfloor \log_2(4L^2) \rfloor + 4\lfloor \log_2(2L^2) \rfloor + 7}{z(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right), \quad (105)$$

where we have used the repeat-until-success rotation synthesis from [12]; the remaining part is assigned to synthesize the remaining R_z gates, which require

$$N_{T2} = (18r + 1) \left(0.53 \log_2 \left(\frac{18r + 1}{x(1 - y)\Delta E \cdot \tau} \right) + 4.86 \right) \quad (106)$$

T gates. Including the directly applied T gates, we obtain a T count of

$$N_T = N_{T1} + N_{T2}. \quad (107)$$

Assuming 1 Toffoli = 2 T, the total Toffoli count is given by

$$N_{Tot} = N_q \left(N_{Tot} + \frac{N_T}{2} \right), \quad (108)$$

where we remind the readers that $N_q \approx \frac{0.76\pi}{y\tau\Delta E}$.

As in the qubitization algorithm, we set $\Delta E = 0.51\%L^2$. Then, we numerically minimize N_{Tot} over the parameters x, y, z and τ , under the constraint:

$$\tau < \left(\frac{\sqrt{2}}{W} \right)^{1/3}, \quad (109)$$

such that $\delta < \sqrt{2}/r^2$, and we observe that $r \gtrsim 10$; thus, the constraint $\delta \ll 1$ in (53) is satisfied. We find typically that $0.01 \lesssim x \lesssim 0.03$, $0.1 \lesssim \tau \lesssim 0.2$, $y \sim 0.6$, and $z \sim 0.005$ are optimal.

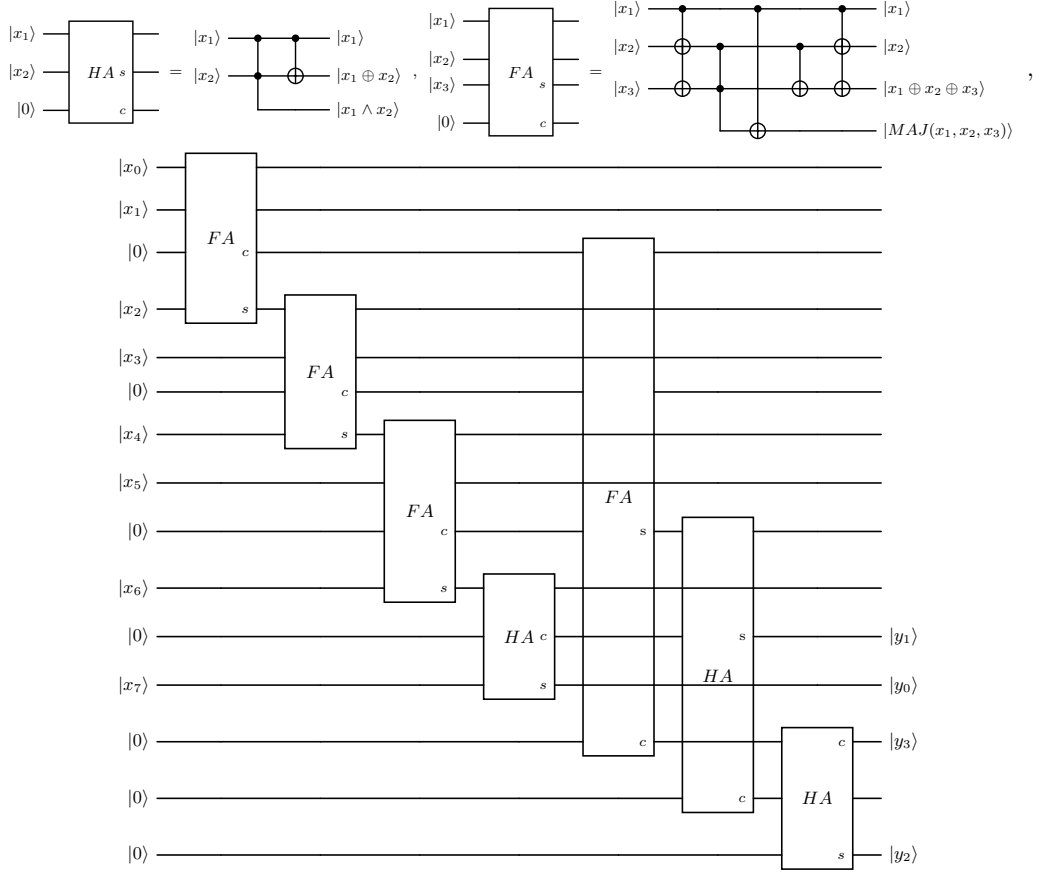
We additionally estimate the resource requirements for three other scenarios: (1) we use baseline HWP instead of catalyzed HWP, and we at most effect the same-angle R_z gates in batches of $L^2/2$ using (2) catalyzed HWP and (3) baseline HWP. The batched approaches limit the number of ancilla qubits required. For all scenarios, we set $u = 8$, $t_1 = 1$, $t_2 = 0.85$, $t_3 = 0.85$, $t_4 = 0.85$, and $\Delta E = 0.51\%L^2$, as in the qubitization case. All Toffoli count and qubit count, including ancilla qubits for QPE, RUS rotation synthesis and HWP, and catalyst qubits when necessary, are shown in Supplementary Table 7.

