

# Fault-Tolerant Quantum Simulation of Materials Using Bloch Orbitals

Nicholas C. Rubin,<sup>1,\*</sup> Dominic W. Berry,<sup>2,†</sup> Fionn D. Malone,<sup>1</sup> Alec F. White,<sup>3</sup> Tanuj Khattar,<sup>1</sup>  
A. Eugene DePrince, III,<sup>1,4</sup> Sabrina Sicolo,<sup>5</sup> Michael Kühn,<sup>5</sup> Michael Kaicher,<sup>5</sup> Joonho Lee,<sup>1,6</sup>  
and Ryan Babbush<sup>1,‡</sup>

<sup>1</sup>Google Research, Venice, California 90291, USA

<sup>2</sup>School of Mathematical and Physical Sciences, Macquarie University, Sydney, NSW 2109, Australia

<sup>3</sup>Quantum Simulation Technologies, Inc., Boston 02135, USA

<sup>4</sup>Department of Chemistry and Biochemistry, Florida State University, Tallahassee, Florida, USA

<sup>5</sup>BASF SE, Carl-Bosch-Strasse 38, Ludwigshafen 67063, Germany

<sup>6</sup>Department of Chemistry, Columbia University, New York, New York, USA



(Received 10 February 2023; accepted 18 August 2023; published 6 October 2023)

The simulation of chemistry is among the most promising applications of quantum computing. However, most prior work exploring algorithms for block encoding, time evolving, and sampling in the eigenbasis of electronic structure Hamiltonians has either focused on modeling finite-sized systems, or has required a large number of plane-wave basis functions. In this work, we extend methods for quantum simulation with Bloch orbitals constructed from symmetry-adapted atom-centered orbitals so that one can model periodic *ab initio* Hamiltonians using only a modest number of basis functions. We focus on adapting existing algorithms based on combining qubitization with tensor factorizations of the Coulomb operator. Significant modifications of those algorithms are required to obtain an asymptotic speedup leveraging translational (or, more broadly, Abelian) symmetries. We implement block encodings using known tensor factorizations and a new Bloch orbital form of tensor hypercontraction. Finally, we estimate the resources required to deploy our algorithms to classically challenging model materials relevant to the chemistry of lithium nickel oxide battery cathodes within the surface code. We find that even with these improvements, the quantum runtime of these algorithms is on the order of thousands of days and further algorithmic improvements are required to make realistic quantum simulation of materials practical.

DOI: [10.1103/PRXQuantum.4.040303](https://doi.org/10.1103/PRXQuantum.4.040303)

## I. INTRODUCTION

Recently, the first quantization quantum algorithms and constant factor resource estimation analysis for molecular systems [1,2] have been adapted to materials [3]. While the first quantization approach using a plane-wave representation is attractive due to the smooth convergence to the continuum limit [4,5] a local basis representation such as atom-centered basis sets has other advantages. Similar to the molecular simulation setting, local basis functions can be advantageous when describing spatially localized phenomena such as heterogeneous catalysis or

efficiently describing cusps [6]. Moreover, local bases are often more compact than plane waves and are thus more practical for high-accuracy wave function or embedding based approaches, which are increasingly being applied to crystalline solids. As a result, it is important to explore how advances in classical second quantized wave-function methods can be adopted for use in quantum algorithms [7].

The desire for systematically improvable electronic structure methods to treat the many examples of strongly correlated phenomena [8–10] in the condensed phase has recently driven the application of *ab initio* wave-function theories to the periodic setting [11–23]. Standard treatments of symmetry in wave-function theories [24,25] can be used to exploit the translational symmetry of periodic systems, thus enabling the application of post-Hartree-Fock methods to material systems. Despite these advantages, classical *ab initio* treatment of such problems is limited due to the large simulation cells needed to converge to the thermodynamic limit. This drawback has further driven the use of embedding theories [26–31] and downfolding [32]. Naturally, one may ask if fault-tolerant

\*Corresponding authors. [nickrubin@google.com](mailto:nickrubin@google.com)

†[dominic.berry@mq.edu.au](mailto:dominic.berry@mq.edu.au)

‡[babbush@google.com](mailto:babbush@google.com)

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

quantum computers can alleviate the computational burden associated with *ab initio* simulation of solids within the local basis framework.

In this paper, we describe how to extend molecular quantum simulation algorithms of second quantization Hamiltonians represented in local basis sets to periodic systems using the qubitization framework [33,34]. Though the general structure of the algorithms is largely unchanged, introducing symmetry—i.e., symmetry adapting the block encodings—requires nontrivial modifications to realize an improvement in the asymptotic complexity. The first steps in this direction were taken in Ref. [35] using the “sparse” Hamiltonian representation. We provide an alternative derivation for block encodings using this representation and introduce symmetry-adapted block encodings for three other more performant tensor factorizations of the Hamiltonian: single factorization (SF), double factorization (DF), and tensor hypercontraction (THC). The result of these developments is orders of magnitude improvement in the quantum resources required to simulate materials compared to straightforward sparse approaches presented in Ref. [35]. Incorporating symmetry into the block encoding of the Hamiltonian requires efficiently taking advantage of the selection rule between irreducible representation labels of the single-particle basis. Simply adapting the existing algorithms to iterate over the additional indices does not lead to the optimal scaling and would not provide a speedup.

For each of the four Hamiltonian representations we describe the origin of the asymptotic speedup (or lack thereof in one case), provide compiled algorithms for constant factor resource estimates, and compare the performance to nonsymmetry-adapted block encodings. We note that the derived symmetry-adapted block encodings apply to any Abelian point-group symmetry with minor modifications. For SF, sparse, and DF the symmetry-adapted block encodings provide an asymptotic speedup for walk operator construction proportional to the square root of the number of  $k$  points used to sample the Brillouin zone. For THC, there is no asymptotic improvement due to the linear cost of unary iteration in the block encoding. Going beyond asymptotic analysis and compiling to total Toffolis, we find that for DF and THC using symmetry-adapted block encodings provides no asymptotic speedup over their nonsymmetry-adapted counterparts due to the increased number of applications of the walk operator for fixed precision phase estimation. DF and THC are sensitive to the numerical compression of the Hamiltonian, and thus we expect the number of walk operator applications can be decreased. Moreover, there are significant classical advantages to using the symmetry-adapted block encodings coming from the reduced classical complexity of representing the Hamiltonian as the system size is increased towards the thermodynamic limit. Apart from these practical considerations, the incorporation of

symmetry significantly expands the scope of quantum simulation of materials. Indeed, the understanding of physical phenomena in materials is intimately related to symmetry and symmetry breaking. For example, understanding spin-orbit coupling [36], topological insulators [37], electron-phonon interactions [38], thermal and electrical conductivity [39,40], and superconductivity [41] is either impossible or extremely challenging without incorporating symmetry labels into the theoretical description. Here we present a general framework for incorporating symmetry adaptation into quantum algorithms.

As was the case in the molecular setting [33], the quantum algorithms in this work rely heavily on unary iteration, (advanced) quantum read-only memory QRO(A)M [42], and coherent alias sampling [43]. The leading-order gate complexity for these algorithms mostly comes from the need to load data from QROAM, which has a complexity that scales like  $\mathcal{O}(\sqrt{\Gamma})$  where  $\Gamma$  is amount of data needed to specify the Hamiltonian. At a high level, incorporating symmetry leads to a reduction in the amount of information required to specify the Hamiltonian by  $N_k$ , where  $N_k$  is the number of  $k$  points used to sample the Brillouin zone, and this is the source of the  $\mathcal{O}(\sqrt{N_k})$  speedups derived in this paper. In the nonsymmetry-adapted case the origin of the highest complexity subroutine is different for every Hamiltonian factorization. We consider four representations of the symmetry-adapted two-electron integral tensor in this work: one is a seven tensor containing orbital and symmetry labels, one is a five tensor containing two symmetry labels, two orbital labels, and an auxiliary label, and two hypercontraction representations. In the first representation simply iterating over the seven-index representation of the electron interaction term is sufficient for providing the  $\sqrt{N_k}$  improvement. In the five-index representation we modify the molecular SF algorithm by outputting the irreducible representation label in the outer prepare so that we can compute the irreducible representation labels for the operator to be squared on the fly. Without this additional information output and the modular computation of the symmetry label the correct asymptotic scaling would not be achieved. Finally, in the two-tensor hypercontraction representations we further compress the representation and also introduce an ancilla work space, which limits the number of Givens rotations we need to output data for. Further details are enumerated in each Hamiltonian representation section of Sec. III.

