Solving free fermion problems on a quantum computer

Maarten Stroeks [1,2,*] Daan Lenterman [3] Barbara M. Terhal [1,2] and Yaroslav Herasymenko [2,4,†]

[1]*QuTech, TU Delft, Lorentzweg 1, 2628 CJ Delft, The Netherlands*
[2]*Delft Institute of Applied Mathematics, TU Delft, 2628 CD Delft, The Netherlands*
[3]*Department of Physics, ETH Zürich, CH-8093 Zürich, Switzerland*
[4]*QuSoft and CWI, Science Park 123, 1098 XG Amsterdam, The Netherlands*

Simulating noninteracting fermion systems is a common task in computational many-body physics. In the absence of translational symmetries, modeling free fermions on N modes usually requires $\text{poly}(N)$ computational resources. While often moderate, these costs can be prohibitive in practice when large systems are considered. We present several free fermion problems that can be solved by a quantum algorithm with substantially reduced computational costs. The memory costs are exponentially improved, $\text{poly}\log(N)$. The run-time improvement, compared to the best known classical algorithms, is either exponential or significantly polynomial, depending on the geometry of the problem. The simulation of free fermion dynamics belongs to the BQP-hard complexity class (i.e., as hard as any (decision) problem that can be solved on a quantum computer). This implies (under standard assumptions) that our algorithm yields an exponential speedup for *any* classical algorithm at least for some geometries. The key technique in our algorithm is the block encoding of objects such as correlation matrices and Green's functions into a unitary. We demonstrate how such unitaries can be efficiently realized as quantum circuits, in the context of dynamics and thermal states of tight-binding Hamiltonians. The special cases of disordered and inhomogeneous lattices, as well as large nonlattice graphs, are presented in detail. Finally, we show that our simulation algorithm generalizes to other promising targets, including free-boson systems.

I. INTRODUCTION AND BACKGROUND

Quantum many-body dynamics can be naturally simulated by a quantum computer [1], enabling its applications in condensed matter and quantum chemistry. For a system of size N, standard quantum algorithms use $\text{poly}(N)$ resources for such simulations. It implies an exponential advantage over classical methods, when dealing with a generic many-body system. Such a general advantage may not hold in special cases of interest, such as the modeling of free fermions, where the best classical algorithms also have $\text{poly}(N)$ cost [2–4]. This classical efficiency has been key to many successes of computational physics, as free fermions model a variety of systems in condensed matter and quantum chemistry; they have also been used in computational strategies for solving interacting fermion systems, using mean-field (Hartree-Fock), perturbative methods or dynamical mean-field theory. Nonetheless, in the practical simulations of noninteracting fermions, even the most efficient numerical methods become too expensive for large systems. This motivates the key question of this work: can a quantum computer boost free fermion simulations

beyond what can be done classically? We answer this question in the affirmative, presenting quantum algorithms with an exponential speedup and memory compression for several free fermion problems.

To appreciate the value of such an exponential reduction, consider numerical simulations of free fermion models of materials and interfaces for quantum transport [5,6]. These can become prohibitive when involving more than $N = 10^9$ modes, which is of practical interest when simulating three-dimensional lattice models. Upon compression, a system of 10^9 sites can be described by $n = 30$ qubits. Larger systems of practical interest could still be accessed with moderately sized quantum computers. Indeed, even simulating one mole ($N \simeq 10^{24}$) of fermionic modes requires fewer than $n = 80$ qubits in compressed form. This opens the door to modeling free fermions near the thermodynamic limit—a desired but often challenging goal.

Our result is based on an understanding of the reduced classical complexity of free fermion systems. As an inspiration, we used the fact that the matchgate computations and the dynamics of free fermion problems on $N = 2^n$ modes can be simulated in compressed form, using $O(n)$ *space* on a quantum computer [7–11]. In this work, we go beyond these memory compression results to identify free fermion problems that also permit an exponentially improved, $\text{poly}(n)$ quantum *run-time*. Our key idea is to represent the relevant 2^n-sized object—such as the correlation matrix or a Green's function of a free fermion state—as a block of an n-qubit unitary. This unitary can be given as an efficient quantum circuit; we provide explicit construction methods by

—————
*Contact author: m.e.h.m.stroeks@tudelft.nl
†Contact author: yaroslav@cwi.nl

leveraging the modern quantum algorithm toolbox of block-encoding manipulations [12–16]. In particular, we show how to construct the desired unitary for free fermion states arising from time dynamics or thermal equilibrium. Given block encodings of the aforementioned objects into a circuit, we show how to accurately extract various physical quantities for a state, including the occupation number on a given site, or energy density across the entire system. We analyze the application of our methods to free fermion models on d-dimensional lattices and expander graphs. For the particular case of d-dimensional lattices, we argue that a polynomial run-time improvement can be expected, based on the comparison with best available classical algorithms. For expander graphs, the same analysis suggests an *exponential* speedup. On a more general geometry, the problem of single-particle time dynamics is BQP hard [17]—as hard as any problem that can be efficiently solved by a quantum computer. This rigorously proves that our approach offers an exponential quantum speedup at least for some geometries (as long as quantum computers can offer exponential speedups in principle). Finally, we outline the generalization of our approach to systems beyond free fermions.

Our work can be viewed as a fermionic counterpart to Ref. [17], which shows how the time dynamics of a system of coupled oscillators can be solved exponentially faster on a quantum versus a classical computer—with further applications in Ref. [18]. Compared to the alternative and recent work [19] that focuses on encoding a correlation matrix into a state, our method using block encodings has an exponential advantage in signal strength for the extraction of local observables (see Appendix A for more details).

We note that quantum algorithms for compressed simulation of *interacting* fermionic models have also been considered in, e.g., Ref. [20], where a Fermi-Hubbard model is simulated in the $O(1)$-particle subspace. By contrast, we consider N-mode systems with as many as $\Theta(N)$ particles.

II. PRELIMINARIES

Throughout this work, we set $N = 2^n$. A particle-conserving free fermion Hamiltonian H can be written as

$$H = \sum_{i=0, j=0}^{N-1, N-1} h_{ij} a_j^\dagger a_i, \tag{1}$$

with Hermitian matrix h, which we will assume to be $O(1)$ sparse (i.e., there are at most a constant number of nonzero entries in each row) and $|h_{ij}| \leqslant 1$. Here, $\{a_i^\dagger, a_j\} = \delta_{ij}, \{a_i, a_j\} = \{a_i^\dagger, a_j^\dagger\} = 0$. We denote the fermionic particle number operator as $\hat{N} = \sum_{i=0}^{N-1} a_i^\dagger a_i$, and we restrict ourselves to Hamiltonians that preserve particle number [21]. We allow for states ρ with an arbitrary number of particles $\mathrm{Tr}\,(\hat{N}\rho)$, which in general may scale with $N = 2^n$. Observe that in the case of single-particle dynamics $\mathrm{Tr}\,(\hat{N}\rho) = 1$, the fermionic nature of the system does not come into play and bosonic or fermionic dynamics are equivalent.

The Hermitian correlation matrix M of a fermionic state ρ on N modes is defined as

$$M_{ij} = \mathrm{Tr}\,(a_i^\dagger a_j \rho) \in \mathbb{C}, \tag{2}$$

and obeys $0 \leqslant M \leqslant I$, and $\mathrm{Tr}(M) = \langle \hat{N} \rangle$. M contains observable information about the fermionic state ρ: for example, M_{jj} is the mean fermion occupation number of a state ρ in the mode j. Furthermore, an expectation value of a free fermion Hamiltonian [Eq. (1)] can be expressed as $\mathrm{Tr}(H\rho) = \sum_{i,j} h_{ij} M_{ji}$. If ρ is itself free fermionic, expectation values of *interacting* Hamiltonians can also be obtained from M, using Wick's theorem.

Throughout this work, we will use $[N = 2^n]$ in a non-traditional way, namely, offset by 1: $[N] \equiv \{0, \dots, N-1\}$. We also use the standard notation $f(x) = O(g(x))$ if a function is asymptotically upper bounded by const $\cdot g(x)$, $f(x) = \Omega(g(x))$ if lower bounded, and $f(x) = \Theta(g(x))$ if both [i.e., scaling in the same way as const $\cdot g(x)$].

III. OUTLINE

In Sec. IV, we detail our objects of interest: correlation matrices for the time-evolved and thermal equilibrium states, as well as the Green's function matrix. Each of these objects carries physically meaningful information about the system and has a form $F(h)$—an explicit matrix function of h.

In Secs. V–VIII, we demonstrate how the information contained in these matrices can be efficiently extracted from a quantum computer, using the framework of so-called *block encodings*. Figure 1 provides a visual scheme, illustrating the structure of our approach. Section V explains the block-encoding framework, namely, how any $N \times N$ sized matrix A can be encoded into a block of a unitary U_A on $O(n) = O(\log N)$ qubits. We also review the basic tools to produce and manipulate such unitaries U_A, which were previously established in the literature. Given the matrix functions $F(h)$ of our interest, we will aim to produce the block-encoding $U_{F(h)}$ as compact quantum circuits.

The starting point of our circuit construction are smaller unitaries that encode h itself; these unitaries are called sparse access oracles (as the matrix h is required to be sparse). In Sec. VI, we show how to to implement the sparse access oracles as quantum circuits of size poly $\log N$. Such implementations are specific to the model of interest: we discuss the cases of d-dimensional lattice models and some expander graph geometries; we also demonstrate that quenched disorder can be introduced efficiently.

In the following Sec. VII, having implemented the sparse access oracles for h, we move to the second step of the construction—realizing the block encoding of matrix functions $F(h)$ of our interest. We detail how this can be done with quantum circuits of size that scales polynomially in $\log N$, as well as polynomially in parameters of the respective function, such as the evolution time t, the inverse temperature β, or the Green's function regularization parameter δ^{-1}.

Being able to run a circuit that realizes the block encoding of the matrix $F(h)$ is not sufficient for a successful simulation: one still needs an efficient method to extract physically relevant information from $F(h)$. Section VIII shows how this can indeed be done, using a Hadamard test and basic sampling techniques. In particular, we demonstrate that the local observables and global densities (such as the total energy density) can be accurately extracted from a block encoding
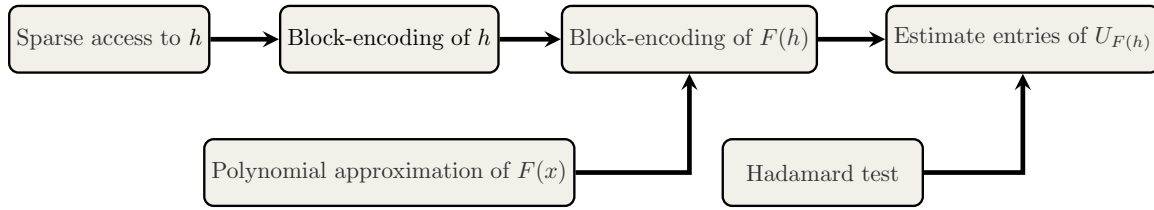
FIG. 1. Overview of the proposed quantum computational method to extract properties of free fermionic systems such as the entries of matrices listed in Sec. IV. The elements of the construction illustrated here are described in detail in Sects. V–VIII.

of $F(h)$, while maintaining the poly $\log N$ complexity of the algorithm.

Sections IX and X deal with a crucial question: does our approach provide a significant speedup compared to a classical computation? This question can be answered in the affirmative from two perspectives. In Sec. IX, we take a complexity theory perspective and observe that simulating free fermionic time dynamics using $\log N$ qubits is BQP hard. In other words, for a classical computer it is strictly as hard as simulating a general quantum computation on $\log N$ qubits—which is widely assumed to be exponentially hard in the number of qubits. This establishes that our approach yields an exponential quantum speedup for at least some system geometries. In Sec. X, we take a more practical perspective and focus on the geometries of direct physical interest (such as those given in Sec. VI). For these models, we compare the performance of our algorithm with the best classical algorithms that are currently available. In particular, we find that the quantum algorithm yields a power-$(d+1)$ polynomial speedup when simulating the time dynamics of d-dimensional lattice models. For simulations of the expander models, we demonstrate an exponential quantum speedup.

We close the main text with the Sec. XI, where we sketch how our approach can be generalized to other systems, such as free fermions with pairing terms ($\sim \Delta a_j a_k$) and free bosons with particle conservation. In Sec. XII, we discuss the future directions.

IV. OBJECTS OF INTEREST

We consider three kinds of target objects—matrix functions of h, whose entries encode the physically relevant information.

(1) Correlation matrices $M^{(\beta)}$ of thermal states $\rho_\beta = e^{-\beta H}/\mathrm{Tr}(e^{-\beta H})$ associated with free fermion Hamiltonians H:

$$M^{(\beta)} = \frac{I}{I + e^{\beta h}}. \tag{3}$$

The eigenvalues $n_\beta(\epsilon_i) = (1 + e^{\beta \epsilon_i})^{-1}$ of $M^{(\beta)}$ correspond to the Fermi-Dirac distribution, with ϵ_i the eigenenergies of h, and $\langle \hat{N} \rangle_\beta = \sum_i n_\beta(\epsilon_i)$. Note that h here includes a chemical potential term—$\mu \mathbb{I}$, if needed.

(2) Correlation matrices $M(t)$ of time-evolved states $\rho(t)$ (where the time evolution of $\rho(0)$ is under a free fermion Hamiltonian H):

$$M(t) = e^{iht} M e^{-iht}, \tag{4}$$

with M denoting the correlation matrix of $\rho(0)$.

In fact, we will consider a slightly more general object:

$$M(t_1, t_2) = e^{iht_1} M e^{-iht_2}, \tag{5}$$

the entries of which correspond to

$$M_{ij}(t_1, t_2) = \mathrm{Tr}(a_i^\dagger(t_1) a_j(t_2) \rho), \tag{6}$$

with Heisenberg operators $a_i^\dagger(t)$, $a_j(t)$ with respect to the free fermion Hamiltonian H.

Note that for a Hamiltonian $H = H_0 + V$ with free fermionic H_0 and interacting perturbation V, after applying $U(t) = e^{-iHt}$ to an initial free fermionic state ρ, observables involving creation and annihilation operators can be obtained from $M(t_1, t_2)$ in Eq. (5). This can be done via a perturbative expansion of $U(t) = e^{-iHt}$ and using Wick's theorem.

(3) The Green's function (in the frequency domain) with respect to a thermal state ρ_β of a free fermion Hamiltonian:

$$G^{(\delta, \beta, \omega)}(h) = \frac{\delta}{2} \Bigg[\left(1 - \frac{1}{1 + \exp(\beta h)} \right) \frac{1}{i\delta - (h + \omega)} + \left(\frac{1}{1 + \exp(\beta h)} \right) \frac{-1}{i\delta + (h + \omega)} \Bigg], \tag{7}$$

with $\delta > 0$ a regularization parameter.