The Toffoli count of the catalyzed HWP implementation is roughly 70 times higher than that of the FH simulation at the same lattice size, which can be explained as follows: For an $L \times L$ pnictide simulation, the Toffoli count, at leading order, is $50rL^2$ according to (103), where r scales as $\sqrt{W}$ according to (55) and the factor of $50L^2$ is due to the Trotter step circuit implementation. For an $L \times L$ FH simulation, the Toffoli count, at leading order, is $4rL^2$ according to (71). The values of $\sqrt{W}$ for the pnictide model, at the considered parameter setting, range from roughly 106 to 850 for $L = 4$ to 32, which are roughly 6 times higher than $\sqrt{W}$ for the FH model, at the considered parameter setting; along with the factor of 12.5 difference between the costs of a Trotter step between the two models, we obtain roughly the factor of 70 difference between the Toffoli counts of FH and pnictide simulations. Furthermore, we observe that the Trotter step implementation has a ~ 2 times higher impact on the Toffoli count than the Trotter error bound. A similar analysis holds for other less Toffoli-efficient Trotter step implementations, e.g., baseline HWP, which will increase the Trotter step implementation but not affect the Trotter error.

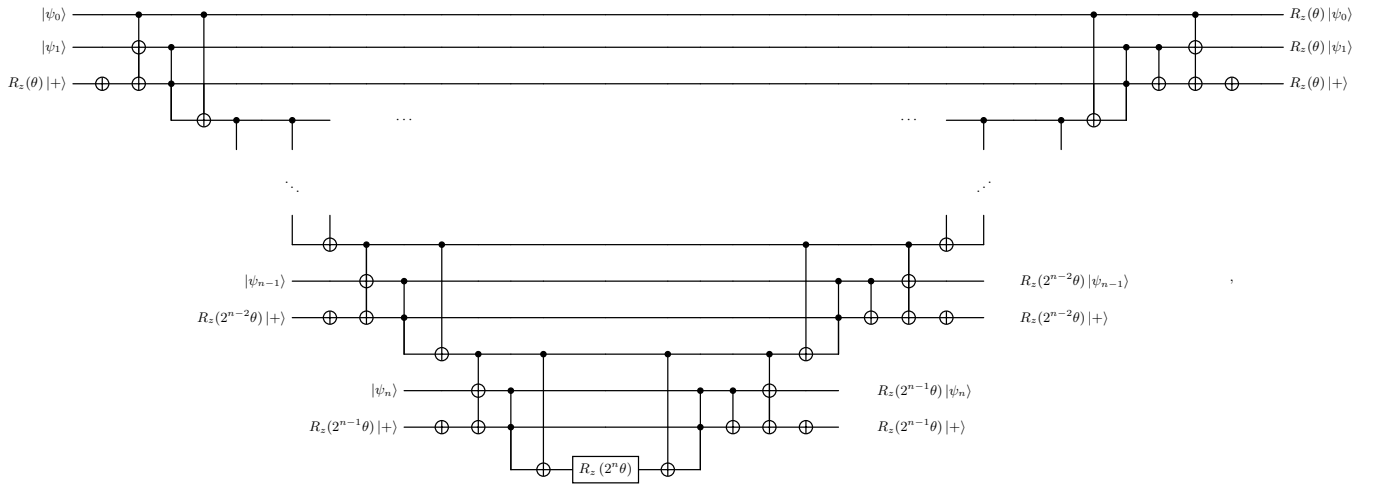
L	W	Catalyzed HWP		Baseline HWP		Batched, catalyzed HWP		Batched, baseline HWP	
		# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits	# Toffoli	# qubits
4	1.14e4	4.16e7	175	7.59e7	129	8.35e7	101	1.86e8	73
6	2.56e4	3.57e7	341	5.36e7	288	5.31e7	197	1.14e8	162
8	4.54e4	3.36e7	573	4.53e7	513	4.57e7	331	8.76e7	289
10	7.10e4	3.28e7	859	3.97e7	799	3.99e7	491	6.50e7	449