In parallel with recent studies estimating quantum resources required to simulate high-value molecular targets [33,44–46], we estimate the quantum resources required to simulate an open materials science problem related to the cathode structure of lithium nickel oxide (LNO) batteries. The LNO systems are universally observed in the high-symmetry  $R3m$  structure, which is at odds with the predicted Jahn-Teller activity of low-spin trivalent Ni [47]; more background can be found in Sec. V A. This

discrepancy combined with the difficulty of synthesizing pure LNO, the size of the unit cells [48], and potential strong correlation at the high-symmetry structure [49] makes the LNO problem an interesting application target for quantum simulation advantage. This realistic problem frames the algorithmic improvements articulated in this paper and the prospects of the quantum advantage given modern electronic structure methods. We find that the required resource estimates for simulating a set of benchmark systems and the LNO problem before reaching the thermodynamic limit are already substantial. In fact, the large simulation cells required to converge these calculations to the thermodynamic limit is ultimately a significant hurdle for *ab initio* simulations.

The layout of the rest of the paper is as follows: Sec. II describes the atom-centered basis sets and the Hamiltonian that we use, Sec. III describes the qubitization algorithm and the origin of the asymptotic speedup in constructing walk operators using each of the four Hamiltonian representations. Each subsection is dedicated to a particular Hamiltonian factorization and describes the qubitization algorithm and how to calculate associated parameters. Section IV compares all methods and extrapolates quantum resources required to simulate a diamond crystal converged towards the thermodynamic limit, and Sec. V reports the accuracy and correlation analysis of various electronic structure methods for LNO while providing estimates of quantum computing resources and runtimes. We close with prospects for this class of methods.

## II. ELECTRONIC STRUCTURE HAMILTONIAN OF MATERIALS IN BLOCH ORBITALS

Although plane-wave basis sets are used in most periodic density-functional-theory (DFT) calculations, there is a long history of local-basis methods as well. The use of a localized basis set has a number of advantages over plane waves: (1) 0D (molecular), 1D, 2D, and 3D systems can be treated on an equal computational footing. (2) Calculations on low-density systems with large unit cells can be more efficient [50–52]. (3) Hartree-Fock exchange can be more efficiently computed in the smaller, local-orbital basis [51–54]. (4) The local-orbital representations can lower the computational cost of correlation corrections with a more compact representation of the virtual space. (1)–(3) have spurred the development of local-orbital DFT and Hartree-Fock methods with Gaussian orbitals [51–53,55] and numerical atomic orbitals [56], while (4) has been behind recent work to apply correlated electronic structure theory to periodic solids [11,12,14–18]. In the following subsection, we describe the symmetry-adapted periodic sum of Gaussian-type orbitals used in this work.

### A. Basis functions and matrix elements

A local basis function,  $\tilde{\chi}_p$ , can be adapted to the translational symmetry of a lattice to form a periodized function

$$\chi_{p,\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{T}} e^{i\mathbf{k}\cdot\mathbf{T}} \tilde{\chi}_p(\mathbf{r} - \mathbf{T}), \quad (1)$$

where  $\mathbf{T}$  represents a lattice translation vector and  $\mathbf{k}$  is a crystal momentum vector lying in the first Brillouin zone. The lattice momentum  $\mathbf{k}$  labels an irreducible representation of the group of translations defined by the translational symmetry of the material. Functions of this form are easily verified to be Bloch functions in that

$$\chi_{p,\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} u_{p,\mathbf{k}}(\mathbf{r}), \quad (2)$$

where  $u_{p,\mathbf{k}}(\mathbf{r})$  has the same periodicity as the lattice.

Orbitals are constructed from a linear combination of the underlying Bloch orbitals,

$$\phi_{i\mathbf{k}}(\mathbf{r}) = N_k^{-1/2} \sum_p c_{p,i}(\mathbf{k}) \chi_{p,\mathbf{k}}(\mathbf{r}), \quad (3)$$

where  $N_k$  is the total number of  $k$  points. The expansion coefficients,  $c_{p,i}(\mathbf{k})$  are determined from the appropriate periodic self-consistent field procedure, usually Hartree-Fock or Kohn-Sham DFT. The resulting orbitals are normally constrained to be orthogonal by convention and can serve as a basis for representing the second-quantized Hamiltonian.

The matrix elements of a one-electron operator,

$$T_{p\mathbf{k}_p,q\mathbf{k}_q} = \int dr \phi_{p\mathbf{k}_p}^*(r) \mathcal{O}_1 \phi_{q\mathbf{k}_q}(r) \quad (4)$$

are nonzero only when  $\mathbf{k}_p = \mathbf{k}_q$  as long as  $\mathcal{O}_1$  has the translational symmetry of the lattice. We can use a similar strategy to derive the structure of the two-electron integrals, which are given by

$$V_{p\mathbf{k}_p,q\mathbf{k}_q,r\mathbf{k}_r,s\mathbf{k}_s} = \int \int dr_1 dr_2 \phi_{p\mathbf{k}_p}^*(\mathbf{r}_1) \phi_{q\mathbf{k}_q}(\mathbf{r}_1) \mathcal{O}_2 \phi_{r\mathbf{k}_r}^*(\mathbf{r}_2) \phi_{s\mathbf{k}_s}(\mathbf{r}_2). \quad (5)$$

The translational symmetry of the Bloch orbitals implies the two-electron operator  $\mathcal{O}_2$  matrix elements can only be nonzero when  $(\mathbf{k}_p + \mathbf{k}_r - \mathbf{k}_q - \mathbf{k}_s) = \mathbf{G}$  where  $\mathbf{G}$  is a reciprocal lattice vector. We note that this expression for nonzero matrix elements by symmetry is a specific instance of the more general expression. More generally, given a group  $\mathfrak{g}$  with its irreducible representations labeled by  $\{\Gamma_i\}$ , the two-electron integral is nonzero by symmetry whenever  $\Gamma_p \otimes \Gamma_q \otimes \Gamma_r \otimes \Gamma_s$  contains the complete symmetric representation [24]. For periodic systems  $\mathfrak{g}$  is the set of

translational symmetries. Despite this sparsity, the evaluation of the nonzero matrix elements for all basis functions is often a major computational bottleneck whenever local basis sets are used.

Local orbitals provide a more compact representation than plane waves, so fewer basis functions are needed. Unfortunately, there are  $O(N_k^3 N^4)$  generally nonzero two-electron matrix elements for  $N_k$   $k$  points and  $N$  basis functions in the primitive cell. For very large calculations locality can be exploited to yield asymptotically linear-scaling DFT methods [57,58]. Linear-scaling Hartree-Fock is also possible for insulators [59]. This linear regime is almost never reached in practice, and it is usually advantageous to instead reduce the cost by tensor factorization.

Although our discussion has been thus far general with regard to the choice of local basis functions, Gaussian basis functions are by far the most popular choice in molecular calculations, and crystalline Gaussian orbitals are also a popular choice for periodic calculations. This popularity is due to the existence of analytic formulas, which allow for fast, numerically exact evaluation of the matrix elements of most common operators. Despite the existence of efficient numerical techniques, the large number of two-electron integrals that must be evaluated in periodic calculations requires a more efficient procedure. Traditionally, this is accomplished with the Gaussian plane-wave (GPW) method [50,60], which requires only storage of  $O(N_k^2 N^2 n_{pw})$  integrals where  $n_{pw}$  is the number of plane waves used to evaluate the integrals. In molecular calculations, the most common decomposition is called the resolution of the identity (RI) or sometimes density fitting (DF) [61–64]. This procedure requires the storage of  $O(N_k^2 N^2 n_{aux})$  integrals where  $n_{aux}$  is the size of the auxiliary basis set. Both the GPW and the RI method can be considered as density-fitting approaches where the former uses a plane-wave fitting basis and the latter uses a Gaussian fitting basis. For this reason, the RI approach is often called “Gaussian density fitting” (GDF) in the context of periodic calculations [65–69].

The two-electron integral tensor can be further factorized into a product of five two-index tensors as was done in the tensor hypercontraction (THC) method of Martinez and coworkers [70–72]. Factorizations of this form are most useful for correlated methods where they have the potential to lower the computational scaling. In this work we present a translational symmetry-adapted form of the tensor hypercontraction for the two-electron integral tensors of periodic systems.