$G^{(\delta, \beta, \omega)}(h)$ is a Fourier transform of the time-domain Green's function, the entries of which are given by [here we use time-ordering unlike in Eq. (6)]

$$G_{ij}(t_1, t_2) = \begin{cases} i\mathrm{Tr}(a_i^\dagger(t_1) a_j(t_2) \rho_\beta), & \text{for } t_1 \geqslant t_2, \\ -i\mathrm{Tr}(a_j(t_2) a_i^\dagger(t_1) \rho_\beta), & \text{for } t_1 < t_2, \end{cases}$$

$$= \begin{cases} \left(i e^{ih(t_1 - t_2)} \frac{1}{1 + \exp(\beta h)} \right)_{ij}, & \text{for } t_1 \geqslant t_2, \\ \left(-i e^{ih(t_1 - t_2)} \left(1 - \frac{1}{1 + \exp(\beta h)} \right) \right)_{ij}, & \text{for } t_1 < t_2. \end{cases} \tag{8}$$

The regularization parameter δ in Eq. (7) ensures that the Fourier transform converges in the case of an isolated system, but can also model interactions with a bath at finite temperature [22].

V. BLOCK ENCODINGS

Let us consider encoding a Hermitian $(N \times N)$-dimensional matrix A into a block of an $(n + m)$-qubit unitary U_A. In general, an n-qubit matrix A is said to be block encoded into U_A if it is equal to the block of U_A where m

qubits are in a trivial state, with some coefficient α:

$$A_{ij} = \alpha \langle i|_n \langle 0|_m U_A |j\rangle_n |0\rangle_m. \qquad (9)$$

Here, the matrix indices $i, j \in [N]$ are interpreted as bit strings of length n. The coefficient $\alpha \geqslant 1$ arises from the fact that $\|U_A\| = 1$ while A is arbitrary. If $\|A\| \leqslant 1$, we can take $\alpha = 1$. For a useful block encoding, the coefficient α should not blow up beyond poly $\log N$. Fortunately, in the applications considered in this work, α will remain a small constant. For the same reasons of maintaining efficiency, we will limit the number of ancillary qubits m to $O(\log N)$.

We will also allow block encoding with error ε, the deviation in operator norm between A and $\alpha \langle 0|_m U_A |0\rangle_m$.

Definition 1. For a matrix A on n qubits and $\alpha, \varepsilon \in \mathbb{R}_+$, an $(m + n)$-qubit unitary U_A is an (α, m, ε)-block encoding of A, if

$$\|A - \alpha (\langle 0|^{\otimes m} \otimes \mathbb{1}) U (|0\rangle^{\otimes m} \otimes \mathbb{1})\| \leqslant \varepsilon, \qquad (10)$$

where $\| \cdot \|$ is the spectral norm.

The quantum circuits that approximately block encode the matrix functions $F(h)$ are built using elementary circuits U_h that block encode h. These latter block-encoding U_h, in turn, contain unitaries that realize so-called sparse query access to h. To access an s-sparse matrix h, i.e., a matrix that has up to $s = O(1)$ nonzero entries in any row, we will use "oracle" unitaries O_r and O_a that produce the entries of h. The "row" oracle O_r returns, for a given row i, all column indices where the matrix h has nonzero entries. The "matrix entry" oracle O_a returns the value of h (given with n_a bits) for a given row and column index. This way, entries of h can be retrieved without explicit access to the $\Theta(2^n)$ nonzero entries of matrix h. Let us formally define the *oracle tuple* $\mathcal{O}_h$ of a sparse matrix h containing the row and matrix entry oracles, and also their inverses and controlled versions as follows.

Definition 2 (Sparse access oracle tuple $\mathcal{O}_h$). Sparse access for an s-sparse $2^n \times 2^n$ matrix h is defined as

$$O_r |i\rangle |0\rangle^{\otimes s(n+1)} = |i\rangle |r(i,1)\rangle |r(i,2)\rangle \cdots |r(i,s)\rangle, \quad \forall i \in [2^n],$$

$$O_a |i\rangle |j\rangle |0\rangle^{\otimes n_a} = |i\rangle |j\rangle |h_{ij}\rangle, \quad \forall i, j \in [2^n], \qquad (11)$$

where $r(i,k)$ is the index for the kth nonzero entry of the ith row of h. Let us now cover a few technicalities. O_r is a matrix acting on $(s + 1)(n + 1)$ qubits, and so the first qubit of $|i\rangle$ is in $|0\rangle$. To accommodate rows with less than s nonzero entries, one uses the following. If the ith row contains $s' < s$ nonzero entries, then the last $(s - s')(n + 1)$ qubits are put in the state $|1\rangle |k\rangle$. Note that for states $|r(i,1)\rangle \cdots |r(i,s')\rangle$, the first qubit is in $|0\rangle$. h_{ij} is the value of the (i,j)th entry of h, described by a bit string with n_a binary digits (we will assume this representation to be exact). O_a is a matrix acting on $2n + n_a$ qubits.

Furthermore, we define the controlled version of the above sparse access, consisting of

$$C\text{-}O_r = O_r \otimes |1\rangle\langle 1|_a + \mathbb{1} \otimes |0\rangle\langle 0|_a,$$

$$C\text{-}O_a = O_a \otimes |1\rangle\langle 1|_a + \mathbb{1} \otimes |0\rangle\langle 0|_a, \qquad (12)$$

where each matrix now acts on an additional (ancillary) qubit a. We call the collection of six oracles $(O_r, O_a, C\text{-}O_r, C\text{-}O_a, O_r^{-1}, O_a^{-1}, C\text{-}O_r^{-1}, C\text{-}O_a^{-1})$ the sparse access *oracle tuple* $\mathcal{O}_h$ of h.

The relation between this definition of the oracle tuple and another common definition is discussed in Appendix B for completeness.

Let us now present the following statements, relating the construction of the block encoding of h and that of polynomials of h. Note that these block-encoding constructions contain calls to oracles from the oracle tuple $\mathcal{O}_h$ in Definition 2. We shall use these results when constructing the block encodings of our desired matrix functions. The following statements use Definition 1.

Proposition 1 (Lemma 48 in Ref. [23]). A $(s, n + 3, \varepsilon_{\text{BE}_h})$-block encoding of h, U_h (and its controlled version) consists of $O(1)$ calls to oracles from $\mathcal{O}_h$ tuple, $O(n + \log^{5/2}(s^2/\varepsilon_{\text{BE}_h}))$ elementary gates, and $O(sn + n_a + \log^{5/2}(s^2/\varepsilon_{\text{BE}_h}))$ ancillary qubits. Here, n_a denotes the number of bits with which the entries of h are specified.

Proposition 2 (Theorem 31 in Ref. [14]). Let $p_d(x)$ denote a degree-d polynomial, such that $|p_d(x)| \leqslant 1/2$ for $x \in [-1, +1]$. Then, a $(1, n + 5, 4d\sqrt{\varepsilon_{\text{BE}_h}/s} + \delta)$-block encoding of $p_d(h/s)$, $U_{p_d(h/s)}$ consists of $O((n + 4)d)$ elementary gates, and at most d calls to unitaries U_h, U_h^{-1}, or controlled U_h. The classical description of this circuit can be obtained classically in poly$(d, \log(1/\delta))$ time.

VI. SPARSE ACCESS REALIZATION FOR PHYSICAL SYSTEMS

The starting point for our method is to realize the sparse access tuple $\mathcal{O}_h$ for the system Hamiltonian h, using efficient quantum circuits. In particular, we need circuit realizations for unitaries O_r and O_a [Eq. (11)]; these in fact can be given as (reversible) classical circuits, as no entanglement generation is required. Then, the controlled and inverse unitaries from $\mathcal{O}_h$ can also be obtained as efficient circuits (with a constant factor overhead), controlling or inverting the circuits for O_r and O_a gate by gate. Please note that "efficient" in our case means poly $\log N$ gate complexity, i.e., polynomial in the number of qubits rather than the size of h. In other words, simply looking up the entries of the $N \times N$ matrix h would not suffice, as that takes time that is exponentially longer than desired. Despite this difficulty, the requirement of the efficient implementation of $\mathcal{O}_h$ can be satisfied for a variety of h of interest.

A large family of free fermionic models for which the sparse access to h can be efficiently realized are d-dimensional tight-binding models. Consider a d-dimensional square lattice $\mathcal{L}$ with $L_1 \times L_2 \times \cdots \times L_d = N_s$ sites, with either periodic or open boundaries. For each site $\vec{x}$, let there be up to $N_0 = O(1)$ on-site degrees of freedom such as spin or local orbital degrees of freedom. We can thus represent each fermionic mode using $n = (\prod_{i=1}^{d} \lceil \log_2 L_i \rceil) \times \lceil \log_2 N_0 \rceil$ qubits as $|\vec{x} = (x_1, \ldots, x_d), o\rangle$, where $N_s = \Theta(2^n)$. Inside the lattice, let there be $O(1)$ nonoverlapping rectangular domains, modeling different physical regions such as leads versus bulk regions, where parameters in H can be different. We thus consider Hamiltonians of the following form:

$$H = \sum_{o_1, o_2} \sum_{\vec{x} \in \mathcal{L}, |\vec{t}|_M \leqslant l} h_{\vec{x}, o_1, \vec{x}+\vec{t}, o_2} a^\dagger_{\vec{x}+\vec{t}, o_2} a_{\vec{x}, o_1} + \text{H.c.}, \qquad (13)$$

where it is understood (but notationally awkward) that the sum over $\vec{x} \in \mathcal{L}$, $|\vec{t}|_M \leqslant l$ only counts each possible hopping term

once. In addition, we have

$$h_{\vec{x},o_1,\vec{x}+\vec{t},o_2} = g(o_1, o_2, D(\vec{x}), D(\vec{x}+\vec{t}), \vec{t}),$$

$$\left| h_{\vec{x},o_1,\vec{x}+\vec{t},o_2} \right| \leqslant 1. \tag{14}$$

Here, $|.|_M$ means Manhattan distance in the lattice; the maximal range of the interaction is posited to be constant—$l = O(1)$. The function $D(\vec{x})$ returns the domain to which $\vec{x}$ belongs: since the domains are rectangular regions, $D(\vec{x})$ can be efficiently computed using standard reversible arithmetic circuits. If $\vec{x}$ or $\vec{x}+\vec{t}$ does not belong to any domain (for example, $\vec{x}+\vec{t}$ is beyond the boundaries of the lattice), the coefficient $h_{\vec{x},o_1,\vec{x}+\vec{t},o_2} = 0$. Thus, the function g only takes in $O(1)$ information and all $O(1)$ possible nonzero outputs of g can be stored classically, using, say, $O(n_a)$ bits. To realize the oracles $\mathcal{O}_h$ from Definition 2 as poly(n)-sized quantum circuits, observe that one can efficiently generate the $O(1)$ input to g and lookup the relevant information.

Going beyond local d-dimensional models, we give an example of a model on an *expander graph* that has sparse query access. These graphs have the important property that the number of vertices that lie a distance d away from a given vertex scales exponentially in d. Free fermionic models on such graphs have been a subject of recent interest, especially in the studies of Anderson localization on random regular graphs [24–27]. In Appendix C, we provide details of the realization of $\mathcal{O}_h$ as poly(n)-sized quantum circuits for a simple example: the Margulis expander graph.

So far, we have proposed models with efficient sparse access where there was only a limited number of possible options for the hopping parameters, and they were input "by hand." This is in line with a necessary limitation—even though the system has size N, we should be unable to assign every mode an independent value of the hopping parameter.

However, this restriction can be somewhat relaxed. In particular, one can show that local quenched disorder can also be incorporated into h. This has the significance for physics application, as it allows to study Anderson localization. For simplicity, let us focus on realizing on-site disorder in a single domain D^* of a tight-binding model. This means that we introduce a single change to the Hamiltonian of Eqs. (13) and (14). Namely, if $D(\vec{x}) = D^*$ and $\vec{t} = 0$ (both equalities are efficiently checkable), the value of $h_{\vec{x},o_1,\vec{x}+\vec{t},o_2}$ will be replaced by

$$h_{\vec{x},o_1,\vec{x}+\vec{t},o_2} = \delta_{o_1,o_2} \mathrm{PRF}(\vec{x}), \tag{15}$$

where $\delta_{a,b}$ is the Kronecker symbol and PRF is a pseudo-random function of the lattice site coordinate $\vec{x}$. Note that a pseudo-random function can be realized as an efficient classical circuit [28,29]. Other models of local disorder can be realized similarly. We note that an independent work [30] discusses the application of simulating disordered free fermions in more detail.

VII. BLOCK ENCODINGS OF RELEVANT MATRIX FUNCTIONS

Given the poly(n)-effort sparse access tuple $\mathcal{O}_h$, we now aim to realize a block encoding of the desired matrix functions of h (Sec. IV) with an efficient quantum circuit. We will

approximate these functions with polynomials of sufficiently low degree, enabling us to use standard methods of block-encoding manipulation (Proposition 2).

To construct the polynomial approximations, let us first establish the following. Proposition 2 prescribes how degree-d polynomials $p_d(x)$ with $x = h/s$ can be block encoded, with s the sparsity of h. We thus require a polynomial approximation $p_d(x)$ to our functions of interest $F(h = sx)$ to be sufficiently accurate in the domain $x \in [-\|h\|/s, +\|h\|/s]$. It can be argued straightforwardly that this domain is at most $[-1, +1]$ by bounding the spectral norm of h:

Proposition 3. Let h denote an $s = O(1)$-sparse Hermitian $N \times N$ matrix with $|h_{ij}| \leqslant 1$, $\forall i, j$. The spectral norm $\|h\|/s \leqslant 1$ by the Gershgorin circle lemma (which says that every eigenvalue of h lies within at least one of the N disks $D_i = \{z \in \mathbb{C} : |z - h_{ii}| \leqslant \sum_{j \neq i} |h_{ij}|\}$).

To block encode the thermal correlation matrix in Eq. (3) and the thermal Green's function in Eq. (7), we need to approximate the functions

$$f^{(\beta)}(x) := \frac{1}{4} \frac{1}{1 + \exp(\beta s x)} \tag{16}$$

and

$$g^{(\delta,\beta,\omega)}(x) := \frac{1}{4} \frac{\delta}{2} \left[\left(1 - \frac{1}{1 + \exp(\beta s x)} \right) \frac{1}{i\delta - (sx + \omega)} \right.$$
$$\left. + \left(\frac{1}{1 + \exp(\beta s x)} \right) \frac{-1}{i\delta + (sx + \omega)} \right] \tag{17}$$

in the domain $x \in [-1, +1]$. These functions have poles in the complex plane at $z = (2k+1)i\pi/\beta$ (with $k \in \mathbb{Z}$) and at $z = (\pm i\delta - \omega)/s$, respectively. Since these poles might lie in the unit circle for general β and δ, we have to resort to polynomial approximation techniques beyond Taylor approximations to obtain a sufficiently accurate approximation for $x \in [-1, +1]$. In particular, we will employ Bernstein's theorem:

Lemma 1 ([31]). Let $f(x)$ be analytic on $[-1, +1]$ and analytically continuable to the interior of an ellipse defined by $E_r = \{\frac{1}{2}(z + z^{-1}) : |z| = r\}$ (for some real-valued $r \geqslant 1$). Furthermore, let $|f(z)| \leqslant C$ for $z \in E_r$. The error with respect to their polynomial approximation $p_d(x)$ (Chebyshev expansion truncated at degree d) can be bounded as

$$\max_{x \in [-1,+1]} |f(x) - p_d(x)| \leqslant \frac{2Cr^{-d}}{r - 1}. \tag{18}$$

Using this result, we derive the following error bounds for the polynomial approximations of Eqs. (16) and (17). The proofs of Lemmas 2 and 3 are given in Appendixes D and F.

