Using the fermionic swap network described in the main paper, we can also simulate Trotter steps of the 2D Hubbard model with gate depth $O(\sqrt{N})$ on a linear array of qubits. We can do this for Hubbard models with and without spin, but it is currently not clear how one might efficiently handle periodic boundary conditions with the same strategy. Below, we explain how this algorithm would work for the 2D Hubbard model with spins but note that a simple extension of the algorithm is possible for models in $d$ dimensions with gate depth $O(N^{d-1}/d)$.

The 2D Hubbard Hamiltonian with spins is

$$H = -t \sum_{\langle pq \rangle, \sigma} (a_{p,\sigma}^\dagger a_{q,\sigma} + a_{q,\sigma}^\dagger a_{p,\sigma}) + U \sum_p n_{p,\uparrow} n_{p,\downarrow}$$

where $\langle pq \rangle$ indicates that the sum should be taken over all pairs of spin-orbitals $(p, \sigma)$ and $(q, \sigma)$ which are adjacent on the 2D Hubbard lattice. The Hubbard Hamiltonian is a special case of the general electronic structure Hamiltonian where many of the terms are zero; whereas the general electronic structure Hamiltonian has $O(N^2)$ terms, the Hubbard Hamiltonian has only $O(N)$ terms. The first step in our procedure will be to use the Jordan-Wigner transformation to map Eq. (1) to a qubit Hamiltonian. One needs to choose a particular ordering of the orbitals for the Jordan-Wigner transformation in order for our technique to work. The ordering we choose is explained in Figure 1.

With the term ordering depicted in Figure 1, terms are arranged so that we may immediately simulate all of the $n_p n_q$ terms. The difficult part of this simulation is the hopping terms $a_{p,\sigma}^\dagger a_{q,\sigma} + a_{q,\sigma}^\dagger a_{p,\sigma}$. With the ordering of Figure 1, one can also immediately simulate half of the horizontal hopping terms. The final step performs a series of $O(\sqrt{N})$ layers of fermionic swaps depicted in Figure 2 which cycles all spin-orbitals through configurations in which they are adjacent to all orbitals with which they share a hopping term.

This algorithm would appear to be the most efficient strategy for simulating the 2D Hubbard model on a linear array of qubits. However, note that given planar qubit connectivity, there is an obvious way to implement Trotter steps of $O(1)$ depth that is readily apparent (and likely anticipated by those authors) from the techniques of [1]. However, the mapping in Ref. [1] requires doubling the number of qubits in the simulation and involves a more complicated (though still local) Hamiltonian; this constant overhead may be significant for moderate $N$. 

**FIG. 1.** Depiction of the mapping of Hubbard sites to a linear qubit chain. The circles each represent a spin-orbital. As labeled, red circles contain spin-up orbitals and blue circles contain spin-down orbitals. In the Hubbard Hamiltonian, the on-site interaction gives a diagonal couplings between the two spin-orbitals within each spatial orbital (e.g. $n_{3,\uparrow} n_{3,\downarrow}$) and the hopping terms are off-diagonal between adjacent spatial orbitals of the same spin (e.g. $a_{5,\downarrow}^\dagger a_{6,\downarrow} + a_{6,\downarrow}^\dagger a_{5,\downarrow}$). The arrows between the circles indicate the canonical ordering that should be used in the Jordan-Wigner transformation. The general pattern here is that we alternate whether the up or down spin-orbital comes first across the rows, and we alternate whether to order in ascending or descending order across columns.
STATE PREPARATION BY GIVENS ROTATION

Here, we provide a pedagogical explanation of the strategy based on Givens rotations discussed in the main text. We will show that one can implement any $2^N \times 2^N$ unitary operator of the form

$$U(u) = \exp \left( \sum_{pq} \log u_{pq} \left( a_p^\dagger a_q - a_q^\dagger a_p \right) \right) \quad (2)$$

where $\log u_{pq}$ is the $(p,q)$ element of the $N \times N$ matrix $\log u$, with a sequence of exactly $\binom{N}{2}$ rotations of the form

$$R_{pq}(\theta) = \exp \left[ \theta_{pq} \left( a_p^\dagger a_q - a_q^\dagger a_p \right) \right]. \quad (3)$$