## B. The second-quantized Hamiltonian

We can express the second-quantized electronic structure Hamiltonian as

$$H = H_1 + H_2, \quad (6)$$

$$H_1 = \sum_{\sigma} \sum_{\mathbf{k}} \sum_{pq} h_{p\mathbf{k},q\mathbf{k}} a_{p\mathbf{k}\sigma}^{\dagger} a_{q\mathbf{k}\sigma}, \quad (7)$$

$$h_{p\mathbf{k},q\mathbf{k}} = T_{p\mathbf{k},q\mathbf{k}} - \frac{1}{2} \sum_{r,\mathbf{Q}} V_{p\mathbf{k},r\mathbf{Q},r\mathbf{Q},q\mathbf{k}}, \quad (8)$$

$$H_2 = \frac{1}{2} \sum_{\sigma,\tau} \sum_{\mathbf{Q},\mathbf{k},\mathbf{k}'} \sum_{pqrs} V_{p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q}),r(\mathbf{k}'\ominus\mathbf{Q}),s\mathbf{k}'} a_{p\mathbf{k}\sigma}^{\dagger} a_{q(\mathbf{k}\ominus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\ominus\mathbf{Q})\tau}^{\dagger} a_{s\mathbf{k}'\tau}. \quad (9)$$

We first introduce summation limits for each symbol as we will commonly use short-hand summation formulas to indicate multiple sums. For each variable  $\{p, q, r, s\}$ , summation is performed over the range  $[0, N/2 - 1]$  indexing the spatial orbital or band,  $\{\mathbf{Q}, \mathbf{k}, \mathbf{k}'\}$  summation is performed over the Brillouin zone (BZ) at a set number of  $k$  points of which there are  $N_k$ , and  $\{\sigma, \tau\}$  are electron spin variables and summed over  $\{\uparrow, \downarrow\}$ . Nonmodular differences of  $\mathbf{k}$ ,  $\mathbf{Q}$ , and  $\mathbf{k}'$  span twice the Brillouin zone. Because  $V$  needs to be indexed by values in the Brillouin zone, we use modular subtraction indicated by  $\ominus$ . That is, if the number of points in each dimension is  $N_x, N_y, N_z$ , we perform subtraction modulo  $N_x, N_y, N_z$  in each direction, respectively.

The Hamiltonian is generally complex Hermitian with fourfold symmetry of the two-electron integrals. We note that the following generic complex Coulomb integral symmetries are present

$$\begin{aligned} V_{p\mathbf{k}_p,q\mathbf{k}_q,r\mathbf{k}_r,s\mathbf{k}_s} &= V_{r\mathbf{k}_r,s\mathbf{k}_s,p\mathbf{k}_p,q\mathbf{k}_q} = V_{q\mathbf{k}_q,p\mathbf{k}_p,s\mathbf{k}_s,r\mathbf{k}_r}^* \\ &= V_{s\mathbf{k}_s,r\mathbf{k}_r,q\mathbf{k}_q,p\mathbf{k}_p}^* \end{aligned} \quad (10)$$

from integration index relabeling and complex conjugation. In the following sections, we demonstrate how the sparse structure of the two-electron integral tensor affects the scaling of block encoding the Hamiltonian for implementation of qubitized quantum walk oracles. The cost of qubitization is greatly affected by the representational freedom of the underlying Hamiltonian expressed as a linear combination of unitaries. We demonstrate how to construct the sparse, single-factorization (SF), double-factorization (DF), and tensor-hypercontraction (THC) integral decompositions of Bloch orbital Hamiltonians and analyze the cost of such simulations for a variety of materials.

For all algorithms, we will make a comparison to the case of a  $\Gamma$ -point calculation on a supercell composed of  $N_k$  primitive cells in the geometry described by the  $k$ -point sampling. This allows us to directly observe the proposed speedup due to symmetry adapting. To demonstrate the scaling of symmetry-adapted block encoding, we estimate quantum simulation resource requirements for the series of systems listed in Table I. Range-separated density fitting [69] is used to construct integrals with



TABLE I. Crystal structures lattice parameters used for the systems studied in this work. The lattice parameters were chosen to be at or near their experimental equilibrium values.

System	Structure	Atoms in cell	Lattice parameters	Spin orbitals cc-pVDZ	Spin orbitals cc-pVTZ
C	diamond	2	3.567 [75]	52	116
Si	diamond	2	5.43 [75]	52	116
BN	zinc blende	2	3.616 [75]	52	116
LiCl	rocksalt	2	5.106 [76]	52	98
AlN	wurzite	4	(a) 3.11 (c) 4.981 [75]	104	220
Li	bcc	2	3.51 [77]	52	80
Al	fcc	2	4.0479 [78]	52	104

Dunning-type correlation-consistent basis sets [73] and the Goedecker-Teter-Hutter (GTH) family of pseudopotentials for Hartree-Fock [74]. For each Hamiltonian, cutoffs for the factorization are selected so that the Møller-Plesset second-order perturbation theory (MP2) error in the total energy is below one millihartree per cell or formula unit depending on the system. While prior works used coupled-cluster theory, MP2 is used here for computational efficiency.

### III. QUBITIZATION OF MATERIALS HAMILTONIANS

Similar to fault-tolerant resource estimates for molecular systems represented in second quantization [2,33,45,46,79], we compare the number of logical qubits and number of Toffoli gates required to implement phase estimation on unitaries that use block encoding [80] and qubitization [34] to encode the Hamiltonian spectrum in a Szegedy walk operator [81] for various linear combination of unitary (LCU) [82] representations of the Hamiltonian. All LCUs represent the Hamiltonian as

$$H = \sum_{\ell=1}^L \omega_{\ell} U_{\ell}, \quad (11)$$

where  $\omega_{\ell} \in \mathbb{R}$ ,  $\omega_{\ell} \geq 0$ , and  $U_{\ell}$  is a unitary operator. One can then construct the operators

$$\text{PREPARE}|0\rangle^{\otimes \log(L)} \mapsto \sum_{\ell=1}^L \sqrt{\frac{\omega_{\ell}}{\lambda}} |\ell\rangle \equiv |\mathcal{L}\rangle, \quad (12)$$

$$\text{SELECT}|\ell\rangle|\psi\rangle \mapsto |\ell\rangle U_{\ell}|\psi\rangle, \quad (13)$$

$$\lambda = \sum_{\ell=1}^L \omega_{\ell}, \quad (14)$$

where  $|\psi\rangle$  is the system register, and  $|\ell\rangle$  is an ancilla register used to index each term in the LCU. The walk operator constructed from SELECT and a reflection operator built from PREPARE,  $R = 2|\mathcal{L}\rangle\langle\mathcal{L}| \otimes \mathbb{1} - \mathbb{1}$ , has eigenvalues proportional  $e^{\pm i \arccos E_n/\lambda}$  where  $E_n$  is an eigenvalue of the Hamiltonian in Eq. (11).

It was shown in Refs. [2] and [33] when ensuring that SELECT is self-inverse, only the reflection operator  $R$  needs to be controlled on the ancilla for phase estimation and not SELECT. Therefore, the Toffoli cost of phase estimating the walk operator scales as

$$\left\lceil \frac{\pi\lambda}{2\epsilon_{\text{PEA}}} \right\rceil (C_S + C_P + C_{P^\dagger} + \log(L)), \quad (15)$$

where  $C_S$  is the cost for implementing the SELECT oracle and  $C_P$  is the cost for implementing the PREPARE oracle,  $C_{P^\dagger}$  is the cost for the inverse PREPARE oracle, and  $\epsilon_{\text{PEA}}$  is the target precision for phase estimation. Thus the main costs for sampling from the eigenspectrum of a second quantized operator are the costs to implement SELECT, PREPARE, and PREPARE $^\dagger$ . These costs need to be multiplied by a factor proportional to  $\lambda/\epsilon_{\text{PEA}}$  for the number of walk steps needed for phase estimation. Note that when computing intensive quantities, such as the energy per cell, the  $\lambda$  factor is scaled by  $1/N_k$ .

The particular choice of LCU changes all of these costs. Prior works have investigated the resource requirements for simulating molecules with four different LCUs. While all these methods can be used without modification in supercell calculations at the  $\Gamma$  point, the construction of molecular SELECT and PREPARE do not exploit any symmetries and are not applicable away from the  $\Gamma$  point—e.g., at the Baldereschi point [83].

The leading costs in constructing SELECT and PREPARE for second-quantized Hamiltonians is the circuit primitive that functions similar to a read-only-memory (ROM) called QROM. The QROM primitive is a gadget that takes a memory address, potentially in superposition, and outputs data, also potentially in superposition. There are currently two variations of QROM that have different costs; traditional QROM that has linear Toffoli complexity when outputting  $L$  items with any amount of data associated with each item, and advanced QROM (called QROAM) with reduced non-Clifford complexity [84]. It uses a select-swap circuit construction with Toffoli cost

$$\left\lceil \frac{L}{k} \right\rceil + m(k-1), \quad (16)$$

TABLE II. Generically, qubitized quantum walks scale as  $\mathcal{O}(\sqrt{\Gamma})$  in space and  $\mathcal{O}(\lambda\sqrt{\Gamma}/\epsilon)$  in time where  $\Gamma$  is the amount of information required to specify the Hamiltonian within a particular representation. For double factorization,  $\Xi$  is the sum of the average rank of the second factorization, which is expected to scale as  $\mathcal{O}(N_k N)$ , which is the number of orbitals in the primitive cell or bands.  $\tilde{\Xi}$  is the average rank of the second factorization in the supercell calculation and is also expected to scale as  $\mathcal{O}(N_k N)$ . The tilde on  $\mathcal{O}$  is used to account for logarithmic factors, and can include variables not explicitly given in the scaling.  $\lambda$  for each LCU is different and is denoted as a subscript indicating the LCU type and if it is  $\lambda$  for the supercell version.