Lemma 2 (Simplified version of Lemma D1 in Appendix D). For the function $f^{(\beta)}(x)$ in Eq. (16) (with $\beta, s \geqslant 0$), one can efficiently construct a polynomial $p_d(x)$ of degree d such that

$$\max_{x \in [-1,+1]} |f^{(\beta)}(x) - p_d(x)| \leqslant \mathrm{poly}(\beta s)/d. \tag{19}$$

Lemma 3 (Simplified version of Lemma F1 in Appendix F). For the function $g^{(\delta,\beta,\omega)}(x)$ in Eq. (17) (with $\beta, \delta, s > 0$), one

can efficiently construct a polynomial $p_d(x)$ of (even) degree d such that

$$\max_{x\in[-1,+1]}|g^{(\delta,\beta,\omega)}(x) - p_d(x)|$$
$$\leqslant (\text{poly}(\beta s) + \text{poly}(s/\delta))/d. \qquad (20)$$

Combining Lemmas 2 and 3 with Propositions 1 and 2, we directly obtain Theorems 1 and 2 below. The detailed proofs are given in Appendixes D and F. Note that—crucially, because of the factors $\frac{1}{4}$ in Eqs. (16) and (17)—the polynomials $p_d(x)$ that are block encoded obey $|p_d(x)| \leqslant 1/2$ for $x \in [-1, +1]$, provided that the error of the polynomial approximation is $O(1)$. The size of the circuits that block encode $M^{(\beta)}(h)$ in Eq. (3) and $G^{(\delta,\beta,\omega)}(h)$ in Eq. (7) is poly(n), provided that $\beta, 1/\delta = \text{poly}(n)$ and when the oracles from $\mathcal{O}_h$ are poly(n)-sized circuits (such as those in Sec. VI).

Theorem 1 [Block encoding of the thermal correlation matrix Eq. (3)]. For an s-sparse Hamiltonian h on n qubits, assume access to the oracle tuple $\mathcal{O}_h$. We denote the controlled $(1, n+5, \varepsilon)$-block encoding of $\frac{1}{4}M^{(\beta)} = \frac{1}{4}I/(I + \exp(\beta h))$ by C-$U_{M^{(\beta)}}$. The implementation of this block encoding for $\beta = \text{poly}(n)$ requires poly$(n)/\varepsilon$ calls to oracles from the oracle tuple $\mathcal{O}_h$, $O(n) + n_a + \log^{5/2}(\text{poly}(n)/\varepsilon^4)$ ancillary qubits, and $O(n) + \text{poly}(n)/\varepsilon + \log^{5/2}(\text{poly}(n)/\varepsilon^4)$ additional elementary gates. To implement this block encoding, an additional classical computing time of poly$(n/\varepsilon, \log(1/\varepsilon))$ is required.

Theorem 2 [Block encoding of the thermal Green's function Eq. (7)]. For an s-sparse Hamiltonian h on n qubits, assume access to the oracle tuple $\mathcal{O}_h$. We denote the controlled $(1, n+5, \varepsilon)$-block encoding of $\frac{1}{4} G^{(\delta,\beta,\omega)}(h)$ in Eq. (7) by C-$U_{G^{(\delta,\beta,\omega)}}$. The implementation of this block encoding for $\beta, 1/\delta = \text{poly}(n)$ requires poly$(n)/\varepsilon$ calls to oracles from the oracle tuple $\mathcal{O}_h$, $O(n) + n_a + \log^{5/2}(\text{poly}(n)/\varepsilon^4)$ ancillary qubits, and $O(n) + \text{poly}(n)/\varepsilon + \log^{5/2}(\text{poly}(n)/\varepsilon^4)$ additional elementary gates. To implement this block encoding, an additional classical computing time of poly$(n/\varepsilon, \log(1/\varepsilon))$ is required.

Next, let us focus on block encoding the time-evolved correlation matrix $M(t)$ in Eq. (4). To block encode it, we will use a block encoding of $\exp(iht)$ as a subroutine. The construction of this latter block encoding through polynomial approximations is already considered in Refs. [13,23], and we will use this construction from Ref. [23] directly. We construct a block encoding of $M(t)$ using the product of block encodings of $\exp(iht)$, an initial correlation matrix M, and $\exp(-iht)$. A detailed proof of Theorem 3 is given in Appendix E. There, we in fact consider a block encoding of the more general object $M(t_1, t_2)$ in Eq. (5).

Theorem 3 [Simplified version of Theorem E1 in Appendix E. Block encoding of the time-evolved correlation matrix in Eq. (4)]. For an s-sparse Hamiltonian h on N fermionic modes, assume access to the oracle tuple $\mathcal{O}_h$. In addition, assume access to the $(\alpha, m, \varepsilon_M)$-block-encoding U_M of a correlation matrix M of a fermionic state on N modes. The $(\alpha, 2n+m+10, \varepsilon + \varepsilon_M)$-block-encoding $U_{M(t)}$ of $M(t) = e^{iht}Me^{-iht}$ can be produced using $D(\alpha, \varepsilon, t) = O(|t| + \log(\alpha/\varepsilon))$ calls to oracles from the tuple $\mathcal{O}_h$, and a single use of the block-encoding U_M. Moreover, one uses

$O(n|t| + \log(\alpha/\varepsilon) + D(\alpha, \varepsilon, t)(n + \log^{5/2}(\alpha|t|/\varepsilon)))$ elementary gates and $O(n_a + \log^{5/2}(\alpha|t|/\varepsilon))$ ancillary qubits.

VIII. EXTRACTING OBSERVABLES

Having explicitly constructed (α, m, ε)-block-encoding $U_{F(h)}$ of our objects of interest $F(h)$, let us detail how to extract relevant observables from such block-encoding unitaries. If $U_{F(h)}$ is given as a poly(n)-sized quantum circuit, the real and imaginary parts of $F(h)_{ij}$ can be extracted efficiently using the so-called Hadamard test using an ancillary-qubit-controlled $U_{F(h)}$. Note that the circuit size required to implement controlled $U_{F(h)}$ scales the same as $U_{F(h)}$, up to a constant factor overhead. We can extract $F(h)_{ij}$ with an accuracy specified in the next Lemma 4. This Lemma is stated for a general block-encoding unitary and is proved in Appendix G. From Lemma 4, it is clear that the error up to which $F(h)_{ij}$ can be estimated is $1/\text{poly}(n)$, since we allow for at most poly(n) calls to the block-encoding unitaries.

Lemma 4. Given an n-qubit matrix A. Let C-U_A (acting on $n + m + 1$ qubits) denote the controlled version of the (α, m, ε)-block-encoding U_A of A. An estimate $\hat{A}_{ij}$ of entry A_{ij} can be obtained s.t. $|\hat{A}_{ij} - A_{ij}| \leqslant \varepsilon + \alpha\tilde{\varepsilon}$ with probability at least $1 - \delta$, using poly(n)-sized circuits and at most $D(\tilde{\varepsilon}, \delta) = \Theta(\tilde{\varepsilon}^{-2} \log(4\delta^{-1}))$ calls to C-U_A.

Combining Lemma 4 with Theorems 1–3, we can, respectively, estimate entries of $M^{(\beta)}$ in Eq. (3), $G^{\delta,\beta,\omega}$ in Eq. (7), and $M(t)$ in Eq. (4), up to $1/\text{poly}(n)$ error with poly(n) effort. Note that—asymptotically—the circuit implementing the controlled block encodings (which is required for the Hadamard test in Lemma 4) is of the same size as the block-encoding circuits themselves.

We note that in case when $F(h) = M$ is a correlation matrix and H corresponds to a lattice model, one can also obtain correlation matrix entries in momentum space—by using U_M and the efficient quantum Fourier transform circuit [32].

Going beyond individual matrix elements, for any local fermionic Hamiltonian term H_x in H, for example, $H_x = (h_{ij}a_j^\dagger a_i + h_{ij}^* a_i^\dagger a_j)$ (with $|h_{ij}| \leqslant 1$) or $H_x = (V_{ijkl}a_i^\dagger a_j^\dagger a_k a_l + V_{ijkl}^* a_l^\dagger a_k^\dagger a_j a_i)$ (with $|V_{ijkl}| \leqslant 1$), the expectation of that term with respect to a state ρ can be efficiently extracted from the block encoding of its correlation matrix U_M [33]. In this way, one can also obtain the total energy density of ρ relative to a system Hamiltonian H. To do so, one can sample from the Hamiltonian terms uniformly at random and evaluate the expectation value of individual terms as mentioned above. For H being a free fermion Hamiltonian, this sampling can be implemented using the sparse access model discussed below; this method of sampling can be extended to interacting Hamiltonians. We can obtain the following concentration bound on this evaluated energy density e, assuming, for simplicity, that the expectation of an individual term is learned from U_M without error. By assumption, we have that $|\text{Tr}(H_x\rho)| \leqslant 1$ for each Hamiltonian term H_x. This allows us to infer the Chernoff bound, which says that for sample size $S = \Theta(\varepsilon^{-2} \log(\delta^{-1}))$, we have

$$\mathbb{P}(|e - \text{Tr}(H\rho)/K| \leqslant \varepsilon) \geqslant 1 - \delta, \qquad (21)$$

where $K = \Theta(2^n)$ is the number of terms in the Hamiltonian H. Similarly, densities of other Hermitian operators can be learned through sampling, such as the particle density $\langle \hat{N} \rangle / 2^n = \text{Tr}(M)/2^n$.

IX. COMPLEXITY

We have presented a method for simulating free fermionic systems on $N = 2^n$ modes with polynomial resources in n in a variety of settings. The naive classical treatment of 2^n fermionic modes, on the other hand, requires exponential resources. Therefore, the naive speedup of our quantum method is exponential. However, our approach comes with manifest qualifications, namely, the requirement for the oracle tuple $\mathcal{O}_h$ to be implementable using poly(n)-sized quantum circuits, time dynamics being simulable only for time $t = \text{poly}(n)$, thermal states for $\beta = \text{poly}(n)$, and Green's functions for $\beta, 1/\delta = \text{poly}(n)$. Competing classical approaches could hypothetically exploit this structure of our setting. To settle this issue, one can readily argue that our method generally yields an exponential quantum speedup, by showing that it solves a BQP-complete problem. Roughly speaking, BQP-complete problems are the hardest problems that can be efficiently solved by a quantum computer [34]. Since for single-particle dynamics, the character of the particle—be it a boson, fermion, or distinguishable particle—is not relevant, BQP hardness of time dynamics follows in principle from Theorem 3 in Ref. [17], using techniques such as those developed in Ref. [35]. For completeness, we provide a slightly different proof for the complexity of the evolution of a multiparticle fermionic state in Appendix H.

Theorem 4. Let ρ_0 be a (multiparticle) fermionic state on 2^n modes, such that its correlation matrix M_0 is sparse, and the access oracle tuple $\mathcal{O}_{M_0}$ can be implemented as a poly(n)-sized quantum circuit. Given a quadratic Hamiltonian H on 2^n modes, let h be as in Eq. (1) and sparse, and we assume that the oracle tuple $\mathcal{O}_h$ is implemented as a poly(n)-sized quantum circuit. For $t = \text{poly}(n)$, the problem is to decide whether, for some given mode j, $n_j(t) = \text{Tr}(a_j^\dagger a_j e^{-iHt} \rho_0 e^{iHt}) \geqslant 1/p(\sqrt{n})$ (with p a polynomial) or $\leqslant \exp(-\sqrt{n})$, given a promise that either one is the case. This problem is BQP complete.

X. QUANTUM SPEEDUP IN A VARIETY OF SETTINGS

We have established that our algorithms in principle provide an exponential speedup, at least in the setting of time evolution. In this section, we argue what the speedup is for several models of physical importance. To that end, let us first argue that for $d = O(1)$-dimensional lattice models, entries of our matrix functions of interest (see Sec. IV) can be estimated *classically* with poly(n) effort for $\beta, 1/\delta, t = \text{poly}(n)$.

Lieb-Robinson bounds [36–38] imply that the time evolution of observables such as the occupation number of a mode i at some position (starting from a product state with some modes occupied and others unoccupied) is only affected by $O(t^d) = \text{poly}(n)$ sites in a ball of radius proportional to t around that position. Similarly, Ref. [36] shows that, for a given mode i, the thermal correlation matrix entries $|M_{ij}^{(\beta)}|$ decay exponentially with the distance between modes i and

j, with a characteristic length $O(\beta)$. Mode i is therefore only nontrivially correlated with $O(\beta^d) = \text{poly}(n)$ modes in a ball of radius $O(\beta)$ around it. This latter fact suggests that an entry $M_{ij}^{(\beta)}$ can be classically evaluated with poly(n) effort, provided that $\beta = \text{poly}(n)$. Let us formalize this as follows.

Lemma 5. Let $h \in \mathbb{C}^{2^n \times 2^n}$ be an $s = O(1)$-sparse matrix that corresponds to a $d = O(1)$-dimensional lattice model; cf. Eq. (1) with entries as in Eqs. (13) and (14). Assume poly(n)-effort classical access to the oracles O_r and O_a (see Definition 2) for h. Let $F(h)$ be a matrix function of h. If $\max_{x \in [-1,+1]} |F(x) - p_K(x)| \leqslant \text{poly}(n)/K$ with $p_K(x)$ a degree-K polynomial, then an entry $F(h)_{ij}$ can be estimated with that same error using poly(K) × poly(n) classical effort. For some $K = \text{poly}(n)$, the error thus becomes $1/\text{poly}(n)$ with poly(n) classical effort.

Proof. If one is able to estimate $\langle i|h^k|j \rangle$ for any $k \in \{0, 1, \ldots, K\}$ with effort E, then $\langle i|p_K(h)|j \rangle = \sum_{k=0}^{K} \alpha_k \langle i|h^k|j \rangle$ can be evaluated with effort $K \times E$. By assumption, $\langle i|F(h)|j \rangle$ can then be classically approximated up to poly(n)/K error with $K \times E$ effort. Since h corresponds to a $d = O(1)$-dimensional lattice model, $h^k|j \rangle$ is only supported on $O(k^d) = \text{poly}(k)$ $|i \rangle$'s. We can thus evaluate each $\langle i|h^k|j \rangle$ for $k \in \{0, 1, \ldots, K\}$ using poly(k) calls to the oracles and with a total $E = \text{poly}(k) \times \text{poly}(n)$ computational effort. Therefore, $\langle i|F(h)|j \rangle$ can be approximated classically with poly(n)/K error with $K \times \text{poly}(k) \times \text{poly}(n) = \text{poly}(K) \times \text{poly}(n)$ effort. Clearly, there is a $K = \text{poly}(n)$ so that the error becomes $1/\text{poly}(n)$ and which yields a poly(n) classical effort. ∎

Combined with Lemmas 2 and 3, Lemma 5 implies the following for $d = O(1)$-dimensional lattice models. In the parameter regimes of Theorems 1 and 2, entries of the thermal correlation matrix in Eq. (3) and of the thermal Green's function in Eq. (7) can be estimated up to $1/\text{poly}(n)$ error with poly(n) classical effort.