12	1.02e5	3.25e7	1219	3.78e7	1152	3.83e7	697	5.92e7	648
14	1.39e5	3.31e7	1634	3.58e7	1567	3.63e7	930	5.09e7	881
16	1.82e5	3.13e7	2123	3.56e7	2049	3.60e7	1209	4.95e7	1153
18	2.30e5	3.11e7	2665	3.42e7	2591	3.48e7	1513	4.46e7	1457
20	2.84e5	3.14e7	3273	3.35e7	3199	3.43e7	1855	4.20e7	1799
22	3.44e5	3.10e7	3943	3.31e7	3869	3.36e7	2231	3.98e7	2175
24	4.09e5	3.10e7	4689	3.28e7	4608	3.38e7	2655	3.97e7	2592
26	4.80e5	3.28e7	5487	3.27e7	5406	3.34e7	3103	3.82e7	3040
28	5.56e5	3.11e7	6352	3.25e7	6271	3.33e7	3590	3.73e7	3527
30	6.39e5	3.08e7	7279	3.25e7	7198	3.31e7	4111	3.64e7	4048
32	7.27e5	3.08e7	8281	3.23e7	8193	3.32e7	4679	3.67e7	4609

Supplementary Table 7: Resource estimates of Trotter simulations and the Trotter error bounds W of the $L \times L$ pnictide model with $u = 8$, $t_1 = 1$, $t_2 = 0.85$, $t_3 = 0.85$, $t_4 = 0.85$, and $\Delta E = 0.51\%L^2$, as well as the Trotter error bounds W , at this parameter setting. Note that (i) catalyzed HWP always yields the lowest Toffoli count, (ii) batched, baseline HWP always yields the lowest qubit count, and (iii) batched, catalyzed HWP always yields similar Toffoli counts to baseline HWP, while using fewer qubits.

Supplementary Note 3 Hamming-Weight Phasing Circuits



Supplementary Figure 2: Hamming-weight computation circuit for an 8-bit input. The HA and FA operations are half- and full-adder, respectively.



Supplementary Figure 3: Generalized phase-gradient circuit taken from equation 168 of [37], which is derived from recursive applications of the phase-catalysis circuit in figure 14 of [9]. Note that $|\psi_0\rangle$ and the gates controlled by it can be entirely removed for the catalyzed Hamming-weight phasing.