Representation	Qubits	Toffoli complexity	SC qubits	SC Toffoli
sparse	$\tilde{\mathcal{O}}(N_k^{3/2} N^2)$	$\tilde{\mathcal{O}}(N_k^{3/2} N^2 \lambda_{\text{sparse}}/\epsilon)$	$\tilde{\mathcal{O}}(N_k^2 N^2)$	$\tilde{\mathcal{O}}(N_k^2 N^2 \lambda_{\text{sparse,SC}}/\epsilon)$
SF	$\tilde{\mathcal{O}}(N_k N^{3/2})$	$\tilde{\mathcal{O}}(N_k N^{3/2} \lambda_{\text{SF}}/\epsilon)$	$\tilde{\mathcal{O}}(N_k^{3/2} N^{3/2})$	$\tilde{\mathcal{O}}(N_k^{3/2} N^{3/2} \lambda_{\text{SF,SC}}/\epsilon)$
DF	$\tilde{\mathcal{O}}(\sqrt{N_k} N \sqrt{\Xi})$	$\tilde{\mathcal{O}}(\sqrt{N_k} N \sqrt{\Xi} \lambda_{\text{DF}}/\epsilon)$	$\tilde{\mathcal{O}}(N_k N \sqrt{\tilde{\Xi}})$	$\tilde{\mathcal{O}}(N_k N \sqrt{\tilde{\Xi}} \lambda_{\text{DF,SC}}/\epsilon)$
THC	$\tilde{\mathcal{O}}(N_k N)$	$\tilde{\mathcal{O}}(N_k N \lambda_{\text{THC}}/\epsilon)$	$\tilde{\mathcal{O}}(N_k N)$	$\tilde{\mathcal{O}}(N_k N \lambda_{\text{THC,SC}}/\epsilon)$

where  $k$  is a power of 2 for outputting  $L$  items of data where each item of data is  $m$  bits long. The notation  $k$  here for an integer should not be confused with  $\mathbf{k}$  for the crystal momentum vector. It needs  $m(k-1)$  ancillas, so increases the logical ancilla count in exchange for reduced Toffoli complexity. Given that  $L \gg m$  in the case of periodic calculations, we can minimize this function by selecting  $k \approx \sqrt{L/m}$  and thus the Toffoli and ancilla cost generically go as  $\mathcal{O}(\sqrt{Lm})$ . It is also possible to adjust  $k$  to reduce the ancilla count while increasing the Toffoli count. Having QROAM output the minimal amount of information to represent the Hamiltonian is at the core of the  $\sqrt{N_k}$  improvements we derive in many of the block encodings. We will also demonstrate that for all LCUs the lowest scaling can be linear in the Bloch orbital basis size  $\mathcal{O}(N_k N)$  due to the requirement to perform unary iteration at least once over the entire basis.

Another primitive that becomes the dominant cost in constructing symmetry-adapted SELECT is the multiplexed-controlled swap between two registers. The controlled swap between two registers uses unary iteration [2] on  $L$  items to swap  $M$  elements between two registers at the cost of  $\mathcal{O}(LM)$  Toffolis. For simulating materials, this primitive is commonly encountered when swapping all band indices with a particular irreducible representation label, or  $k$  point, into a working register at a cost of  $\mathcal{O}(N_k N)$ . The necessity of coherently moving data thus puts a limit on the total savings one can achieve by leveraging Abelian symmetries. The cost of moving data must be weighed against the benefits, which we describe in each section below. In Table II we summarize the space complexity, in terms of logical qubits, and time complexity, in terms of Toffolis of the four LCUs when considering translational symmetry on the primitive cell and without (denoted as SC for supercell).

For the sparse LCU, exploiting primitive cell translational symmetry reduces the amount of symmetry unique information in the Hamiltonian by a factor of  $N_k$ , which translates to a reduction of  $\sqrt{N_k}$  savings in Toffoli complexity and ancilla complexity. For sparse PREPARE, the square-root savings originates from the QROAM cost of

outputting “alt” and “keep” values for the coherent alias sampling component of the state preparation. For sparse SELECT controlled application of all Pauli terms has linear cost in the basis size  $\mathcal{O}(N_k N)$  and is not the dominant cost. The supercell calculation does not exploit the  $k$ -point symmetry of the Hamiltonian matrix elements and thus has worse scaling.

The single factorization LCU leverages the fact that the Coulomb integral tensor is positive semidefinite and can be written in a quadratic form. For molecular systems without symmetry—i.e.,  $C_1$  symmetry, the factorization results in a three tensor where there are two orbital indices and one auxiliary index that scales as the number of orbitals in the system [85]. For simulations where orbitals now have point-group symmetry labels, such as  $k$  points, each three-tensor factor can now be arranged into a five-tensor; two symmetry labels (irreducible representation labels), two-band labels (orbital labels), and one auxiliary index, which still scales with the number of bands due to density fitting of the cell periodic part of the density [69,86], i.e.,

$$(pq|rs) = \sum_n L_{pq,n} L_{sr,n}^* \xrightarrow{\text{symmetry}} \sum_n L_{\mathbf{k}_p \mathbf{k}_q p q n} L_{\mathbf{k}_s \mathbf{k}_r s r n}^* \quad (17)$$

Thus the origin of the  $\sqrt{N_k}$  improvement for the symmetry-adapted block encoding with a single-factorization LCU lies in the fact that the auxiliary index has  $N_k$  lower scaling in comparison to a supercell variation where the Cholesky factorization or density fitting is performed on the entire supercell two-electron integral tensor. The single factorization algorithm is also dominated by the QROM cost of PREPARE—of which there are two state preparations. The inner-state preparation [33] for the  $k$ -point symmetry-adapted algorithm requires outputting  $\mathcal{O}(N_k^2 N^3)$  to be used in the state preparation leading to  $\mathcal{O}(N_k N^{3/2})$  Toffoli and qubit complexity. Contrasting this to the supercell calculation, we see a  $\sqrt{N_k}$  savings due to the fact that the inner state preparation requires only  $N_k^2$  information and not  $N_k^3$  information. We elaborate on this point further in Sec. III B. SELECT is implemented in a similar fashion to sparse, scaling as  $\mathcal{O}(N_k N)$ , and is not a dominant cost.

The double factorization LCU represents the Hamiltonian in a series of nonorthogonal bases and leverages that a linear combination of ladder operators can be constructed by a similarity transform of a single fermionic ladder operator, or Majorana operator, by a unitary generated by a quadratic fermionic Hamiltonian. In the molecular case, the dominant cost for these algorithms is the QROM to output the rotations for the similarity transform and implementing the basis rotations with the programmable gate-array circuit primitive [33,45] for SELECT. When taking advantage of primitive cell symmetry we reduce the amount of data needed to be output by QROM by  $N_k$ , which results in a  $\sqrt{N_k}$  savings in the Toffoli complexity. Because we are using advanced QROM, this output size advantage is also observed in the logical qubit requirements. As mentioned previously, computing the total number of Toffolis requires scaling the walk operator cost by a linear function of  $\lambda$ . We find that using a canonical orbital basis set, the total Toffoli cost is higher than the commensurate supercell total Toffoli cost because  $\lambda$  for the symmetry-adapted case increases. The origin of the increase is related to a reduced variational freedom when selecting nonorthogonal bases and is further discussed in Sec. III C.

Finally, in the THC LCU there is no asymptotic speedup because the molecular algorithm had the lowest possible scaling for second quantized algorithms. This stems from the fact that even iterating over the basis once with unary iteration to apply an operator indexed by basis element has a Toffoli cost of  $\mathcal{O}(N_k N)$ . As we will discuss in Sec. III D our symmetry-adapted algorithm offers other benefits such as enabling the classical precomputation of the THC factors by exploiting symmetry and lowering the number of controlled rotations.

We now describe the Hamiltonian factorization used in each LCU, the calculation of  $\lambda$  associated with each Hamiltonian factorization, and outline the construction of the qubitization oracles. Detailed compilations are provided for each LCU in the Appendices. In each section we provide numerical evidence that the symmetry-adapted oracles have the reported scaling by plotting the Toffoli requirements to synthesize SELECT + PREPARE + PREPARE<sup>-1</sup> compared against the number of  $k$  points sampled ( $N_k$ ).