Using similar reasoning, entries of the time-evolved correlation matrix $M(t)$ in Eq. (4) can be evaluated classically with poly(n) effort for $t = \text{poly}(n)$. In fact, assuming exact classical access to entries $\langle k|M|l \rangle$ of an initial correlation matrix M for given (k, l), one can obtain entries $M(t)_{ij}$ with $1/\exp(n)$ error. The improved error scaling comes from the fact that the polynomial approximation error of $\exp(ith)$ can be bounded by $1/\exp(n)$ even for degree $K = \text{poly}(n)$, provided that $t = \text{poly}(n)$. A detailed treatment is given in Appendix I. Note that if we apply the time evolution to $M'^{(\beta)}$ (where $M'^{(\beta)}$ is the thermal correlation matrix corresponding to some $h' \neq h$), the accuracy reduces to $1/\text{poly}(n)$ due to the error in estimating entries of $M'^{(\beta)}$.

Despite losing the exponential speedup for $d = O(1)$-dimensional lattice models, let us argue that we retain a power-$(d + 1)$ polynomial speedup for such models. Let us focus on the task of estimating entries of the time-evolved correlation matrix from Eq. (4). In particular, let us focus on the task of time evolution for t proportional to the Lieb-Robinson time t_{LR}, which is the time it takes for a Lieb-Robinson light cone to contain the entire system. For lattice models, $t_{\text{LR}} = N^{1/d}$. To then compute an entry of the correlation matrix $M(t) = e^{iht} M e^{-iht}$, known classical algorithms require $\Omega(Nt) = \Omega(t^{d+1})$ run-time [39]. Given the poly log(N) · t run-time of our quantum algorithm, we obtain

TABLE I. Asymptotic run-times for evaluating entries of time-evolved correlation matrices [with $1/\text{poly}(n)$ error] for three different system types: lattice models, expander graphs, and general sparse models. For the former two, we start from a thermal correlation matrix at $\beta = O(1)$ (of some h', different from h used for time evolution). For the latter, we start from a Slater determinant (free fermion pure) state. The third row gives the Lieb-Robinson time (only denoted for lattice models and expander graphs), which corresponds to the time it takes the Lieb-Robinson light cone to contain the entire system. The run-times of classical algorithms (for evolutions over a time interval that is at least the Lieb-Robinson time) and our quantum algorithms are given. In addition, we provide the associated speedups for the lattice models and expander graphs at the Lieb-Robinson time, and the speedup for general sparse models at $t = \text{poly}(n)$. The run-times required to prepare the starting state are denoted by $r_{\text{prep},C}$ and $r_{\text{prep},Q}$, for, respectively, the classical and quantum algorithms. Note that $r_{\text{prep},Q} = \text{poly}(n)$ in all three scenarios.

	$d = O(1)$-dimensional lattice models	Expander graphs	General sparse models
Classical algorithms[a]		$r_{\text{prep},C} + N \cdot t$	
Quantum algorithms		$r_{\text{prep},Q} + \text{poly}\log(N) \cdot t$	
Lieb-Robinson time	$N^{1/d}$	$\log(N)$	–
Speedup	Power-$(d + 1)$ polynomial[b]	Exponential[b]	Exponential[c]

[a]Run-times of—to the best of our knowledge—the best classical algorithm for these applications [39].
[b]Speedups compared to the aforementioned classical algorithms.
[c]Speedup assuming that it takes exponential (in n) time to solve BQP-complete problems classically.

a power-$(d + 1)$ polynomial speedup. In particular, this yields a cubic speedup for $d = 2$ lattices and quartic speedup for $d = 3$ —which can be of interest in early fault-tolerant devices [40].

Crucially, our method can also be applied to settings other than lattice models, and the exponential speedup for those settings can be maintained. Let us consider tight-binding models on expander graphs, such as the Margulis graph considered in Sec. VI. The Lieb-Robinson time, due to the expansion property of the graph, will be logarithmic in the number of modes N: $t_{\text{LR}} = \log(N)$. We note that light cones also grow rapidly in other graphs with log-sized diameter, such as the hyperbolic lattices (see Ref. [41] for recent studies of such tight-binding models). We expect to recover the full exponential quantum speedup for their simulation because at $t_{\text{LR}} = \log(N)$, the quantum run-time is $\text{poly}\log(N)$ while known classical algorithms have run-time $\Omega(Nt) = \Omega(N \log(N))$ [39]. This speedup can be of particular interest for, e.g., the study of Anderson localization on expander graphs [24–27].

To summarize the quantum advantage in different problem settings, Table I gives an overview of the asymptotic run-times of classical algorithms and our quantum algorithms, and associated quantum speedups for the problem of time evolution.

XI. GENERALIZATIONS

The time-evolution framework presented in this paper can be made more general and applied to systems beyond free fermions. In a general quantum system described by a Hamiltonian H, one can consider a N-sized set of operators $\{O_j\}$ such that $[H, O_k] = \sum_{j=1}^{N} h_{jk} O_j$. This is sufficient for a matrix $M_{jk} = \text{Tr}(\rho O_j^\dagger O_k)$ to transform as $M \mapsto e^{-iht} M e^{iht}$ under time evolution. Further assuming that h is a hermitian matrix allows treatment of M as a block encoding of the type considered in this work. Beyond the free fermionic systems on which we focused in this work, this general framework admits fermionic H that includes pairing ($\Delta a_j a_k +$ H.c.) terms. In this case, the relevant set $\{O_j\}$ would include not just annihilation but also creation operators. Another example is a system of 2^n free bosons with particle conservation, in which

case $\{O_j\}$ should be chosen as bosonic annihilation operators. Beyond $\text{Tr}(\rho O_j^\dagger O_k)$, one can consider $M_{j_1,...,j_l;k_1,...,k_{l'}} = \text{Tr}(\rho O_{j_1}^\dagger ..O_{j_l}^\dagger O_{k_1}..O_{k_{l'}})$, which can be considered as a rectangular matrix acting on $n \cdot \max(l, l')$ qubits, and block encoded accordingly. The time evolution of these objects is defined similarly to that of M_{jk}, and therefore can be easily found as a block encoding, given the block encoding of the initial state. The flexibility of this general block-encoding framework is comparable to the one based on "shadow" states, presented in Ref. [19] (see Appendix A for a discussion of the differences).

XII. DISCUSSION

In this work, we develop quantum algorithms that solve several free fermion problems. We discuss in detail what type of speedup is achieved over classical algorithms and present generalizations of our approach.

One obvious avenue for future research is to apply our method to other matrix functions of h. For example, one should be able to estimate the free energy density of a 2^n-mode free fermion system $\frac{F}{2^n} = -(\beta 2^n)^{-1} \log \text{Tr} (e^{-\beta H}) = -(\beta 2^n)^{-1} \text{Tr} (\log(I + e^{-\beta h}))$ with error ε, using a polynomial approximation of the function $\log(I + e^{-\beta h})$ for $\beta = \text{poly}(n)$, the block encoding of h, and sampling entries to model the trace function. Using an estimate of the free energy density $F/2^n = (\langle H \rangle_\beta - \beta^{-1} S(\rho_\beta))/2^n$, one can in turn estimate an entropy density, given an energy density estimate, or a derivative of $F/2^n$ with respect to β such as the specific heat. Another possible generalization of our work is a $\text{poly}(n)$-efficient estimation of matrix elements or observable expectations due to free fermionic *dissipative* dynamics, which was shown to be classically simulatable in $O(2^{3n})$ time in Ref. [42].

One could also consider how block-encoding techniques fare when applied to estimating entries of a free-bosonic thermal correlation matrix $M^{(\beta)} = I/(e^{\beta h} - I)$ of Bose-Einstein form. A block encoding of a polynomial approximation as developed in Lemma 2 and Theorem 1 in Sec. VII requires a $\text{poly}(n)$ bound on the mode occupation number (so that the matrix function be block encoded), which can, however, grow

as large as the number of particles for a Bose-Einstein condensate. Mathematically, the Bose-Einstein distribution with $\epsilon_i \geqslant 0$ has a singularity at $\epsilon_i = 0$, which has to be avoided (by choosing a small enough chemical potential μ) in order to place any bound. Note that similar points about only algebraic speedups for local lattice models (Lemma 5) were made for bosonic/oscillator systems in a more recent work [43].

Another outstanding open direction is to compute and optimize the precise implementation overhead and circuit depth for our proposed algorithms, as applied to simulation problems of practical interest.

Let us point out an open question in the setting of time dynamics on 2^n fermionic modes [cf. Eq. (4)]. One task that can be performed with poly(2^n) classical effort [2] is computing the overlap

$$|\langle S_1 | \exp(-itH) | S_2 \rangle|^2$$
$$= \text{Tr}\Big[\underbrace{\exp(-itH)|S_2\rangle\langle S_2| \exp(itH)}_{|S_3\rangle\langle S_3|} |S_1\rangle\langle S_1| \Big], \quad (22)$$

with $|S_1\rangle$ and $|S_2\rangle$ single-Slater determinant states and H a free fermion Hamiltonian as in Eq. (1), and therefore $|S_3\rangle$ is also a Slater determinant state. If $|S_1\rangle\langle S_1|$ (for simplicity) is a standard-mode-basis Slater determinant state, then it can be expressed as a product of 2^{n+1} creation and annihilation operators. Using Wick's theorem, evaluating this weight-2^{n+1} correlator in Eq. (22) requires evaluating products of 2^n entries of the correlation matrix [cf. Eq. (2)] of state $|S_3\rangle$. This task—at least with naive attempts—cannot be performed using our methods with poly(n) quantum effort, since we can only evaluate poly(n) entries of the time-evolved correlation matrix, although approximate sampling methods could come into play.

ACKNOWLEDGMENTS

DATA AVAILABILITY

No data were created or analyzed in this article.

APPENDIX A: ALTERNATIVE ENCODINGS

In this section, we describe alternative ways of representing a fermionic correlation matrix using qubits and their potential drawbacks.

A compressed representation of free fermionic states on 2^n modes, as well as their dynamics, is readily obtained by using a (mixed) quantum state $\sigma = M/\text{Tr}(M)$ of n qubits to represent the normalized correlation matrix of ρ. One then computes, evolves and measures, with σ to learn properties of ρ or its time dynamics. For pure single-particle free fermionic states ρ, σ is a rank-1 projector, and σ projects onto the bit string $|i\rangle$ when ρ corresponds to $a_i^\dagger |\text{vac}\rangle$, $i = 0, \ldots, N-1$, where $|\text{vac}\rangle$ is the fermionic vacuum state. Once a state σ is prepared, its time evolution can readily be simulated: when ρ evolves via e^{-iHt} with free fermion Hamiltonian H, $\sigma \rightarrow e^{iht}\sigma e^{-iht}$. Sparse oracle access to h—see Definition 2—then allows for the efficient implementation of time evolution in terms of its dependence on t and calls to the oracle [13,15], starting from some easy-to-prepare initial state. For example, the initial state could be a set of fermions in a subset S of 2^m modes $|i\rangle$ (such that an efficient classical circuit can map S onto the set of m-bit strings) or a subset of modes in the Fourier-transformed basis (as the quantum Fourier transform is an efficient quantum circuit). One can also adapt the heuristic quantum Metropolis-Hastings algorithm [44,45] to the Fermi-Dirac distribution and sparse Hamiltonians h, since the algorithm uses quantum phase estimation for e^{iht} at its core. Even though the algorithm converges to the thermal state $\sigma_\beta = M^{(\beta)}/\text{Tr}(M^{(\beta)})$, poly($n$) efficiency is not guaranteed and unlikely for low-enough temperature.

Given a state σ, one can apply any learning algorithm for n-qubit states. For example, one can use shadow tomography [46] to estimate the expectation of L observables, such as $O_k = |k\rangle\langle k|$, $O_{lk}^R = |l\rangle\langle k| + |k\rangle\langle l|$, and $O_{lk}^L = i(|l\rangle\langle k| - |k\rangle\langle l|)$, with computational effort $O(\log(L))$ using random Clifford circuits of poly(n) size.

There are a few disadvantages to this simple and direct method of representing the state via its correlation matrix. It is not immediately obvious how to estimate a time-dependent correlation function as in Eq. (6) as it relates to measurements on $e^{iht_1}\sigma e^{-iht_2}$ that is not a state. Second, and more crucially, any learning of a linear function of σ with accuracy ε leads to learning with accuracy $\varepsilon \text{Tr}(M) = \varepsilon\langle\hat{N}\rangle$ for the correlation matrix M itself. Therefore, one expects poor accuracy for large particle number $\langle\hat{N}\rangle$; this in particular makes it impractical to extract individual matrix elements.

Thus, in the main text of this paper, we choose not to directly encode a correlation matrix as a quantum state, but rather apply quantum computational block-encoding techniques.

Recently, Ref. [19] introduced a general quantum simulation framework with compressed "shadow" quantum states with applications to free bosons and free fermion systems. We note that the results in Ref. [19] use yet a different encoding than the encoding described above, or the block encoding in the main text. Like for the encoding in the previous paragraph, the normalization of the shadow state in Ref. [19] can lead to a loss of efficiency if one wishes to estimate only few entries of the correlation matrix (this loss of efficiency is avoided in our block-encoding method). In particular, the normalization of the shadow state is a, which is bounded as $\sqrt{\sum_j(\langle\hat{N}_j\rangle - 1/2)^2} \leqslant a \leqslant \exp(n)$, where $\langle\hat{N}_j\rangle$ is the occupation number in the mode j of the represented state ρ. On the other hand, when estimating densities, for example, the energy density, our methods use sampling to estimate $\text{Tr}(H\rho)/K$ (with $K = \Theta(2^n)$, the number of terms in H) with some error ε, while Ref. [19] estimates $\text{Tr}(H\rho)/O(2^{n/2}a)$, which, depending on the value a, can be more efficient.

The precise relation between the shadow state approach [19] and the block-encoding framework presented in this work is currently unclear. A plausible hypothesis is that the latter is strictly more powerful, due to the signal strength difference discussed above. A concrete interesting question is whether a shadow state corresponding to $M_{jk} = \text{Tr}(\rho a_j^\dagger a_k)$ [or, more generally, $\text{Tr}(\rho O_j^\dagger O_k)$] can always be produced using a block-encoding U_M of M. In the "typical" case $\text{Tr}(M^\dagger M) = \Theta(2^n)$, this can be done simply by acting with U_M on the maximally entangled state between j and k registers, and postselecting on the zero value of ancillary qubits. This "Choi–Jamiołkowski" strategy, however, does not give a constant success rate when $\text{Tr}(M^\dagger M) = o(2^n)$, and should be adapted.