Notice that $R_{pq}(\theta)$ is a special case of the basis transformation unitary $U(u)$ from Eq. (2) which occurs when $U(R_{pq}(\theta)) = R_{pq}(\theta)$. By the definition of the matrix logarithm we see that

$$r_{pq}(\theta) = \begin{pmatrix}
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\cdot & \cdots & \cdot & \cdots & \cdot & \cdots & \cdot \\
0 & \cdots & \cos(\theta) & \cdots & -\sin(\theta) & \cdots & 0 \\
\cdot & \cdots & \cdot & \cdots & \cdot & \cdots & \cdot \\
0 & \cdots & \sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\
\cdot & \cdots & \cdot & \cdots & \cdot & \cdots & \cdot \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}. \quad (4)$$

The cosine terms appear in the $p^{th}$ and $q^{th}$ entries along the diagonal, and the positive (negative) sine term appears at the intersections of row $p$ ($q$) and column $q$ ($p$). We see then that $R_{pq}(\theta)$ represents a $2^N \times 2^N$ matrix whereas $r_{pq}(\theta)$ represents an $N \times N$ matrix. Note that $r_{pq}(\theta)$ is a Givens rotation matrix.

Crucial to the procedure we will describe is the fact that the map $U(u)$ is a homomorphism under matrix multiplication:

$$U(u_a) \cdot U(u_b) = U(u_a \cdot u_b). \quad (5)$$

We now prove this. To construct our proof, we introduce a representation of a Slater determinant, $C$, which is a matrix whose columns hold the coefficients of the orbitals in some basis $|\phi_i\rangle$. This matrix is an element of the Grassmann algebra. It is the natural object obtained from a classical mean-field calculation that defines a Slater determinant within the specified basis. The matrix can be computed in two equivalent ways. First, the mapping from
the representation to the full space is given by the Plücker embedding \( \Phi \)

\[
|\Phi(C)\rangle = \wedge \left( \sum_j C^j_j |\phi_j\rangle \right)
\]

(6)

where \( \wedge \) denotes the Grassmann wedge product. Second, and more commonly in electronic structure and quantum mechanics, this map can be expressed conveniently in terms of second quantization as

\[
|\Phi(C)\rangle = \prod_i c_i^\dagger |\emptyset\rangle \quad c_i^\dagger = \sum_j C^j_i a_j^\dagger
\]

(7)

where \( |\Phi(C)\rangle \) is in the full Hilbert space, \( |\emptyset\rangle \) is the Fermi vacuum, and \( a_j^\dagger \) expresses the occupation of an orbital site \( |\phi_i\rangle \). We will first show that the map satisfies

\[
U(u) |\Phi(C)\rangle = |\Phi(uC)\rangle = |\Phi(\tilde{C})\rangle
\]

(8)

where we have defined \( \tilde{C} = uC \). We begin as

\[
|\Phi(\tilde{C})\rangle = U(u) \prod_i c_i^\dagger |\emptyset\rangle
\]

(9)

\[
= \prod_i U(u)c_i^\dagger U(u)^\dagger |\emptyset\rangle = \prod_i \tilde{c}_i^\dagger |\emptyset\rangle
\]

where \( \tilde{c}_i^\dagger = U(u)c_i^\dagger U(u)^\dagger \), the rotated vacuum \( |\tilde{\emptyset}\rangle = U(u)^\dagger |\emptyset\rangle = |\emptyset\rangle \) due to vanishing action on the vacuum, and we used the fact that anti-Hermitian operators generate the unitary group. To demonstrate this equality, we wish to show that

\[
\tilde{c}_i^\dagger = \sum_j \tilde{C}^j_i a_j^\dagger.
\]

(10)

Using the BCH expansion to determine \( \tilde{c}_i^\dagger \), we find

\[
\tilde{c}_i^\dagger = U(u)c_i^\dagger U(u)^\dagger = e^{\tilde{\kappa}} c_i^\dagger e^{-\tilde{\kappa}}
\]