### A. The sparse Hamiltonian representation

In the ‘‘sparse’’ method, the Hamiltonian described in Eq. (8) and Eq. (9) is directly translated to Pauli operators, which form the LCU. Under the Jordan-Wigner transformation, we take

$$a_{p\mathbf{k}\sigma} \mapsto \vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})/2, \quad (18)$$

$$a_{p\mathbf{k}\sigma}^\dagger \mapsto \vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})/2, \quad (19)$$

where the notation  $\vec{Z}$  is being used to indicate that there is a string of  $Z$  operators on qubits up to (not including) that on which  $X_{p\mathbf{k}\sigma}$  or  $Y_{p\mathbf{k}\sigma}$  acts upon. This requires a choice of ordering for the qubits indexed by  $p$ ,  $\mathbf{k}$ , and  $\sigma$ . We need only apply the string of  $Z$  operators for the same value of  $\sigma$ , because we always have matching annihilation and creation operators for the same spin  $\sigma$  (so any  $Z$  gates on the other spin would cancel). We also adopt a convention that the ordering of qubits for the Jordan-Wigner transformation takes  $\mathbf{k}$  as the more significant bits, with qubits for all  $p$  with a given  $\mathbf{k}$  grouped together. For most of the discussion, we will not need to explicitly consider this ordering.

With the Jordan-Wigner transform the one-body component of the Hamiltonian takes on the form

$$\begin{aligned} H_1 = & \frac{i}{4} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p, q=1} \text{Re}(h_{p\mathbf{k}, q\mathbf{k}}) \\ & \times \{ \vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma} \} \\ & + \frac{i}{4} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p, q=1} \text{Im}(h_{p\mathbf{k}, q\mathbf{k}}) \\ & \times \{ \vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma} \} + \sum_{\mathbf{k}} \sum_{p=1} h_{p\mathbf{k}, p\mathbf{k}} \mathbb{1}. \end{aligned} \quad (20)$$

We provide the full derivation for this expression in Appendix A 1. To derive the two-body operator LCU we use only complex conjugation symmetry in contrast to the molecular derivation that used eightfold symmetry. The two-body Hamiltonian can be written as

$$\begin{aligned} H_2 = & \frac{1}{4} \sum_{\sigma, \tau \in \{\uparrow, \downarrow\}} \sum_{\mathbf{Q}, \mathbf{k}, \mathbf{k}'} \sum_{p, q, r, s=1}^{N_k} \sum_{N/2}^{N/2} \left[ V_{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'} a_{p\mathbf{k}\sigma}^\dagger \right. \\ & a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \ominus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau} + V_{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'}^* \\ & \left. a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} a_{s\mathbf{k}'\tau}^\dagger a_{r(\mathbf{k}' \ominus \mathbf{Q})\tau} \right], \end{aligned} \quad (21)$$

where  $\ominus$  indicates modular subtraction as defined above. In the case where  $\mathbf{Q} \neq 0$  or  $p \neq q$  and  $r \neq s$ , we can move the creation and annihilation operators using the fermionic anticommutation relations to give the term on the second line as

$$V_{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \ominus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger. \quad (22)$$

The Jordan-Wigner representation then gives the expression in square brackets in Eq. (21) as

$$\begin{aligned} & \frac{1}{16} \left\{ V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'} [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})] [\vec{Z}(X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} + iY_{q(\mathbf{k}\oplus\mathbf{Q})\sigma})] [\vec{Z}(X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} - iY_{r(\mathbf{k}'\oplus\mathbf{Q})\tau})] \right. \\ & \quad [\vec{Z}(X_{s\mathbf{k}'\tau} + iY_{s\mathbf{k}'\tau})] + V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}^* [\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})] \\ & \quad \left. [\vec{Z}(X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} - iY_{q(\mathbf{k}\oplus\mathbf{Q})\sigma})] [\vec{Z}(X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} + iY_{r(\mathbf{k}'\oplus\mathbf{Q})\tau})] [\vec{Z}(X_{s\mathbf{k}'\tau} - iY_{s\mathbf{k}'\tau})] \right\}. \end{aligned} \quad (23)$$

Then we can separate Eq. (23) into real and imaginary components as

$$\begin{aligned} & \frac{1}{8} \left\{ \text{Re}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}') [\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} + \vec{Z}Y_{p,\mathbf{k},\sigma} \vec{Z}Y_{q,\mathbf{k}\oplus\mathbf{Q},\sigma}] [\vec{Z}X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}X_{s\mathbf{k}'\tau} + \vec{Z}Y_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}Y_{s\mathbf{k}'\tau}] \right. \\ & \quad - \text{Re}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}') [\vec{Z}Y_{p,\mathbf{k},\sigma} \vec{Z}X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} - \vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k}\oplus\mathbf{Q})\sigma}] [\vec{Z}Y_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}X_{s\mathbf{k}'\tau} - \vec{Z}X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}Y_{s\mathbf{k}'\tau}] \\ & \quad + \text{Im}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}') [\vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} - \vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k}\oplus\mathbf{Q})\sigma}] [\vec{Z}X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}X_{s\mathbf{k}'\tau} + \vec{Z}Y_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}Y_{s\mathbf{k}'\tau}] \\ & \quad \left. + \text{Im}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}') [\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k}\oplus\mathbf{Q})\sigma}] [\vec{Z}Y_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}X_{s\mathbf{k}'\tau} - \vec{Z}X_{r(\mathbf{k}'\oplus\mathbf{Q})\tau} \vec{Z}Y_{s\mathbf{k}'\tau}] \right\}. \end{aligned} \quad (24)$$

In accounting for cases where  $\mathbf{Q} = 0$  with  $p = q$  or  $r = s$ , the same expression is obtained, but there are also one-body terms obtained. These result in a total one-body operator

$$\tilde{H}_1 = H_1 + \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p,q=1}^{N_k} \sum_{r=1}^{N_k} \sum_{\mathbf{k}'} V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',r\mathbf{k}'} a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \quad (25)$$

A full derivation of this expression can be found in Appendix A 2.

Using the representation of the one-body and two-body operators as Pauli operators, we have a linear combination of unitaries form. The  $\lambda$  associated with this LCU is

$$\lambda = \lambda_{\tilde{H}_1} + \lambda_{H_2} \quad (26)$$

$$\lambda_{\tilde{H}_1} = \sum_{\mathbf{k}} \sum_{pq} \left\{ \left| \text{Re}[h_{p\mathbf{k},q\mathbf{k}}] + \text{Re} \left[ \sum_{\mathbf{k}',r} V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',r\mathbf{k}'} \right] \right| + \left| \text{Im}[h_{p\mathbf{k},q\mathbf{k}}] + \text{Im} \left[ \sum_{\mathbf{k}',r} V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',r\mathbf{k}'} \right] \right| \right\} \quad (27)$$

$$\lambda_{H_2} = \sum_{\mathbf{k},\mathbf{k}',\mathbf{Q}} \sum_{pqrs} \left\{ \left| \text{Re}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}) \right| + \left| \text{Im}(V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}) \right| \right\}. \quad (28)$$

In determining  $\lambda_{\tilde{H}_1}$  there is a factor of 2 due to the summation over spin  $\sigma$  and then a factor of 2 accounting for the fact that each expression in braces in Eq. (20) is the sum of two different Pauli strings. As a result these factors have canceled the original 1/4 prefactor. In the expression for  $\lambda_{\tilde{H}_1}$  we have also summed over the native one-body terms and the contributions from the two-body terms. For  $\lambda_{H_2}$  we had a factor of 1/8 in Eq. (24), which is multiplied by the factor of 1/4 in Eq. (21). The two sums over spin  $\sigma$  and  $\tau$  give a factor of 4. Then for each of the real and imaginary parts in Eq. (24) there were sums over eight Pauli strings, giving a factor of 8. As a result these factors have also canceled in the expression for  $\lambda_{H_2}$ . Note that there is a factor of 2 between this expression and that in Ref. [33], even when we just consider  $V$  that is real. The reason is that

in Ref. [33] there was eightfold symmetry, where here we have only fourfold symmetry. That is, here we have symmetry when simultaneously swapping the pairs  $p, q$  and  $r, s$ , whereas in Ref. [33] there are two symmetries from swapping  $p, q$  or  $r, s$  on their own. That meant it was possible to express the Hamiltonian as in Eq. (A2) of that work, then in Eq. (A3) of that work the Jordan-Wigner mapping was used in the form

$$a_{p\sigma}^\dagger a_{q\sigma} + a_{q\sigma}^\dagger a_{p\sigma} \mapsto \frac{X_{p\sigma} \vec{Z}X_{q\sigma} + Y_{p\sigma} \vec{Z}Y_{q\sigma}}{2}. \quad (29)$$

In this mapping there has been a cancelation of half the Pauli strings, which results in  $\lambda$  being reduced by a factor of 2. Here we have only fourfold symmetry, so the value



of  $\lambda$  for the two-body term is a factor of 2 larger than that in Ref. [33].