-
- [1] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, *Phys. Rev. X* **8**, 041015 (2018).
 - [2] S. Greenaway, W. Pol, and S. Sim, A case study against qsvt: assessment of quantum phase estimation improved by signal processing techniques (2024), arXiv:2404.01396 [quant-ph].
 - [3] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Phys. Rev. Lett.* **114**, 090502 (2015).
 - [4] G. H. Low and I. L. Chuang, *Quantum* **3**, 163 (2019).
 - [5] Y. Nam, Y. Su, and D. Maslov, *NPJ Quantum Information* **6**, 26 (2020).
 - [6] A. S. Holevo, *Probabilistic and statistical aspects of quantum theory*, Vol. 1 (Springer Science & Business Media, 2011).
 - [7] D. W. Berry, B. L. Higgins, S. D. Bartlett, M. W. Mitchell, G. J. Pryde, and H. M. Wiseman, *Phys. Rev. A* **80**, 052114 (2009).
 - [8] E. T. Campbell, *Quantum Science and Technology* **7**, 015007 (2021).
 - [9] C. Gidney and A. G. Fowler, *Quantum* **3**, 135 (2019).
 - [10] P. Jordan and E. Wigner, *Zeitschrift für Physik* **47**, 631 (1928).
 - [11] A. J. Bay-Smidt, F. R. Klausen, C. Sünderhauf, R. Izsák, G. C. Solomon, and N. S. Blunt, Fault-tolerant quantum simulation of generalized hubbard models (2025), arXiv:2501.10314 [quant-ph].
 - [12] V. Kliuchnikov, K. Lauter, R. Minko, A. Paetznick, and C. Petit, *Quantum* **7**, 1208 (2023).
 - [13] C. Jones, *Phys. Rev. A* **87**, 022328 (2013).
 - [14] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, *PRX Quantum* **2**, 030305 (2021).
 - [15] C. Gidney, *Quantum* **2**, 74 (2018).
 - [16] V. V. Shende, S. S. Bullock, and I. L. Markov, in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC '05 (Association for Computing Machinery, New York, NY, USA, 2005) p. 272–275.
 - [17] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, A new quantum ripple-carry addition circuit (2004), arXiv:quant-ph/0410184 [quant-ph].
 - [18] J.-Y. P. Delannoy, M. J. P. Gingras, P. C. W. Holdsworth, and A.-M. S. Tremblay, *Phys. Rev. B* **79**, 235130 (2009).
 - [19] B. Dalla Piazza, M. Mourgil, M. Guarise, H. Berger, T. Schmitt, K. J. Zhou, M. Grioni, and H. M. Rønnow, *Phys. Rev. B* **85**, 100508 (2012).
 - [20] M. Hirayama, Y. Yamaji, T. Misawa, and M. Imada, *Phys. Rev. B* **98**, 134501 (2018).
 - [21] J.-B. Morée, M. Hirayama, M. T. Schmid, Y. Yamaji, and M. Imada, *Phys. Rev. B* **106**, 235150 (2022).
 - [22] B. W. Lebert, B. Bacq-Labreuil, M. P. M. Dean, K. Ruotsalainen, A. Nicolaou, S. Huotari, I. Yamada, H. Yamamoto, M. Azuma, N. B. Brookes, F. Yakhov, H. Miao, D. Santos-Cottin, B. Lenz, S. Biermann, and M. d'Astuto, *Phys. Rev. B* **108**, 024506 (2023).
 - [23] S. Raghu, X.-L. Qi, C.-X. Liu, D. J. Scalapino, and S.-C. Zhang, *Phys. Rev. B* **77**, 220503 (2008).
 - [24] A. Moreo, M. Daghofer, J. A. Riera, and E. Dagotto, *Phys. Rev. B* **79**, 134502 (2009).
 - [25] Q. Wang, M. Li, C. Monroe, and Y. Nam, *Quantum* **5**, 509 (2021).
 - [26] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, H. Neven, and R. Babbush, *Quantum* **4**, 296 (2020).
 - [27] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, *Nature* **450**, 393 (2007).
 - [28] M. Suzuki, *Journal of Mathematical Physics* **32**, 400 (1991).
 - [29] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, *Phys. Rev. X* **11**, 011020 (2021).
 - [30] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge university press, 2010).
 - [31] Y. Nam and D. Maslov, *npj Quantum Information* **5**, 44 (2019).
 - [32] A. Kan and Y. Nam, Lattice quantum chromodynamics and electrodynamics on a universal quantum computer (2022), arXiv:2107.12769 [quant-ph].
 - [33] A. Kan and Y. Nam, *Quantum Science and Technology* **8**, 015008 (2022).
 - [34] D. E. Muller and F. P. Preparata, *Journal of the ACM (JACM)* **22**, 195 (1975).
 - [35] A. Schubert and C. B. Mendl, https://github.com/qc-tum/fermi_hubbard_commutators (2023).
 - [36] A. Schubert and C. B. Mendl, *Phys. Rev. B* **108**, 195105 (2023).
 - [37] G. Wang and A. Kan, *Quantum* **8**, 1504 (2024).