APPENDIX B: REMARKS ON ORACLE CONVENTIONS

In this work, we define row and matrix entry oracles as in Definition 2. An alternative definition of a row oracle, used in, for instance, Ref. [14], is

$$O_r^{\text{alt}}|i\rangle|k, 0^{(n+1)-\lceil \log(s)\rceil}\rangle = |i\rangle|r(i,k)\rangle, \quad \forall i \in [2^n], k \in [s],$$
(B1)

with O_r^{alt} acting on $2(n+1)$ qubits. Again, if row i contains $s' < s$ nonzero entries, then the last $n+1$ qubits are set to $|1\rangle|k\rangle$. We note that having access to O_r in Eq. (11) implies access O_r^{alt}, and vice versa.

In Ref. [14], O_r^{alt} and O_a are used to block encode a sparse matrix h. In principle, this block-encoding scheme requires another (column) oracle O_c^{alt} when it is used to block encode *general* sparse matrices h. If h is also Hermitian, which is the case for all applications considered in this work, this block encoding can be implemented with just O_r^{alt} and O_a, since O_c^{alt} can be realized using O_r^{alt} and some SWAP gates.

APPENDIX C: MARGULIS EXPANDER GRAPHS

In the main text, we have provided an example of a d-dimensional model that has sparse query access. Going beyond these models, in this Appendix we consider an example of a model on an *expander graph* that has sparse query access. Expander graphs are bounded-degree graphs, which have the so-called *expansion* property. In particular, when counting the vertices away from a given vertex by a distance d, one obtains a number that scales exponentially with d. We will focus on realizing sparse access for a particular simple example, which is the Margulis expander graph.

A Margulis graph $\mathcal{G}_M$ of size N^2 has vertices v labeled by tuples $v = (v_1, v_2) \in [N] \times [N]$; an edge between two vertices u and v is placed if $u = t_l(v)$, where the functions t_l for $l \in [4]$ are defined as $t_0((v_1, v_2)) = (v_1 + 1 \bmod N, v_2)$, $t_1((v_1, v_2)) = (v_1, v_2 + 1 \bmod N)$, $t_2((v_1, v_2)) = (v_1 + v_2 \bmod N, v_2)$, and $t_3((v_1, v_2)) = (v_1, v_2 + v_1 \bmod N)$. In other words, the first two types of edges are simple nearest-neighbor links along the vertical and horizontal directions, with periodic boundary conditions. From this perspective, the edges t_2 and t_3 are geometrically nonlocal, and are the source of the expansion property of the graph. We define our tight-binding Hamiltonian on the Margulis graph as follows. Each fermionic mode is labeled by the vertex of the graph, so the total number of modes is N^2. The Hamiltonian takes the form

$$H_{\text{Marg}} = \sum_{l \in [4]} \sum_{v \in [N] \times [N]} (a_v^\dagger a_{t_l(v)} + a_{t_l(v)}^\dagger a_v).$$
(C1)

For a given v, modular addition circuits allow to efficiently generate a list of $u = t_l^{\pm 1}(v)$. This list can be used to construct an oracle O_r; to ensure distinct outputs, if some of eight values of u coincide, one stores only one of the colliding outputs. The oracle O_a then represents collisions with an increased matrix element h_{vu}, realized by counting the times u occurs in the list of $t_l^{\pm 1}(v)$. We expect that more models on expander graphs can be implemented in a similar way—especially in the family of constant degree Ramanujan Cayley graphs, of which the Margulis graph is an example.

APPENDIX D: BLOCK ENCODING THE THERMAL CORRELATION MATRIX

In this Appendix, we prove Theorem 1. In particular, we prove a more detailed version of it, namely, Theorem D1 below. In its proof, we use Propositions 1 and 2 on the block encoding of polynomials of sparse matrices, and Proposition 3 and Lemma D1 (of which Lemma 2 is a simplified version) on constructing a polynomial approximation to our desired matrix function $M^{(\beta)}$ in Eq. (3). We will first prove Theorem D1 and then Lemma D1.

As was argued in Sec. VII using Proposition 3, we wish to construct accurate polynomial approximations of $F(sx)$ for $x \in [-1, +1]$. Let us state Lemma D1, which will be proved at the end of this section.

Lemma D1. For a function $f(x) = \frac{1}{4} \frac{1}{1 + \exp \beta sx}$ (with $\beta s > 0$, $x \in [-1, +1]$), one can efficiently construct a polynomial $p_d(x)$ of degree d such that

$$\max_{x \in [-1, +1]} |f(x) - p_d(x)| \leqslant \begin{cases} \frac{3}{d}\left(\frac{\beta s}{\pi}\right)^4, & \text{if } \frac{\beta s}{2\pi} \geqslant 1, \\ \frac{10}{d}\left(\frac{\beta s}{\pi}\right)^2, & \text{if } \frac{\beta s}{2\pi} < 1. \end{cases}$$
(D1)

Theorem D1. For an s-sparse Hamiltonian h on n qubits, assume access to the oracle tuple $\mathcal{O}_h$. We denote the controlled $(1, n+5, \varepsilon_{\text{PA}} + \varepsilon_{p(h)} + \delta)$-block encoding of $M^{(\beta)} = \frac{1}{4} \frac{1}{1 + \exp(\beta h)}$ by $C\text{-}U_{M^{(\beta)}}$. The implementation of this block encoding requires

$$\begin{cases} \Theta\left(\dfrac{\beta^4 s^4}{\varepsilon_{\text{PA}}}\right), & \text{if } \dfrac{\beta s}{2\pi} \geqslant 1, \\[3mm] \Theta\left(\dfrac{\beta^2 s^2}{\varepsilon_{\text{PA}}}\right), & \text{if } \dfrac{\beta s}{2\pi} < 1 \end{cases}$$
(D2)

calls to oracles from the oracle tuple $\mathcal{O}_h$,

$$O\left(sn + n_a + \log^{5/2}\left(16s^9\beta^8 / \left(\varepsilon_{\text{PA}}^2 \varepsilon_{p(h)}^2\right)\right)\right)$$
(D3)

ancillary qubits, and

$$O\left(n + (n+4)\beta^4 s^4 / \varepsilon_{\text{PA}} + \log^{5/2}\left(16s^9\beta^8 / \left(\varepsilon_{\text{PA}}^2 \varepsilon_{p(h)}^2\right)\right)\right)$$
(D4)

additional one-qubit and two-qubit gates. To implement this block encoding, an additional classical computing time of $\text{poly}(\beta^4 s^4/\varepsilon_{\text{PA}}, \log(1/\delta))$ is required.

Proof. It follows from Proposition 1 (from Ref. [14]) that with $O(1)$ calls to the oracle tuple $\mathcal{O}_h$, one can construct a $(s, n + 3, \varepsilon_{\text{BE}_h})$-block-encoding U_h of h and its controlled version. For a given $\varepsilon_{\text{BE}_h}$, the required number of ancillary qubits and (additional) elementary gates are given in Proposition 1.

Let $p_d(x)$ denote the degree-d polynomial approximation of the function $\frac{1}{4} \frac{1}{1+\exp(\beta s x)}$ as in Lemma D1. It follows from Lemma D1 that one can efficiently construct p_d such that

$$\|p_d(h/s) - 1/4\, M^{(\beta)}\| \leqslant \begin{cases} \frac{3}{d}\left(\frac{\beta s}{\pi}\right)^4, & \text{if } \frac{\beta s}{2\pi} \geqslant 1, \\ \frac{10}{d}\left(\frac{\beta s}{\pi}\right)^2, & \text{if } \frac{\beta s}{2\pi} < 1, \end{cases} \quad \text{(D5)}$$

Taking $d = \Omega\left(\frac{\beta^4 s^4}{\varepsilon_{\text{PA}}}\right)$ if $\frac{\beta s}{2\pi} \geqslant 1$ and $d = \Omega\left(\frac{\beta^2 s^2}{\varepsilon_{\text{PA}}}\right)$ if $\frac{\beta s}{2\pi} < 1$, we achieve $\|p_d(h/s) - 1/4\, M^{(\beta)}\| \leqslant \varepsilon_{\text{PA}}$.

For $\varepsilon_{\text{PA}} < \frac{1}{4}$, we note that $|p_d(x)| \leqslant 1/2$ for $x \in [-1, +1]$. Therefore, we can apply Proposition 2 (from Ref. [14]): A $(1, n + 5, 4d\sqrt{\varepsilon_{\text{BE}_h}/s} + \delta)$-block encoding of $p_d(h/s)$ consists of at most d uses of unitaries U_h, $U_h^\dagger$ or controlled U_h, and $O((n + 4)d)$ elementary gates. In addition, it requires a classical computation with run-time as stated in Proposition 2. We take $\varepsilon_{p(h)} := 4d\sqrt{\varepsilon_{\text{BE}_h}/s}$ so that for a given $\varepsilon_{p(h)}$, we should ensure that $\varepsilon_{\text{BE}_h} = s\varepsilon_{p(h)}^2/(16d^2)$.

Let the $(1, n + 5, \varepsilon_{p(h)} + \delta)$-block encoding of $p_d(h/s)$ be denoted by $U_{p_d(h/s)}$. We can bound how well the block encoding of $p_d(h/s)$ approximates the block encoding of $1/4\, M^{(\beta)}$ as

$$\begin{aligned} \varepsilon_{\text{Tot}} &= \|1/4\, M^{(\beta)} - \langle 0|^{\otimes a} \otimes \mathbb{1} U_{p_d(h/s)} |0\rangle^{\otimes a} \otimes \mathbb{1}\| \\ &\leqslant \|1/4\, M - p_d(h/s)\| \\ &\quad + \|p_d(h/s) - \langle 0|^{\otimes a} \otimes \mathbb{1} U_{p_d(h/s)} |0\rangle^{\otimes a} \otimes \mathbb{1}\| \\ &\leqslant \varepsilon_{\text{PA}} + \varepsilon_{p(h)} + \delta. \end{aligned} \quad \text{(D6)}$$

We have thus constructed a $(1, n + 5, \varepsilon_{\text{PA}} + \varepsilon_{p(h)} + \delta)$-block encoding of $1/4\, M^{(\beta)}$. To implement this block encoding, we require a number of calls to oracles from the tuple $\mathcal{O}_h$, a number of ancillary qubits, and a number of one-qubit and two-qubit gates as in the lemma statement. ∎

Let us now give the proof of Lemma D1.

Proof. For the proof of this lemma, we will employ Bernstein's theorem (Lemma 1) that bounds the error of Chebyshev approximations. Such a Chebyshev approximation of degree d is of the form $p_d(x) = \sum_{k=0}^{d} a_k T_k(x)$, where $T_k(\cos(\theta)) := \cos(k\theta)$ is the degree-k Chebyshev polynomial of the first kind. The coefficients a_k can be obtained by evaluating

$$a_k = \frac{2}{\pi} \int_{-1}^{+1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}} dx, \quad \text{(D7)}$$

with $\frac{2}{\pi}$ replaced by $\frac{1}{\pi}$ for $k = 0$. Each a_k can be evaluated classically with $\text{poly}(\beta s k)$ resources for $f(x)$ in the lemma statement.

Note that the function $f(z = x + iy) = \frac{1}{1+\exp(\beta s z)}$ for $\beta s > 0$ is analytic for $|y| \leqslant \pi/\beta s$. Hence, we can pick the ellipse $E_r = \{\frac{1}{2}(z + z^{-1}) : |z| = r\}$ with $r = \frac{1}{2}\sqrt{(2\pi/\beta s)^2 + 4}$ on which $f(z)$ is analytic, since within this ellipse $|y| \leqslant \frac{\pi}{2\beta s}$.

We have $|f(z)| \leqslant C = 1$ for $z \in E_r$ since for $|y| \leqslant \frac{\pi}{2\beta s}$, we have

$$|1 + \exp(\beta s z)| \geqslant |1 + \exp(\beta s x)\cos(\beta s y)| \geqslant 1. \quad \text{(D8)}$$

Using Lemma 1, we can thus bound $\max_{x \in [-1, +1]}|f(x) - p_d(x)|$ as

$$\max_{x \in [-1, +1]}|f(x) - p_d(x)| \leqslant \frac{2\left((\pi/\beta s)^2 + 1\right)^{-d/2}}{\frac{1}{2}\sqrt{(2\pi/\beta s)^2 + 4} - 1}. \quad \text{(D9)}$$

Let us distinguish between scenario (1) $\beta s \geqslant 2\pi$ and scenario (2) $\beta s < 2\pi$. For scenario (1), we can bound

$$\frac{1}{2}\sqrt{(2\pi/\beta s)^2 + 4} - 1 \geqslant \frac{1}{12}(2\pi/\beta s)^2. \quad \text{(D10)}$$

Furthermore, in both scenarios (1) and (2), we have that

$$\begin{aligned} \left((\pi/\beta s)^2 + 1\right)^{-d/2} &\leqslant 1/((\pi/\beta s)^2 d/2 + 1) \\ &\leqslant 1/((\pi/\beta s)^2 d/2). \end{aligned} \quad \text{(D11)}$$

Combining these two facts leads to the following bound in scenario (1):

$$\max_{x \in [-1, +1]}|f(x) - p_d(x)| \leqslant \frac{12}{d}\left(\frac{\beta s}{\pi}\right)^4. \quad \text{(D12)}$$

In scenario (2), we can simply bound the denominator in Eq. (D9) by

$$\frac{1}{2}\sqrt{(2\pi/\beta s)^2 + 4} - 1 \geqslant \frac{1}{2}\sqrt{5} - 1 \geqslant 1/10. \quad \text{(D13)}$$

Combining this with the upper bound above for the numerator in Eq. (D9) (which holds in both scenarios), we obtain the following upper bound in scenario (2).

$$\max_{x \in [-1, +1]}|f(x) - p_d(x)| \leqslant \frac{40}{d}\left(\frac{\beta s}{\pi}\right)^2. \quad \text{(D14)}$$

∎

APPENDIX E: BLOCK ENCODING THE TIME-EVOLVED CORRELATION MATRIX

In this Appendix, we prove Theorem E1 below, which is a generalization of Theorem 3 for block-encoding $M(t_1, t_2)$ in Eq. (5). We will use a result from Ref. [23] on block-encoding $\exp(iht)$ using a block encoding of h. Note that the error of the block encoding of $M(t_1, t_2)$ in the theorem statement accounts for potential errors in the block encoding of the initial correlation matrix as well.

Theorem E1. For an s-sparse Hamiltonian h on 2^n fermionic modes, assume access to the oracle tuple $\mathcal{O}_h$. Also assume access to the $(\alpha, m, \varepsilon_M)$-block-encoding U_M of a correlation matrix M of a fermionic state on 2^n modes. The $(\alpha, 2n + m + 10, \varepsilon + \varepsilon_M)$-block-encoding $U_{M(t_1, t_2)}$ of

$$M(t_1, t_2) = e^{iht_1} M e^{-iht_2} \quad \text{(E1)}$$

can be produced using

$$\begin{aligned} D(\alpha&, \varepsilon, t_1, t_2) \\ &= O(s(|t_1| + |t_2|) + \log(12\alpha(|t_1| + |t_2|)/(|t_1|\varepsilon)) \\ &\quad + \log(12\alpha(|t_1| + |t_2|)/(|t_2|\varepsilon))) \end{aligned} \quad \text{(E2)}$$

calls to oracles from the tuple $\mathcal{O}_h$, and a single use of the block-encoding U_M. Moreover, one uses $O((n+3)(s(|t_1| + |t_2|) + \log(2\alpha(|t_1| + |t_2|)/(|t_1|\varepsilon)) + \log(2\alpha(|t_1| + |t_2|)/(|t_2|\varepsilon)) + D(\alpha, \varepsilon, t_1, \quad t_2)(n + \log^{5/2}(2\alpha s^2(|t_1| + |t_2|)/\varepsilon)))$ one-qubit and two-qubit gates, and $O(n_a + \log^{5/2}(2\alpha s^2(|t_1| + |t_2|)/\varepsilon))$ ancillary qubits (with n_a denoting the number of bits with which the entries of h are specified).