In order to implement the Hamiltonian as a linear combination of unitaries, the first step is to perform a state preparation on  $\mathbf{Q}, \mathbf{k}, \mathbf{k}', p, q, r, s$ . This state preparation corresponds to the sum, then we will perform controlled operations for each of the operators in Eq. (24). The state preparation is applied using coherent alias sampling as described in Ref. [79]. Because there are multiple variables that the state needs to be prepared over, it is convenient to use the QROM to output “ind” values as well as “alt” and “keep” values. Both “ind” and “alt” give values of all variables  $\mathbf{Q}, \mathbf{k}, \mathbf{k}', p, q, r, s$ . Then an inequality test is performed between keep and an equal superposition state, and the result is used to control a swap between ind and alt.

There are both real and imaginary values of  $V_{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'}$ , so we also include a qubit to distinguish between these values in the state preparation. We also do not prepare all values of  $\{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'\}$ . There is symmetry in swapping  $p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q})$  with  $r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'$ , or simultaneously  $p\mathbf{k}$  with  $q(\mathbf{k} \ominus \mathbf{Q})$  and  $r(\mathbf{k}' \ominus \mathbf{Q})$  with  $s\mathbf{k}'$ . Only those values of  $\{p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q}), r(\mathbf{k}' \ominus \mathbf{Q}), s\mathbf{k}'\}$  that give unique values of  $V$  will be prepared. Then the full range can be obtained by using qubits to control these swaps. There is also a complex conjugate needed in the symmetry, which can be applied with a Clifford gate.

The dominant complexity in the preparation comes from the QROM. The number of items of data is  $\mathcal{O}(N_k^3 N^4)$ , and by using the advanced form of QROM the complexity can be made approximately the square root of the total amount of data (number of items of data times the size of each). The size of each item of data is logarithmic in  $N_k$  and  $N$  as well as the allowable error. Therefore, the scaling of the complexity can be given ignoring these logarithmic parts as  $\tilde{\mathcal{O}}(N_k^{3/2} N^2)$ .

To describe the controlled operations needed in order to implement the operation as in Eq. (24), we need to account for the fact that there are two lines for each of the real and imaginary components of  $V$ . In addition, for each line in Eq. (24) there is a product of two factors, each of which is a sum of two terms. To describe the linear combination of unitaries we therefore introduce three more qubits.

- The first is used to distinguish between the two lines for each of the real and imaginary parts in Eq. (24).
- The second distinguishes between the two terms in the first set of square brackets.
- The third distinguishes between the two terms in the second set of square brackets.

When implementing the controlled operations, we perform four operations of the form of  $\bar{Z}X$  or  $\bar{Z}Y$ , with  $X$  or  $Y$  being applied on target qubits indexed by  $p\mathbf{k}, q(\mathbf{k} \ominus \mathbf{Q})$

and so forth. These Pauli strings are applied using the approach of Ref. [33], but in this case there is the additional complication that we need to select between  $X$  or  $Y$ . This selection can be performed simply by performing the controlled Pauli string twice, once for  $X$  and once for  $Y$ . The complexity is proportional to  $N_k N$ , which is trivial compared to the complexity of the state preparation. The choice of whether  $X$  or  $Y$  is performed depends on the value of the three qubits selecting between the terms, as well as the qubit selecting between the real and imaginary parts. The processing of these qubits to determine the appropriate choice of  $X$  or  $Y$  can be performed with a trivial number of gates.

The last part to consider is how the implementation of the one-body part of the Hamiltonian is integrated with the implementation of the two-body part. In the state preparation, amplitudes corresponding to the real and imaginary parts of  $h_{p\mathbf{k}, q\mathbf{k}}$  will be produced, as well as a qubit selecting between the one- and two-body parts. That qubit will be used to also select between the choice of  $X$  and  $Y$ . For the one-body part there is a product of only two of the Pauli strings, so the other two will not be applied at all for the one-body part. See Appendix A3 for a more detailed description of the implementation.

In Fig. 1 we plot the Toffoli complexity to implement  $\text{SELECT} + \text{PREPARE} + \text{PREPARE}^{-1}$  for simulating the aforementioned sample systems using a symmetry-adapted  $\text{SELECT}$  and  $\text{PREPARE}$  at different Monkhorst-Pack grids and different number of bands (cc-pVDZ and cc-pVTZ). We compare the symmetry-adapted calculations to supercell calculations using the same  $\text{SELECT}$  and  $\text{PREPARE}$ . The supercell calculations do not explicitly take into account the symmetry of the primitive cell in the full simulation cell. To demonstrate the symmetry-adapted Hartree-Fock orbitals do not appreciably change the overall scaling with respect to a supercell calculation we plot the total  $\lambda$  for the supercell calculation (which reruns Hartree-Fock on the supercell) and the symmetry-adapted version.

Although Fig. 1 indicates some computational advantage for the symmetry-adapted case the expected  $\sqrt{N_k}$  improvement over the supercell case is not easily observed. The cost of the sparse method largely depends on the number of nonzero elements of  $V$ , which is generically expected to go as  $\mathcal{O}(N_k^3 N^4)$  for the symmetry-adapted case and  $\mathcal{O}(N_k^4 N^4)$  for the supercell case. The scaling ultimately depends on the number of nonzero elements in each block of integrals (indexed by three-momentum indices), which we expect to be independent of supercell size  $N_k$ . For Diamond we plot this dependence in Fig. 2 and demonstrate that convergence is slow and there is a strong  $N_k$  dependence in the number of nonzero elements in each two-electron integral block. This dependence makes observing the improvement in Toffoli cost for symmetry-adapted oracles difficult in the low  $N_k$  regime.

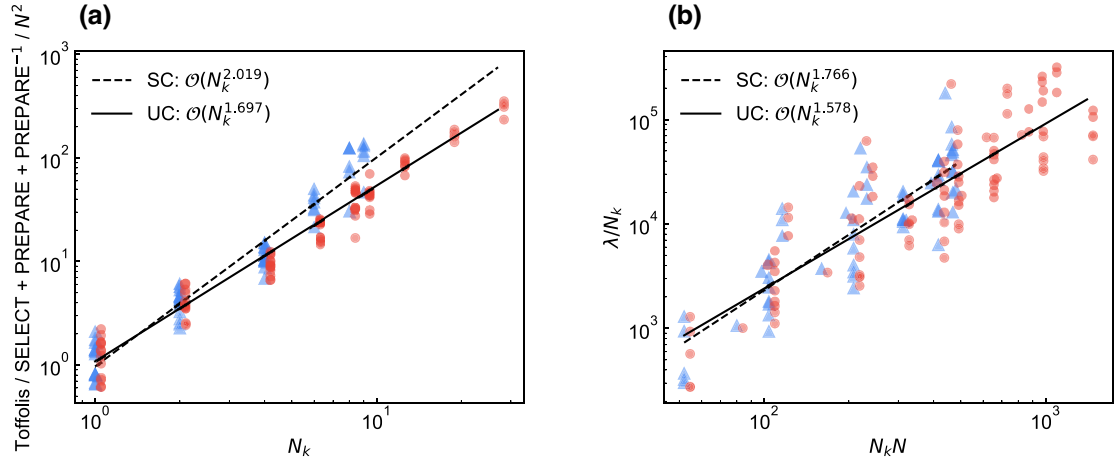


FIG. 1. (a) Sparse Toffoli step complexity versus the number of  $k$  points for systems in Table I using the cc-pVDZ and cc-pVTZ basis set  $\Gamma$ -centered Monkhorst-Pack grids of size  $[1, 1, 1]$  to  $[3, 3, 3]$ . Each point is a single system described at a particular basis set and  $k$  mesh where the threshold for zeroing each nonzero two-electron integral coefficient is determined by MP2 as described earlier. The scaling for implementing the block encoding is shown in the legend. To isolate the  $N_k$  scaling behavior we divide the Toffoli step complexity by the square of the number of basis functions ( $N^2$ ). For supercell we expect a scaling going as  $\mathcal{O}(N_k^2)$  and for symmetry-adapted block encodings we expect a scaling going as  $\mathcal{O}(N_k^{1.5})$ . The ideal symmetry-adapted scaling is not reached due to finite-size effects, which are further discussed in Fig. 2. (b) Total  $\lambda$  for the symmetry-adapted version (denoted UC) and the supercell calculation without explicit primitive cell symmetry (denoted SC) demonstrating no deterioration of  $\lambda$  by symmetry adapting. Similar to (a) all points are a particular system from the benchmark set in a fixed basis and  $k$  mesh.