(11)

\[
= c_i^\dagger + [\tilde{\kappa}, c_i^\dagger] + \frac{1}{2} [\tilde{\kappa}, [\tilde{\kappa}, c_i^\dagger]] + \cdots
\]

where we have defined

\[
\tilde{\kappa} = \sum_{pq}[\log u]_{pq} a_p^\dagger a_q = \sum_{pq} \kappa_{pq} a_p^\dagger a_q.
\]

(12)

Evaluating the first order term, we find that

\[
[\tilde{\kappa}, c_i^\dagger] = \left[ \sum_{pq} \kappa_{pq} a_p^\dagger a_q, \sum_j C^j_i a_j^\dagger \right]
\]

(13)

\[
= \sum_p \left[ \sum_q \kappa_{pq} C^q_j \right] a_p^\dagger;
\]

following to higher orders, we find that the effect is to define a new creation operator whose coefficients in the \( |\phi_i\rangle \) basis are \( e^{\kappa} C^i \), i.e.

\[
\tilde{c}_i^\dagger = \sum_j [uC^i]_j a_j^\dagger,
\]

(14)

which demonstrates the equality

\[
U(u) |\Phi(C)\rangle = |\Phi(\tilde{C})\rangle = |\Phi(uC)\rangle.
\]

(15)
With this equality, we find that
\[
U(u_a)U(u_b) |\Phi(C)\rangle = U(u_a) |\Phi(u_b C)\rangle = |\Phi(u_a u_b C)\rangle
\]
this yields an expansion with coefficients
\[
|\Phi(u_a u_b C)\rangle = \prod_i \tilde{c}_i^\dagger |\emptyset\rangle
\]
\[
\tilde{c}_i^\dagger = \sum_j |u_a u_b C|_j^i a_j^\dagger.
\]
From this we see that
\[
|\Phi(u_a u_b C)\rangle = U(u_a u_b) |\Phi(C)\rangle
\]
and as the representative C we chose was arbitrary, it must hold for any C within the Grassmann algebra, and thus we conclude that
\[
U(u_a)U(u_b) = U(u_a u_b)
\]
which shows the desired property.

Combining Eq. (4) and Eq. (5) brings us to the important observation
\[
R_{pq}(\theta) U(u) = U(r_{pq}(\theta) u).
\]
We will show that by applying a sequence of these rotations, one can implement $U^\dagger$ up to some trivial phases:
\[
\prod_k R_k(\theta_k) U(u) = \sum_{p=1}^N e^{i\phi_p} |p\rangle \langle p|
\]
where the index $k$ represents a particular pair $p, q$ which is applied at iteration $k$ and $e^{i\phi_p}$ is a unit phase. Given this sequence of rotations and the phases defined by $\phi_u$, we may implement $U$ by applying $\prod_p e^{i\phi_p n_p}$ (a single layer of gates) and then reversing the sequence of rotations. We explain how this sequence and these phases can be determined by focusing on how Givens rotations in the smaller space can be used to manipulate $u$. Finding the sequence of rotations in Eq. (22) is equivalent to performing the QR decomposition, which involves decomposing a square matrix into a product of an orthogonal (in our case, unitary) matrix right multiplied by an upper-triangular matrix. This upper-triangular matrix is actually diagonal, with the $p^{th}$ entry given by $e^{i\phi_p}$, as in Eq. (22).

When the Givens rotation matrix $r_{pq}(\theta)$ left multiplies the $N \times N$ unitary matrix $u$, the product is a unitary matrix with entries (assuming $p < q$)
\[
A_{ij} = [r_{pq}(\theta) u]_{ij} = \begin{cases} u_{pj} \cos \theta - u_{qj} \sin \theta & i = p \\ u_{qj} \sin \theta + u_{pj} \cos \theta & i = q \\ u_{ij} & \text{otherwise.} \end{cases}
\]
In order to diagonalize $u$ (as shown in Eq. (22)) our strategy will always be to use Givens rotations in order to rotate an element $A_{qj}$ to zero. Thus, when applying $r_{pq}(\theta)$ to the matrix $A$, we will always choose $\theta = \arctan(-A_{pj}/A_{qj})$ depending on which column $j$ we are targeting. Because each rotation only modifies two rows of $A$, it is possible to carry out this procedure in parallel to reduce the depth. We discuss an effective strategy for the order of parallel rotations in the main text.