Proof. A block-encoding $U_{M(t_1,t_2)}$ of $M(t_1, t_2)$ can be constructed using products of block encoding $U_{\exp(ith)}$ of $\exp(ith)$ (for times t_1 and $-t_2$) and U_M of M [where the latter is a $(\alpha, m, \varepsilon_M)$-block encoding by assumption].

To construct a block encoding of $\exp(iht)$, we employ a block encoding of h. It follows from Proposition 1 (from Ref. [14]) that with $O(1)$ calls to the oracle tuple $\mathcal{O}_h$, one can construct a $(s, n+3, \varepsilon_{\mathrm{BE}_h})$-block-encoding U_h of h and its controlled version. For a given $\varepsilon_{\mathrm{BE}_h}$, the required number of ancillary qubits and (additional) elementary gates are given in Proposition 1.

Corollary 62 in Ref. [23] states that to implement a $(1, n+5, |2t|\varepsilon_{\mathrm{BE}_h})$-block-encoding of $\exp(ith)$, one is required to implement U_h or $U_h^\dagger$ a total of $6s|t| + 9\log((6/(|t|\varepsilon_{\mathrm{BE}_h}))$ times, and controlled U_h or controlled $U_h^\dagger$ three times. In addition, one has to use $O((n+3)(s|t| + \log((2/\varepsilon_{\mathrm{BE}_h}))$ two-qubit gates and $O(1)$ ancillary qubits. So, to implement the $(1, n+5, |2t|\varepsilon_{\mathrm{BE}_h})$-block-encoding of $\exp(ith)$, one is required to call $\mathcal{O}_h$ a total of $O(s|t| + \log(6/(|t|\varepsilon_{\mathrm{BE}_h})))$ times.

Using Lemma 30 in Ref. [14], the block-encoding $U_{M(t_1,t_2)}$ of $M(t_1, t_2)$ can be constructed using the product $U_{M(t_1,t_2)} = (\mathbb{1}_{n+5+m} \otimes U_{\exp(iht_1)})(\mathbb{1}_{2n+10} \otimes U_M)((\mathbb{1}_{n+5+m} \otimes U_{\exp(-iht_2)})$, such that $U_{M(t_1,t_2)}$ is a $(\alpha, 2n+m+10, 2\alpha\varepsilon_{\mathrm{BE}_h}(|t_1| + |t_2|) + \varepsilon_M)$-block encoding. To implement this product, one is thus required to make

$$D(\varepsilon_{\mathrm{BE}_h}, t_1, t_2) = O(s(|t_1| + |t_2|) + \log(6/(|t_1|\varepsilon_{\mathrm{BE}_h}))$$
$$+ \log(6/(|t_2|\varepsilon_{\mathrm{BE}_h}))) \quad \text{(E3)}$$

calls to oracles from the tuple $\mathcal{O}_h$. In addition, one has to use a total of $O((n+3)(s(|t_1| + |t_2|) + \log(1/(|t_1|\varepsilon_{\mathrm{BE}_h}) + \log(1/(|t_2|\varepsilon_{\mathrm{BE}_h}) + D(\varepsilon_{\mathrm{BE}_h}, t_1, \quad t_2)(n + \log^{5/2}(s^2/\varepsilon_{\mathrm{BE}_h}))$ one-qubit and two-qubit gates, and $O(n_a + \log^{5/2}(s^2/\varepsilon_{\mathrm{BE}_h}))$ ancillary qubits.

We stress that a controlled version $C\text{-}U_{M(t_1,t_2)}$ of the block encoding of $U_{M(t_1,t_2)}$ can be implemented with equivalent resources. ∎

APPENDIX F: BLOCK ENCODING THE THERMAL GREEN'S FUNCTION

In this Appendix, we prove Theorem F1, which is a more detailed version of Theorem 2. In its proof, we will again use Propositions 1 and 2 on the block encoding of polynomials of sparse matrices. In addition, we will use Proposition 3 and Lemma F1 (of which Lemma 3 is a simplified version) on constructing a polynomial approximation to our desired matrix function $G^{(\delta,\beta,\omega)}$ in Eq. (7).

As was argued in Sec. VII using Proposition 3, we would like to construct accurate polynomial approximations of $F(sx)$ for $x \in [-1, +1]$. The function to be approximated for block-

encoding $G^{(\delta,\beta,\omega)}$ is

$$g^{(\delta,\beta,\omega)}(x) := \frac{\delta}{8}\left[\left(1 - \frac{1}{1 + \exp(\beta sx)}\right)\frac{1}{i\delta - (sx + \omega)}\right.$$
$$\left. + \left(\frac{1}{1 + \exp(\beta sx)}\right)\frac{-1}{i\delta + (sx + \omega)}\right]. \quad \text{(F1)}$$

Note that $g^{(\delta,\beta,\omega)}(z)$ ($z \in \mathbb{C}$) has poles at $z = \frac{i\delta - \omega}{s}$ and $z = \frac{-i\delta - \omega}{s}$; the regularization parameter δ ensures that these poles lie off the real axis. For convenience, we define the functions $g_1^{(\delta,\omega)}(z) = 1/(i\delta - (sz + \omega))$ and $g_2^{(\delta,\omega)}(z) = -1/(i\delta + (sz + \omega))$. Due to the poles, $|g_{1,2}^{(\delta,\omega)}(x)|$ can still grow as $1/\delta$. To be able to apply Proposition 2 for block-encoding polynomials, we have to ensure that the polynomial that approximates $g^{(\delta,\beta,\omega)}(x)$ has absolute value at most $1/2$ for $x \in [-1, +1]$. That is the reason for including a factor of $\delta/8$ in $g^{(\delta,\beta,\omega)}(x)$ (so that its absolute value is at most $1/4$, and that of its polynomial approximation at most $1/2$ for approximation error less than $1/4$).

Let us first state the following lemma, the proof of which will be provided at the end of this section, which will be used in the proof of Theorem F1 (and thus Theorem 2) on the block encoding of the matrix $G^{(\delta,\beta,\omega)}(h)$.

Lemma F1. For a function $g^{(\delta,\beta,\omega)}(x)$ as in Eq. (F1) (with $\beta, \delta, s > 0$ and $x \in [-1, +1]$), one can efficiently construct a polynomial $p_d(x)$ of (even) degree d such that

$$\max_{x \in [-1,+1]}|g^{(\delta,\beta,\omega)}(x) - p_d(x)|$$

$$\leqslant \begin{cases} \frac{12}{d}\left(\frac{\beta s}{\pi}\right)^4, & \text{if } \frac{\beta s}{2\pi} \geqslant 1, \\ \frac{40}{d}\left(\frac{\beta s}{\pi}\right)^2, & \text{if } \frac{\beta s}{2\pi} < 1. \end{cases} + \begin{cases} \frac{128}{d}\left(\frac{s}{\delta}\right)^4, & \text{if } \frac{2s}{\delta} \geqslant 1, \\ \frac{32}{d}\left(\frac{s}{\delta}\right)^2, & \text{if } \frac{2s}{\delta} < 1. \end{cases}$$
$$\text{(F2)}$$

Theorem F1. For an s-sparse Hamiltonian h on n qubits, assume access to the oracle tuple $\mathcal{O}_h$. We denote the controlled $(1, n+5, \varepsilon_{\mathrm{PA}} + \varepsilon_{p(h)} + \delta_{\mathrm{class}})$-block encoding of $\frac{1}{4}G^{(\delta,\beta,\omega)}(h)$ in Eq. (7) by $C\text{-}U_{G^{(\delta,\beta,\omega)}}$. The implementation of this block encoding requires

$$\begin{cases} \Theta\left(\frac{(\beta s)^4}{\varepsilon_{\mathrm{PA}}}\right), & \text{if } \frac{\beta s}{2\pi} \geqslant 1, \\ \Theta\left(\frac{(\beta s)^2}{\varepsilon_{\mathrm{PA}}}\right), & \text{if } \frac{\beta s}{2\pi} < 1. \end{cases} + \begin{cases} \Theta\left(\frac{s^4}{\delta^4\varepsilon_{\mathrm{PA}}}\right), & \text{if } \frac{2s}{\delta} \geqslant 1, \\ \Theta\left(\frac{s^2}{\delta^2\varepsilon_{\mathrm{PA}}}\right), & \text{if } \frac{2s}{\delta} < 1. \end{cases}$$
$$\text{(F3)}$$

calls to oracles from the oracle tuple $\mathcal{O}_h$,

$$O\left(sn + n_a + \log^{5/2}\left(16s^9(\beta^4 + 1/\delta^4)^2/(\varepsilon_{\mathrm{PA}}^2\varepsilon_{p(h)}^2)\right)\right) \quad \text{(F4)}$$

ancillary qubits, and

$$O\left(n + (n+4)(\beta^4 s^4 + s^4/\delta^4)/\varepsilon_{\mathrm{PA}}\right.$$
$$\left. + \log^{5/2}\left(16s^9(\beta^4 + 1/\delta^4)^2/(\varepsilon_{\mathrm{PA}}^2\varepsilon_{p(h)}^2)\right)\right) \quad \text{(F5)}$$

additional one-qubit and two-qubit gates. To implement this block encoding, an additional classical computing time of $\mathrm{poly}((\beta^4 s^4 + s^4/\delta^4)/\varepsilon_{\mathrm{PA}}, \log(1/\delta_{\mathrm{class}}))$ is required.

Proof. Like in the proof of Theorem D1, we employ Proposition 1 (from Ref. [23]) to construct a $(s, n+3, \varepsilon_{\mathrm{BE}_h})$-block-encoding U_h of h. Using this block encoding, we construct a block encoding of a polynomial approximation of

$G^{(\delta,\beta,\omega)}(h)$. Let $p_d(x)$ denote the degree-d polynomial approximation of the function $g^{(\delta,\beta,\omega)}(x)$ from Lemma F1. It follows from Lemma F1 that one can efficiently construct p_d such that

$$\left|\left|p_d(h/s) - G^{(\delta,\beta)}(\omega,h)\right|\right| \tag{F6}$$

is upper bounded by the right-hand side of the inequality in Eq. (F2). Hence, taking

$$d \leqslant \begin{cases} \Theta\big(\frac{(\beta s)^4}{\varepsilon_{\mathrm{PA}}}\big), & \text{if } \frac{\beta s}{2\pi} \geqslant 1, \\ \Theta\big(\frac{(\beta s)^2}{\varepsilon_{\mathrm{PA}}}\big), & \text{if } \frac{\beta s}{2\pi} < 1, \end{cases} + \begin{cases} \Theta\big(\frac{s^4}{\delta^4 \varepsilon_{\mathrm{PA}}}\big), & \text{if } \frac{2s}{\delta} \geqslant 1, \\ \Theta\big(\frac{s^2}{\delta^2 \varepsilon_{\mathrm{PA}}}\big), & \text{if } \frac{2s}{\delta} < 1, \end{cases} \tag{F7}$$

we obtain $||p_d(h/s) - \delta/8 \, G^{(\delta,\beta)}(\omega,h)|| \leqslant \varepsilon_{\mathrm{PA}}$.

For $\varepsilon_{\mathrm{PA}} \leqslant \frac{1}{4}$, we note that $|p_d(x)| \leqslant 1/2$ for $x \in [-1,+1]$, allowing us to apply Proposition 2 (from Ref. [14]). A $(1, n + 5, 4d\sqrt{\varepsilon_{\mathrm{BE}_h}/s} + \delta)$-block encoding of $p_d(h/s)$ consists of a circuit with $O((n+4)d)$ one-qubit and two-qubit gates, and at most d calls to unitaries U_h, $U_h^\dagger$, or controlled U_h. The classical description of this circuit can be classically computed in $O(\mathrm{poly}(d, \log(1/\delta_{\mathrm{class}})))$ time. We define $\varepsilon_{p(h)} := 4d\sqrt{\varepsilon_{\mathrm{BE}_h}/s}$ so that for a given $\varepsilon_{p(h)}$, we should ensure that $\varepsilon_{\mathrm{BE}_h} = s\varepsilon_{p(h)}^2/(16d^2)$.

Let the $(1, n + 5, \varepsilon_{p(h)} + \delta_{\mathrm{classical}})$-block encoding of $p_d(h/s)$ be denoted by $U_{p_d(h/s)}$. Like in the proof of Theorem D1, we have that $\varepsilon_{\mathrm{Tot}} = \|G^{(\delta,\beta,\omega)}(h) - \langle 0|^{\otimes a} \otimes \mathbb{1} U_{p_d(h/s)} |0\rangle^{\otimes a} \otimes \mathbb{1}\| \leqslant \varepsilon_{\mathrm{PA}} + \varepsilon_{p(h)} + \delta_{\mathrm{class}}$. We have thus constructed a $(1, n + 5, \varepsilon_{\mathrm{PA}} + \varepsilon_{p(h)} + \delta_{\mathrm{class}})$-block encoding of $G^{(\delta,\beta,\omega)}(h)$. To implement this block encoding, we require a number of calls to oracles from the tuple $\mathcal{O}_h$, a number of ancillary qubits, and a number of one-qubit and two-qubit gates as in the lemma statement. ∎

Let us now give the proof of Lemma F1.