## B. The single-factorization Hamiltonian representation

For the “single-factorization” method, the Cholesky decomposition of the two-electron integral tensor can be applied iteratively or the factorized forms can be directly recovered from a density fitted representation of the atomic

orbital integral. The quadratic representation of the two-electron integral tensor is

$$V_{p\mathbf{k}_p, q\mathbf{k}_q, r\mathbf{k}_r, s\mathbf{k}_s} = \sum_n L_{p\mathbf{k}_p, q\mathbf{k}_q, n} L_{s\mathbf{k}_s, r\mathbf{k}_r, n}^* \quad (30)$$

where  $\mathbf{k}_p + \mathbf{k}_r = \mathbf{k}_q + \mathbf{k}_s$  modulo a reciprocal lattice vector  $\mathbf{G}$ , or  $\mathbf{k}_p - \mathbf{k}_q - (\mathbf{k}_s - \mathbf{k}_r) = \mathbf{G}$ . We can identify  $\mathbf{k}_p - \mathbf{k}_q = \mathbf{Q} + \mathbf{G} = \mathbf{k}_s - \mathbf{k}_r$ . Thus the two-body interaction operator can be written as

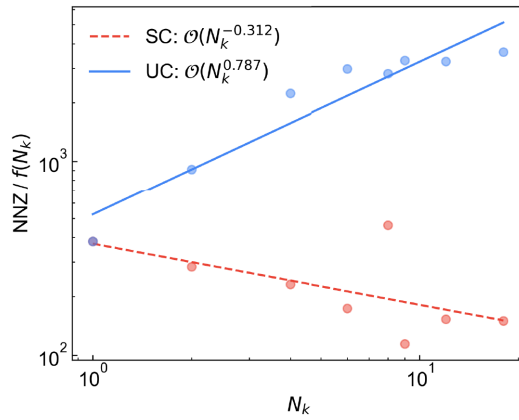


FIG. 2.  $N_k$  dependence on the number of nonzero elements in each two-electron integral block indexed by irrep labels for diamond in a single- $\zeta$ -valence basis with a  $k$  mesh shifted to  $(1/8, 1/8, 1/8)$  of the simulation cell. In the thermodynamic limit this value should be independent of  $N_k$  and thus both supercell (SC) and symmetry-adapted (UC) should have no correlation with  $N_k$ —i.e., a slope of zero. The  $N_k$  dependence for small  $N_k$  makes it difficult to observe the asymptotic improvements from symmetry adapting the sparse qubitization oracles.

$$\hat{H}'_2 = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k/2} L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma} \right) \times \left( \sum_{\tau \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}'} \sum_{rs}^{N_k/2} L_{s\mathbf{k}'r(\mathbf{k}' \ominus \mathbf{Q}), n}^* a_{r(\mathbf{k}' \ominus \mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \right). \quad (31)$$

Due to the reduced symmetry of the complex valued two-electron integral tensor we take additional steps to form Hermitian operators, which can be expressed as Pauli operators under the Jordan-Wigner transform. We express each one-body operator in the product of particle-conserving

one-body operators forming the two-electron operator as

$$\hat{\rho}_n(\mathbf{Q}) = \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} \right),$$

$$\hat{\rho}_n^\dagger(\mathbf{Q}) = \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}^* a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} \right). \quad (32)$$

We now take a linear combination of  $\hat{\rho}_n(\mathbf{Q})$  to form Hermitian operators and represent our two-electron integral operator as a sum of squares of Hermitian operators that are amenable to the approach for the qubitization of one-body sparse operators via a linear combination of unitaries. These operators are denoted  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  and are

defined as

$$\hat{A}_n(\mathbf{Q}) = \frac{1}{2}(\hat{\rho}_n(\mathbf{Q}) + \hat{\rho}_n^\dagger(\mathbf{Q})), \quad (33)$$

$$\hat{B}_n(\mathbf{Q}) = \frac{i}{2}(\hat{\rho}_n(\mathbf{Q}) - \hat{\rho}_n^\dagger(\mathbf{Q})), \quad (34)$$

to give

$$\hat{H}'_2 = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n^M \left( \hat{A}_n^2(\mathbf{Q}) + \hat{B}_n^2(\mathbf{Q}) \right). \quad (35)$$

We have taken advantage of the translational symmetry by performing the sum over  $\mathbf{Q}$  outside the squares of  $\hat{A}$  and  $\hat{B}$ , which reduces the amount of information needed in the representation. In the case  $\mathbf{Q} \neq 0$  we can write  $\hat{A}_n$  as

$$\begin{aligned} \hat{A}_n(\mathbf{Q} \neq 0) &= \frac{1}{2} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} \left( L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}^* a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} \right) \\ &= \frac{1}{2} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} \text{Re}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}] \left( a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} \right) \\ &\quad + \frac{i}{2} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} \text{Im}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}] \left( a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} - a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} \right). \end{aligned} \quad (36)$$

Applying the Jordan-Wigner representation then gives

$$\begin{aligned} \hat{A}_n(\mathbf{Q} \neq 0) &= \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} \left( \frac{i \text{Re}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right. \\ &\quad \left. + \frac{i \text{Im}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right). \end{aligned} \quad (37)$$

The same reasoning can be performed for  $\hat{B}_n(\mathbf{Q} \neq 0)$ , which gives the plus and minus signs between  $a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}$  and  $a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma}$  in Eq. (36) reversed, so the roles of the real and imaginary parts are reversed. As a result we obtain

$$\begin{aligned} \hat{B}_n(\mathbf{Q} \neq 0) &= \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq}^{N_k} \left( \frac{i \text{Im}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right. \\ &\quad \left. + \frac{i \text{Re}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right). \end{aligned} \quad (38)$$

Accounting for the cases with  $\mathbf{Q} = 0$ , we may use the same expressions with an extra identity, which yields a one-body correction when squaring. We show in Appendix B 1 that this results in the total one-body operator

$$\tilde{H}_1 = \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p, q=1}^{N_k} \left( h_{p\mathbf{k}, q\mathbf{k}} + \sum_{r=1}^{N/2} \sum_{\mathbf{k}'} V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'} \right) a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma} \quad (39)$$

as before. Therefore, the associated  $\lambda$  is again  $\lambda_{\hat{H}_1}$  as given in Eq. (26). The  $\lambda$  for the two-body term is then

$$\lambda_V = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n^M \left( \sum_{\mathbf{k}, p, q} (|\operatorname{Re}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]| + |\operatorname{Im}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]|) \right)^2. \quad (40)$$

This expression can be obtained by first summing the absolute values of the weights in the linear combination of unitaries for  $\hat{A}$  and  $\hat{B}$  to give

$$\sum_{\mathbf{k}, p, q} (|\operatorname{Re}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]| + |\operatorname{Im}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]|). \quad (41)$$

This is obtained by noting that the sum over the spin gives a factor of 2, and there are two unitary operators for each of the real and imaginary parts; together these cancel the factor of 4. Then this expression is squared for each of  $\hat{A}$  and  $\hat{B}$ , and there is a sum over  $\mathbf{Q}$  and  $n$  in Eq. (35). A further factor of 1/2 is obtained because we use amplitude amplification on each operator as described in Ref. [33], thus giving our expression for  $\lambda_V$ .

Next we describe the method to block encode the Hamiltonian in this single-factorized representation. The key idea is to perform a state preparation over  $\mathbf{Q}$  and  $n$ , then block encode the squares of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  using a single step of oblivious amplitude amplification (which saves a factor of 2 for the value of  $\lambda$ ). That is, we perform block encodings of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ , reflect on an ancilla register, then apply the block encodings again.

For the initial state preparation on  $\mathbf{Q}, n$ , the number of items of data is  $MN_k + 1$ , where the +1 is for the one-body part of the Hamiltonian. This state preparation is via coherent alias sampling, so the dominant cost is from the QROM needed to output ind and alt values. That has complexity scaling as  $\tilde{O}(\sqrt{MN_k})$ , where the tilde accounts for the size of the items of data.

For both  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  we have weightings according to the real and imaginary parts of  $L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}$ , but the difference is in what operations are performed in the sum. Therefore, for an LCU block encoding, the state-preparation step may be identical between  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ . For each value of  $\mathbf{Q}, n$ , the number of unique values of  $\mathbf{k}, p, q$  to consider is  $N_k N^2 / 4$ . Unlike the supercell case we cannot take advantage of symmetry between  $p$  and  $q$ , because we have  $p\mathbf{k}$  and  $q(\mathbf{k} \ominus \mathbf{Q})$ . The relation between  $\mathbf{k}$  and  $\mathbf{k} \ominus \mathbf{Q}$  is governed by the value of  $\mathbf{Q}$ , which is given in the outer sum, and so we cannot exchange  $p$  and  $q$ . There is a further factor of 2 for the number of items of data, because both real and imaginary parts are needed.