Proof. We wish to approximate $g^{(\delta,\beta,\omega)}(x)$ in Eq. (F2) by a polynomial of degree d. Let us first express $g^{(\delta,\beta,\omega)}(x)$ as

$$\delta/8\big((1 - f^{(\beta)}(x))g_1^{(\delta,\omega)}(x) + f^{(\beta)}(x)g_2^{(\delta,\omega)}(x)\big), \tag{F8}$$

and its degree-d polynomial approximation $p_d(x)$ by

$$\delta/8\big((1 - f_{d/2}^{(\beta)}(x))g_{1,d/2}^{(\delta,\omega)}(x) + f_{d/2}^{(\beta)}(x)g_{2,d/2}^{(\delta,\omega)}(x)\big). \tag{F9}$$

Note that

$$|\delta/8 \, g^{(\delta,\beta,\omega)}(x) - p_d(x)|$$
$$\leqslant \delta/8\big(\big|g_1^{(\delta,\omega)}(x) - g_{1,d/2}^{(\delta,\omega)}(x)\big| + \big|g_2^{(\delta,\omega)}(x) - g_{2,d/2}^{(\delta,\omega)}(x)\big|\big)$$
$$+ 1/2\big|f^{(\beta)}(x) - f_{d/2}^{(\beta)}(x)\big|, \tag{F10}$$

where we have used that $|g_{1,d/2}^{(\delta,\omega)}(x)|, |g_{2,d/2}^{(\delta,\omega)}(x)| \leqslant 2/\delta$ for sufficiently large d (note that $|g_1^{(\delta,\omega)}(x)|, |g_2^{(\delta,\omega)}(x)| \leqslant 1/\delta$). Using the bound on $\max_{x \in [-1,+1]} |f^{(\beta)}(x) - f_{d/2}^{(\beta)}(x)|$ from Lemma D1, and applying Bernstein's theorem [31] (i.e., Lemma 1) to the functions $g_1^{(\delta,\omega)}(x)$ and $g_2^{(\delta,\omega)}(x)$ (with a Bernstein ellipse E_r with $r = \sqrt{(\delta/(2s))^2 + 1}$), we obtain the upper bound on $\max_{x \in [-1,+1]} |g^{(\delta,\beta,\omega)}(x) - p_d(x)|$ in the lemma statement. ∎

APPENDIX G: PROOF OF LEMMA 4

Proof of Lemma 4. By assumption, we have that $|\langle i|A|j\rangle - \alpha\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle| \leqslant \varepsilon$, where U_A acts on $n + m$ qubits.

Let us consider estimating $\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle$, which can alternatively be expressed as

$$\langle 0|^{\otimes m}\langle 0|^{\otimes n}(\mathbb{1} \otimes U_i^\dagger)\, U_A\, (\mathbb{1} \otimes U_j)|0\rangle^{\otimes m}|0\rangle^{\otimes n}, \tag{G1}$$

where U_i, U_j are depth-1 circuits that prepare bit strings i and j. We denote the estimate of $\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle$ by $\widetilde{\langle i|A|j\rangle}$, so that if $|\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle - \widetilde{\langle i|A|j\rangle}| \leqslant \tilde{\varepsilon}$, then $|\langle i|A|j\rangle - \alpha\widetilde{\langle i|A|j\rangle}| \leqslant \varepsilon + \alpha\tilde{\varepsilon}$.

One can obtain the estimate $\widetilde{\langle i|A|j\rangle}$ by running a series of Hadamard test circuits on $n + m + 1$ qubits. These circuits correspond to running

$$(\mathbb{1} \otimes [H\, R_z(\theta)]_a)(\mathbb{1} \otimes |0\rangle\langle 0|_a + U \otimes |1\rangle\langle 1|_a)(\mathbb{1} \otimes H_a), \tag{G2}$$

where $U = (U_i^\dagger \otimes \mathbb{1})\, U_A\, (U_j \otimes \mathbb{1})$, on the state $|0\rangle^{\otimes m}|0\rangle^{\otimes n}|0\rangle_a$ (with the final qubit being an ancillary qubit). The output state of the ancillary qubit is measured a total of $D(\tilde{\varepsilon}, \delta)$ times, half of the times for $\theta = 0$ and half of the times for $\theta = \pi/2$. The fractions of output-0 measurements for $\theta = 0$ and $\theta = \pi/2$ provide estimates of $\frac{1}{2} + \frac{1}{2}\mathrm{Re}(\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle)$ and $\frac{1}{2} - \frac{1}{2}\mathrm{Im}(\langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle)$, respectively. Using a Chernoff concentration bound, one can show that $|\widetilde{\langle i|A|j\rangle} - \langle 0|^{\otimes m}\langle i|U_A|0\rangle^{\otimes m}|j\rangle| \leqslant \tilde{\varepsilon}$ with probability at least $1 - \delta$ for $D(\tilde{\varepsilon}, \delta) = \Theta(\tilde{\varepsilon}^{-2}\log(4\delta^{-1}))$.

One can thus obtain an estimate of $\langle i|A|j\rangle$ (given by $\alpha\widetilde{\langle i|A|j\rangle}$) up to error $\varepsilon + \alpha\tilde{\varepsilon}$ with probability $1 - \delta$, using $D(\tilde{\varepsilon}, \delta) = \Theta(\tilde{\varepsilon}^{-2}\log(4\delta^{-1}))$ calls to C-U_A. ∎

APPENDIX H: BQP COMPLETENESS

Here, we prove Theorem 4, using the next Lemma H1 as a small tool:

Proof of Theorem 4. It is straightforward to see that evaluating the matrix element $M_{jj}(t)$ of the correlation matrix $M(t) = e^{iht}M_0e^{-iht}$ at $t = \mathrm{poly}(n)$ is a problem in BQP, given the promise. By Lemmas 4 and E1, given access to $\mathcal{O}_{M_0}$ and $\mathcal{O}_h$ as poly(n)-sized quantum circuits, the problem is solved with poly(n) quantum effort.

To show BQP hardness of our problem, we use the fact that for any promise problem in BQP of problem size m, we have the following property [34]: the problem can be decided by acting on an $k = \mathrm{poly}(m)$-qubit input $|00\cdots 0\rangle$ with (a uniform family of) poly$(k) = \mathrm{poly}(m)$-sized quantum circuits, outputting 1 (on the first qubit) with probability at least $2/3$ in case YES, and 1 with probability at most $1/3$ in case NO. In addition, one can boost the success and failure probabilities $2/3 \to 1 - \exp(-\Theta(k))$ and $1/3 \to \exp(-\Theta(k))$, by running k instances of the poly(k)-sized circuits in parallel and taking a majority vote on the first qubit of the output state for each instance (and copying the answer onto an ancillary qubit). The circuit corresponding to this boosted scenario acts on $q = k^2$ qubits, and its success and failure probabilities are, respectively, $1 - \exp(-\Theta(\sqrt{q}))$ and $\exp(-\Theta(\sqrt{q}))$. Let the quantum circuit for this problem with boosted probabilities be

$$U = W_L \cdots W_1, \tag{H1}$$

where W_l are elementary one-qubit and two-qubit gates and $L = \text{poly}(k) = \text{poly}(\sqrt{q})$. We represent this decision problem using time evolution with a sparse circuit Hamiltonian. The circuit Hamiltonian, acting on a $q_{\text{clock}} = \log_2(L+1)$-qubit clock space [we assume wlog that $\log_2(L+1)$ is an integer] and the q-qubit space is given by

$$h = \sum_{l=1}^{L} (|l+1\rangle\langle l|_{\text{clock}} \otimes W_l + |l\rangle\langle l+1|_{\text{clock}} \otimes W_l^\dagger). \quad \text{(H2)}$$

We take $n = q_{\text{clock}} + q$ and note that $q_{\text{clock}} < q$ for sufficiently large q, so that $n/2 \leqslant q \leqslant n$. The matrices W_l have at most four nonzero entries in a given row/column. Therefore, h is at most eight sparse. Since $\{W_l\}_{l=1}^{L}$ are unitary matrices, the entries of h are $O(1)$ in absolute value.

Consider the evolution $|\psi(t)\rangle = e^{-iht}|1\rangle_{\text{clock}}|00\cdots0\rangle$ with the Hamiltonian h from Eq. (H2). This state can be decomposed as

$$|\psi(t)\rangle = \sum_{l=1}^{L+1} \alpha_{l,t}|l\rangle_{\text{clock}} \otimes \prod_{l'=1}^{l-1} W_{l'}|00\cdots0\rangle, \quad \text{(H3)}$$

with coefficients $\alpha_{l,t}$ given by

$$\sum_{l=1}^{L+1} \alpha_{l,t}|l\rangle \equiv e^{-iJt}|1\rangle_{\text{clock}}, \quad \text{(H4)}$$

where J is a Hamiltonian on the clock register:

$$J = \sum_{l=1}^{L} (|l+1\rangle\langle l|_{\text{clock}} + |l\rangle\langle l+1|_{\text{clock}}). \quad \text{(H5)}$$

Given the encoding of the clock register, one can write the probability of measuring $|L+1\rangle_{\text{clock}}$ on the clock and measuring $|1\rangle$ on the first of the q qubits as

$$p \equiv |(\langle L+1|_{\text{clock}} \otimes \langle 1|_1)|\psi(t)\rangle|^2$$
$$= \langle 1|_{\text{clock}}\langle 00\cdots0|e^{iht}M_0 e^{-iht}|1\rangle_{\text{clock}}|00\cdots0\rangle, \quad \text{(H6)}$$

with $M_0 = \frac{1}{2^{q_{\text{clock}}+1}} \prod_{j=1}^{q_{\text{clock}}}(\mathbb{1} - Z_{\text{clock},j})(\mathbb{1} - Z_{\text{qubit},1})$. Hence, when the state $U|00\cdots0\rangle$ outputs 1 on the first qubit with probability at least $1 - \exp(-\sqrt{q})$ (YES), it follows through Lemma F1 that $p = \Omega(1/\text{poly}(\sqrt{q})) = \Omega(1/\text{poly}(\sqrt{n}))$. When the state $U|00\cdots0\rangle$ outputs 1 on the first qubit with probability at most $\exp(-\sqrt{q})$ (NO), then $p \leqslant \exp(-\sqrt{q}) \leqslant \exp(-\sqrt{n/2})$ through Lemma F1. Now, observe that M_0 is a valid and sparse correlation matrix of a multiparticle free fermionic state on 2^n modes [in particular, a fraction $\Theta(1/\text{poly}(\sqrt{n}))$ of the modes is occupied], which is evolved in time $t = \text{poly}(\sqrt{n})$ by the sparse Hamiltonian h, after which one wishes to estimate a particular matrix element (labeled, say, by $j = 1_{\text{clock}}, 00\cdots0$) of the time-evolved matrix, which is the problem stated in Theorem 4. The only thing left to argue is that given the description of $\{W_l\}$, one can implement $\mathcal{O}_h$ in Definition 2 as a poly(n)-sized circuit.

Oracle implementation. The oracle O_r from Definition 2, acting on $(s+1)(q_{\text{clock}} + q + 1)$ qubits, can be implemented as follows. For convenience, we label the first $(q_{\text{clock}} + q + 1)$ qubits by A and the last s $(q_{\text{clock}} + q + 1)$-qubit registers by $B_1, \ldots, B_s$. For simplicity and wlog, we assume that all W_l are two-qubit gates and all entries of W_l in their two-qubit

subspaces are nonzero. Note that for each $l \in \{1, 2, \ldots, L\}$, we have access to the labels $Q_1^{(l)}$ and $Q_2^{(l)}$ (with $Q_1^{(l)} < Q_2^{(l)}$) of the qubits on which W_l acts nontrivially. The structure of h is such that each row contains eight nonzero entries (apart from the rows associated with clock states $|1\rangle_{\text{clock}}$ and $|L+1\rangle_{\text{clock}}$), with a row $|i\rangle = |l\rangle_{\text{clock}}|x\rangle$ having four nonzero entries associated with clock register state $|l-1\rangle_{\text{clock}}$ and four nonzero entries associated with clock register state $|l+1\rangle_{\text{clock}}$. These entries correspond to the entries $\langle x_{Q_1^{l-1}}, x_{Q_2^{l-1}}|W_{l-1}|y_1, y_2\rangle$ and $\langle x_{Q_1^l}, x_{Q_2^l}|W_l|y_1, y_2\rangle$ (for $y \in \{0, 1\}^2$), respectively. The rows associated with clock states $|1\rangle_{\text{clock}}$ and $|L+1\rangle_{\text{clock}}$ are four sparse.

We take workspace in the form of $2(L+1)$ additional $(q_{\text{clock}} + q)$-qubit registers (initialized in $|00\cdots0\rangle$), denoted by $C_1, \ldots, C_{2(L+1)}$. For each $j \in \{1, 2, \ldots, L+1\}$, we transform the first $(L+1)$ qubits on registers C_{2j-1} and C_{2j} to $|j\rangle_{\text{clock}}$. Then, for each $j \in \{2, 3, \ldots, L\}$ (so excluding 1 and $L+1$), we flip qubits $q_{\text{clock}} + Q_1^{j-1}$ and $q_{\text{clock}} + Q_2^{j-1}$ on register C_{2j-1} and qubits $q_{\text{clock}} + Q_1^j$ and $q_{\text{clock}} + Q_2^j$ on register C_{2j} to $|1\rangle$. In addition, we flip qubits $q_{\text{clock}} + Q_1^1$ and $q_{\text{clock}} + Q_2^1$ on register C_2 and $q_{\text{clock}} + Q_1^L$ and $q_{\text{clock}} + Q_2^L$ on register C_{2L-1} to $|1\rangle$.

Controlled on the clock state on register A being $|l\rangle_{\text{clock}}$, we set the clock state to $|l-1\rangle_{\text{clock}}$ on registers $B_1, \ldots, B_4$ (provided that $l > 1$) and to $|l+1\rangle_{\text{clock}}$ on registers $B_5, \ldots, B_8$ (provided that $l < L+1$). Controlled on the last q qubits of register A being in state $|x\rangle$, we copy $|x\rangle$ onto the final q qubits of $B_1, \ldots, B_4$, excluding qubits $q_{\text{clock}} + Q_1^{l-1}$ and $q_{\text{clock}} + Q_2^{l-1}$. These latter two qubits are transformed to $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ on registers $B_1, \ldots, B_4$, respectively. Similarly, we copy $|x\rangle$ onto the final q qubits of $B_5, \ldots, B_8$, apart from qubits $q_{\text{clock}} + Q_1^l$ and $q_{\text{clock}} + Q_2^l$, which are, respectively, transformed to $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. These operations make use of the states in the workspace registers $C_1, \ldots, C_{2(L+1)}$, which are uncomputed at the end of the protocol. In accordance with Definition 2, we need to account for rows of h having less than eight nonzero entries. Since the rows of h associated with clock states $|1\rangle_{\text{clock}}$ and $|L+1\rangle_{\text{clock}}$ are four sparse, registers $B_1, \ldots, B_4$ are set to, respectively, $|1\rangle \otimes |5\rangle_{q_{\text{clock}}+q}, \ldots, |1\rangle \otimes |8\rangle_{q_{\text{clock}}+q}$ controlled on the A clock state being $|1\rangle_{\text{clock}}$ [after which registers $(B_1, \ldots, B_4)$ and $(B_5, \ldots, B_8)$ are swapped], and registers $B_5, \ldots, B_8$ are set to, respectively, $|1\rangle \otimes |5\rangle_{q_{\text{clock}}+q}, \ldots, |1\rangle \otimes |8\rangle_{q_{\text{clock}}+q}$ controlled on the A clock state being $|L+1\rangle_{\text{clock}}$. The size of the circuit implementing O_r is poly(n).

To implement oracle O_a, let us note that wlog the entries of W_l are $0, \pm 1/\sqrt{2}$, or 1, so that the entries can be encoded into a three-bit string. By employing additional poly(n)-sized workspace [note that $L = \text{poly}(\sqrt{q})$ and each W_l has 16 entries], the oracle O_a can be implemented [by a poly(n)-sized circuit]. ∎

Remark. Like in Ref. [17], we could have adapted the BQP-verification circuit to output the state $|0\rangle_a \otimes |00\cdots0\rangle$ (so all qubits back to their initial state and an additional ancilla qubit a to 0) with high probability in the NO case, and with low probability in the YES case. This is done by simply copying the answer of the BQP circuit onto an additional ancilla qubit a and applying the gates $W_L \cdots W_1$ in reverse on the other qubits. If we use this cleaned-up circuit, it means that we are

interested in estimating the probability for a specific output state—all qubits in $|0\rangle$ and clock state in $|L+1\rangle_{\text{clock}}$—and this corresponds to estimating an entry of a time-evolved rank-1 projector $\tilde{M}_0$, corresponding to a single-particle state. Hence not surprisingly, time evolution of single-particle states is also BQP complete, as was shown in Theorem 3 in Ref. [17] (where more work was done to bring h in sign-free form to directly correspond to a sum of kinetic and potential energies).

The following lemma, which is used in the proof of Theorem 4, mainly follows the approach of Ref. [17]. Instead of employing this lemma, one could also adapt the coefficients in the hopping Hamiltonian h in Eq. (H2) to allow for a perfect one-dimensional state transfer from $|1\rangle_{\text{clock}} \to |L+1\rangle_{\text{clock}}$, using an idea first suggested by Peres [47] (see also Ref. [17]): such adaptation requires extra ancilla qubit overhead in realizing the time dynamics of h; hence, we omit it.

Lemma F1. For a Hamiltonian $J = \sum_{l=1}^{L}(|l\rangle\langle l+1| + |l+1\rangle\langle l|)$ on a $(L+1)$-dimensional Hilbert space with basis states $|l\rangle$, $l \in \{1, \ldots, L+1\}$, there exists a $t = O(L^2 \log L)$ such that

$$|\langle L+1|e^{-iJt}|1\rangle| = \Omega(1/\sqrt{L}). \tag{H7}$$

Proof. The Hamiltonian J has eigenstates

$$|\psi_k\rangle = \sum_{j=1}^{L+1} \alpha_j^{(k)}|j\rangle, \quad \text{with } \alpha_j^{(k)} = \sqrt{\frac{2}{L+2}} \sin\left(\frac{\pi jk}{L+2}\right), \tag{H8}$$

and eigenvalues

$$\epsilon_k = 2\cos\left(\frac{\pi k}{L+2}\right), \tag{H9}$$

with $k = 1 \cdots L+1$. We note that the gap between any two eigenvalues is at most 4. To prove a lower bound on $|\langle L+1|e^{-iJt}|1\rangle|$, we will derive a lower bound on the gaps $\Delta_m := |\epsilon_{m+1} - \epsilon_m|$ (for $m = 1, 2, \ldots L$) between the eigenvalues of J:

$$\Delta_m = |\epsilon_{m+1} - \epsilon_m|$$
$$\geqslant \frac{\pi}{L+2} \min_{x \in [\frac{m\pi}{L+2}, \frac{(m+1)\pi}{L+2}]} \left|\frac{d \, 2\cos(x)}{dx}\right|$$
$$\geqslant \frac{2\pi}{L+2} \sin\left(\frac{\pi}{L+2}\right) = \Omega(1/(L+2)^2). \tag{H10}$$

Using the eigendecomposition of J, we infer that

$$\langle L+1|e^{-iJt}|1\rangle = \frac{2}{L+2} \sum_{k=1}^{L+1} e^{-i\epsilon_k t}(-1)^{k-1} \sin^2\left(\frac{\pi k}{L+2}\right), \tag{H11}$$

so that

$$|\langle L+1|e^{-iJt}|1\rangle|^2 = \left(\frac{2}{L+2}\right)^2 \sum_{k,k'=1}^{L+1} e^{-i(\epsilon_k - \epsilon_{k'})t}(-1)^{k+k'}$$
$$\times \sin^2\left(\frac{\pi k}{L+2}\right) \sin^2\left(\frac{\pi k'}{L+2}\right). \tag{H12}$$

To show that there must be a time t for which $|\langle L+1|e^{-iJt}|1\rangle|^2 = \Omega(1/L)$, we use the fact that a probabilis-

tically chosen time in a sufficiently large interval will give high success probability [35], and hence there must exist a specific time that works sufficiently well. More precisely, for $k \neq k'$, there must exist a probability distribution $\{p(t)\}_{t=0}^{T} \geqslant 0$, $\sum_{t=0}^{T} p(t) = 1$, such that

$$\left|\sum_{t=0}^{T} p(t)e^{-i(\epsilon_k - \epsilon_{k'})t}\right| \leqslant \varepsilon, \tag{H13}$$

provided that $\Delta = \Omega(1/(L+2)^2)$ and $T = O((L+2)^2 \log(1/\varepsilon))$. Examples of probability distributions for which this is true are given in Ref. [48].

Therefore, for those $\{p(t)\}$'s we have that

$$\left|\sum_{k \neq k'} \sum_{t=0}^{T} p(t)e^{-i(\epsilon_k - \epsilon_{k'})t}(-1)^{k+k'}\right.$$
$$\left. \times \sin^2\left(\frac{\pi k}{L+2}\right) \sin^2\left(\frac{\pi k'}{L+2}\right)\right|$$
$$\leqslant \varepsilon \sum_{k \neq k'} \sin^2\left(\frac{\pi k}{L+2}\right) \sin^2\left(\frac{\pi k'}{L+2}\right)$$
$$= \varepsilon\left(\frac{(L+2)^2}{4} - \frac{3(L+2)}{8}\right) \leqslant \varepsilon\frac{(L+2)^2}{4}, \tag{H14}$$

where the equality follows from direct computation. We thus conclude that

$$\left|\sum_{t=0}^{T} p(t)|\langle L+1|e^{-iJt}|1\rangle|^2\right.$$
$$\left. - \sum_{t=0}^{T} p(t)\left(\frac{2}{L+2}\right)^2 \sum_{k=1}^{L+1} \sin^4\left(\frac{\pi k}{L+2}\right)\right| \leqslant \varepsilon. \tag{H15}$$

The term $\sum_{t=0}^{T} p(t)(\frac{2}{L+2})^2 \sum_{k=1}^{L+1} \sin^4(\frac{\pi k}{L+2})$ can be evaluated to be $\frac{3}{2(L+2)}$. So choosing, for instance, $\varepsilon = \frac{1}{2(L+2)}$, we know that $\sum_{t=0}^{T} p(t)|\langle L+1|e^{-iJt}|1\rangle|^2 = \Omega(\frac{1}{L+2})$. For $T = O((L+2)^2 \log(2(L+2)))$, we conclude that there must be a $t = O(L^2 \log L)$ for which $|\langle L+1|e^{-iJt}|1\rangle|^2 = \Omega(1/L)$. ∎

APPENDIX I: CLASSICALLY ESTIMATING ENTRIES OF THE TIME-EVOLVED CORRELATION MATRIX ON LATTICE MODELS

In this Appendix, we briefly argue the following. For $t = \text{poly}(n)$ and assuming classical access to entries $\langle k|M|l\rangle$ of an initial correlation matrix M for given (k, l), one can obtain entries $M(t)_{ij}$ with $1/\exp(n)$ error with $\text{poly}(n)$ classical effort. To see this, note that $\max_{x \in [-1, +1]} |p_K(x) - \exp(itsx)| = O((t/\sqrt{K})^{K+1})$, with $p_K(x)$ a degree-K Taylor approximation. This implies

$$|(p_K(h/s)Mp_K(-h/s))_{ij} - M(t)_{ij}| = O((t/\sqrt{K})^{K+1}), \tag{I1}$$

where we have used that $\|M\| \leqslant 1$. Note that this error can be bounded by $1/\exp(n)$ for some $K = \text{poly}(n)$. Using the same reasoning as in the proof of Lemma 5, we can obtain

$\langle i|h^{k_1}Mh^{k_2}|j\rangle$ for all $k_1, k_2 \leqslant K = \text{poly}(n)$, giving an estimate of $(p_K(h/s)Mp_K(-h/s))_{ij}$. So, for sufficiently large

$K = \text{poly}(n)$, we obtain an estimate of $(e^{+ith}M_0e^{-ith})_{ij}$ with $1/\exp(n)$ error.

---

[1] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[2] B. M. Terhal and D. P. DiVincenzo, Classical simulation of noninteracting-fermion quantum circuits, Phys. Rev. A **65**, 032325 (2002).

[3] E. Knill, Fermionic linear optics and matchgates, arXiv:quant-ph/0108033 [quant-ph].

[4] S. Bravyi, Lagrangian representation for fermionic linear optics, Quantum Inf. Comput. **5**, 216 (2005).

[5] C. W. Groth, M. Wimmer, A. R. Akhmerov, and X. Waintal, Kwant: A software package for quantum transport, New J. Phys. **16**, 063065 (2014).

[6] T. Kloss, J. Weston, B. Gaury, B. Rossignol, C. Groth, and X. Waintal, Tkwant: a software package for time-dependent quantum transport, New J. Phys. **23**, 023025 (2021).

[7] R. Jozsa, B. Kraus, A. Miyake, and J. Watrous, Matchgate and space-bounded quantum computations are equivalent, Proc. R. Soc. A **466**, 809 (2010).

[8] B. Kraus, Compressed quantum simulation of the ising model, Phys. Rev. Lett. **107**, 250503 (2011).

[9] G. Blázquez-Cruz and P.-L. Dallaire-Demers, Quantum supremacy regime for compressed fermionic models, arXiv:2110.09550 [quant-ph].

[10] B. Senjean, S. Yalouz, and M. Saubanère, Toward density functional theory on quantum computers? SciPost Phys. **14**, 055 (2023).

[11] A. Barthe, M. Cerezo, A. T. Sornborger, M. Larocca, and D. García-Martín, Gate-based quantum simulation of Gaussian bosonic circuits on exponentially many modes, Phys. Rev. Lett. **134**, 070604 (2025).

[12] D. W. Berry, A. M. Childs, and R. Kothari, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 2015), pp. 792–809.

[13] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[14] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC '19* (ACM, New York, 2019).

[15] L. Lin, Lecture notes on quantum algorithms for scientific computation, arXiv:2201.08309.

[16] P. Rall, Quantum algorithms for estimating physical quantities using block encodings, Phys. Rev. A **102**, 022408 (2020).

[17] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma, and N. Wiebe, Exponential quantum speedup in simulating coupled classical oscillators, Phys. Rev. X **13**, 041041 (2023).

[18] S. Danz, M. Berta, S. Schröder, P. Kienast, F. K. Wilhelm, and A. Ciani, Calculating response functions of coupled oscillators using quantum phase estimation, arXiv:2405.08694 [quant-ph].

[19] R. D. Somma, R. King, R. Kothari, T. O'Brien, and R. Babbush, Shadow Hamiltonian simulation, Nat. Commun. **16**, 2690 (2025).

[20] A. Montanaro and S. Stanisic, Compressed variational quantum eigensolver for the fermi-hubbard model, arXiv:2006.01179 [quant-ph].

[21] There are straightforward generalizations, using Majorana fermion language, to just parity-conserving free fermion Hamiltonians.

[22] A. Altland and B. D. Simons, *Condensed Matter Field Theory* (Cambridge University Press, UK, 2010), 2nd ed.

[23] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, arXiv:1806.01838 [quant-ph].

[24] K. S. Tikhonov, A. D. Mirlin, and M. A. Skvortsov, Anderson localization and ergodicity on random regular graphs, Phys. Rev. B **94**, 220203 (2016).

[25] C. Vanoni, B. L. Altshuler, V. E. Kravtsov, and A. Scardicchio, Renormalization group analysis of the Anderson model on random regular graphs, Proc. Natl. Acad. Sci. USA **121**, e2401955121 (2024).

[26] I. García-Mata, O. Giraud, B. Georgeot, J. Martin, R. Dubertrand, and G. Lemarié, Scaling theory of the Anderson transition in random graphs: Ergodicity and universality, Phys. Rev. Lett. **118**, 166801 (2017).

[27] I. García-Mata, J. Martin, R. Dubertrand, O. Giraud, B. Georgeot, and G. Lemarié, Two critical localization lengths in the Anderson transition on random graphs, Phys. Rev. Res. **2**, 012020 (2020).

[28] O. Goldreich, S. Goldwasser, and S. Micali, How to construct random functions, J. ACM **33**, 792 (1986).

[29] A. Banerjee, C. Peikert, and A. Rosen, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, Germany, 2012), pp. 719–737.

[30] J. Chen and G. K.-L. Chan, Quantum linear algebra for disordered electrons, arXiv:2411.00434 [quant-ph].

[31] L. N. Trefethen, Convergence for analytic functions, in *Approximation Theory and Approximation Practice, Extended Edition* (SIAM, USA, 2013), Chap. 8, pp. 55–62.

[32] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge Series on Information and the Natural Sciences (Cambridge University Press, Cambridge, UK, 2000).

[33] From this point onward, all considered states are free fermionic, unless stated otherwise.

[34] E. Bernstein and U. Vazirani, Quantum complexity theory, SIAM J. Comput. **26**, 1411 (1997).

[35] D. Nagaj, Local Hamiltonians in quantum computation, Ph.D. thesis, MIT, 2008.

[36] M. B. Hastings, Decay of correlations in fermi systems at nonzero temperature, Phys. Rev. Lett. **93**, 126402 (2004).

[37] C.-F. A. Chen, A. Lucas, and C. Yin, Speed limits and locality in many-body quantum dynamics, Rep. Prog. Phys. **86**, 116001 (2023).

[38] M. C. Tran, C.-F. Chen, A. Ehrenberg, A. Y. Guo, A. Deshpande, Y. Hong, Z.-X. Gong, A. V. Gorshkov, and A. Lucas, Hierarchy of linear light cones with long-range interactions, Phys. Rev. X **10**, 031009 (2020).

[39] P. C. S. Costa, S. Jordan, and A. Ostrander, Quantum algorithm for simulating the wave equation, Phys. Rev. A **99**, 012323 (2019).

[40] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond quadratic speedups for error-corrected quantum advantage, PRX Quantum **2**, 010103 (2021).

[41] A. Kollár, M. Fitzpatrick, P. Sarnak, and A. Houck, Line-graph lattices: Euclidean and non-Euclidean flat bands, and implementations in circuit quantum electrodynamics, Commun. Math. Phys. **376**, 1909 (2020).

[42] S. Bravyi and R. Koenig, Classical simulation of dissipative fermionic linear optics, Quantum Inf. Comput. **12**, 925 (2012).

[43] K. Sakamoto and K. Fujii, On the quantum computational complexity of classical linear dynamics with geometrically local interactions: Dequantization and universality, arXiv:2505.10445 [quant-ph].

[44] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Quantum Metropolis sampling, Nature (London) **471**, 87 (2011).

[45] J. Jiang and S. Irani, Quantum metropolis sampling via weak measurement, arXiv:2406.16023 [quant-ph].

[46] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nat. Phys. **16**, 1050 (2020).

[47] A. Peres, Reversible logic and quantum computers, Phys. Rev. A **32**, 3266 (1985).

[48] S. Boixo, E. Knill, and R. Somma, Eigenpath traversal by phase randomization, Quantum Inf. Comput. **9**, 833 (2009).