Accounting for the values of  $\mathbf{Q}, n$ , the total number of items of data that must be output by the QROM used

in the state preparation is  $(MN_k + 1)N_k N^2 / 2 = \mathcal{O}(N_k^2 N^3)$ , given that  $M$  scales as  $\mathcal{O}(N)$ . Again because the size of the items of data is logarithmic, this gives a complexity  $\tilde{O}(N_k N^{3/2})$ . In contrast, in the supercell calculation, each  $\hat{A}_n$  and  $\hat{B}_n$  would have  $\mathcal{O}(N_k^2 N^2)$  entries, and the rank would be  $\mathcal{O}(N_k N)$ , for a total number of items of data  $\mathcal{O}(N_k^3 N^3)$ . That would give a complexity  $\tilde{O}(N_k^{3/2} N^{3/2})$ , so there is a factor of  $\sqrt{N_k}$  improvement obtained by taking advantage of the symmetry.

In the state preparation we prepare only  $p, q$  for  $p \leq q$ , and the full range of values should be produced using a swap controlled by an ancilla register. A further subtlety in the implementation as compared to prior work is that the complex conjugate is needed as well. This may be implemented using a sign flip on the qubit indicating the imaginary part, so it is just a Clifford gate.

A major difference is in the selection of operations for  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ . We see that there are two steps where we need to apply an operation of the form  $\vec{Z}X$  or  $\vec{Z}Y$ , and the choice of  $X$  or  $Y$ . The selection of where the  $X$  or  $Y$  is applied (indicated by the subscript) can be implemented in the standard way. The choice of whether  $X$  or  $Y$  is applied depends on four qubits.

1. The qubit selecting between the one- and two-body parts.
2. A qubit selecting between  $A$  and  $B$ , which can simply be prepared in an equal superposition using a Hadamard because there are equal weightings between these operators.
3. A qubit selecting between the real and imaginary parts of  $L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}$ , which was prepared in the state preparation.
4. A qubit selecting between the two terms shown above in each line of the expressions for  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ . This qubit can also be prepared using a Hadamard.

Using a trivial number of operations on these qubits we can determine whether it is  $X$  or  $Y$  that needs to be performed. The cost of the controlled unitary is doubled because we apply a controlled  $\vec{Z}X$  and a controlled  $\vec{Z}Y$ , but this cost is trivial compared to the state-preparation cost so has little effect on the overall complexity.

A further subtlety in the implementation is that in the second implementation of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ , we simply use the qubit flagging the one-body part to control whether the Pauli string  $\vec{Z}X$  or  $\vec{Z}Y$  is applied at all. This ensures that the square is not obtained for the one-body part. For a more in-depth explanation of the implementation, see the circuit diagram in Fig. 3 and the explanation in Appendix B 2.

Figure 4 demonstrates the  $\sqrt{N_k}$  improvement in constructing the walk operator by symmetry adapting. Even



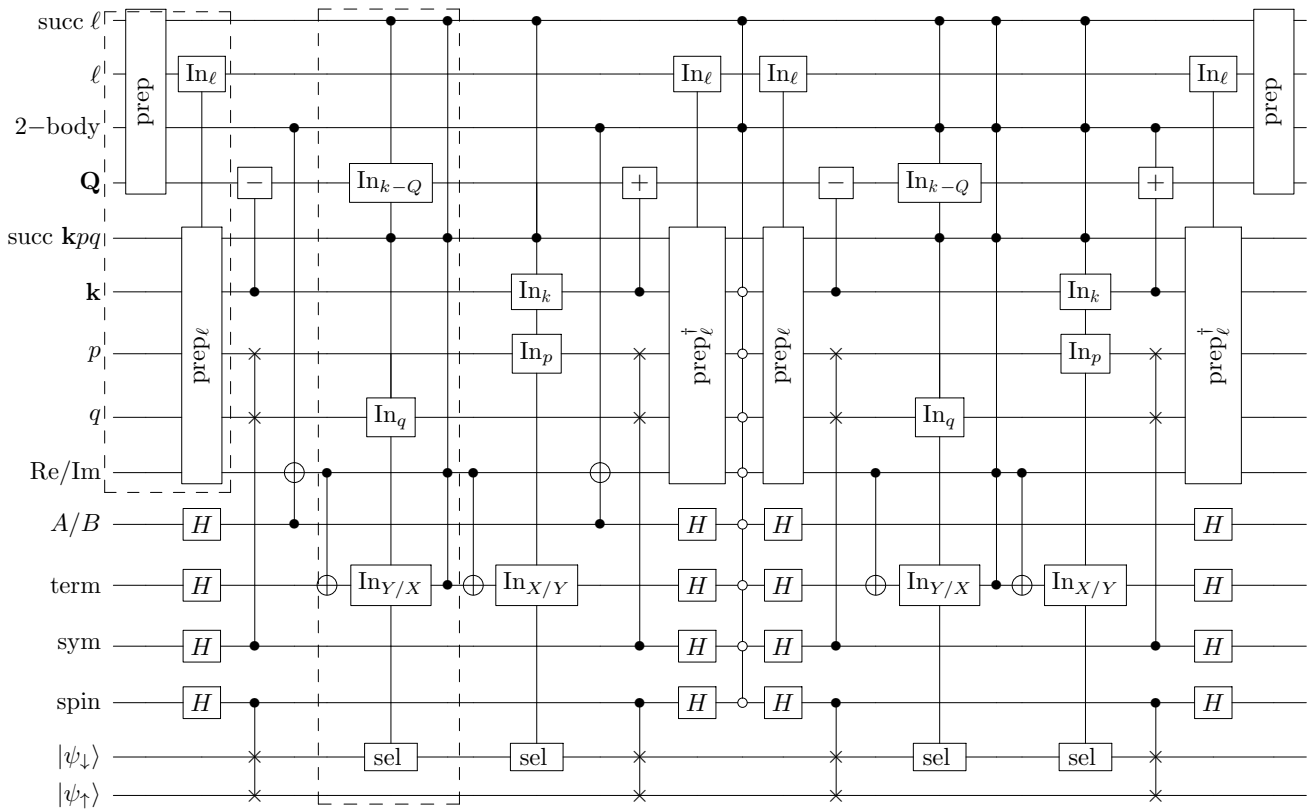


FIG. 3. The circuit for performing the state preparation and controlled operations for the single factorization approach. The register labeled  $\ell$  is a contiguous register for  $\mathbf{Q}, n$ , with  $\mathbf{Q}$  also output in the state preparation. The inner state preparation uses  $\ell$  as a control. Then the minus on  $\mathbf{Q}$  controlled by  $\mathbf{k}$  is to compute  $\mathbf{k} \ominus \mathbf{Q}$ , with that register reset to  $\mathbf{Q}$  later with a controlled addition. A qubit is used to swap  $p$  and  $q$  to generate that symmetry, and another is used to swap the spin-up and spin-down components of the system so the selection need act only on the spin-down component. The qubits labeled Re/Im,  $A/B$ , and “term” are the qubits selecting the real versus imaginary parts,  $A$  versus  $B$ , and the two terms in each line of the Hamiltonian. These correspond to  $b_1, b_2, b_3$  in Appendix B 2, and the Toffoli and CNOT gates are used so that the “term” qubit can be used to select whether the Pauli string with  $X$  or  $Y$  is applied. The selection is performed twice, once for each of the Pauli strings and so is controlled by  $\mathbf{k} \ominus \mathbf{Q}$  and  $q$  the first time, then  $\mathbf{k}$  and  $p$  the second time. A controlled phase between Re/Im and “term” (also controlled by the success flag qubits) is used to generate the correct sign for the term. The block encoding of  $A/B$  is performed twice, with the reflection on the ancilla qubits in the middle generating the step of oblivious amplitude amplification. The third register flags that we have the two-body part of the Hamiltonian, and is used to control the block encoding of  $A/B$  the second time to ensure it is not performed for the one-body part. The dashed boxes enclose the two most expensive parts of the circuit, which implement the bulk of the PREPARE and SELECT steps. The first box (reading left to right), which implements prepare has two steps, an outer prepare (labeled prep) and an inner prepare ( $\text{prep}_\ell$ ). The inner prepare’s cost is dominant and scales like  $\tilde{O}(\sqrt{N_k^2 N^3})$  resulting from a QROAM load over the  $(\ell, \mathbf{k}, p, q)$  registers. The outer prepare has complexity of  $\tilde{O}(\sqrt{N_k N})$  from the  $\ell$  register. The complexity of the selected Majorana operator in the second box is  $\mathcal{O}(N_k N)$  and is dominated by the unary iteration.

for small  $N_k$  there is a clear separation between the cost of SC and symmetry-adapted oracles that agrees with the theoretical scalings of 1.5 and 1.0, respectively.

### C. The double-factorization Hamiltonian representation

In the sparse and SF LCU approaches we have found that there is a factor of  $\sqrt{N_k}$  savings in Toffoli costs and logical qubit costs for symmetry-adapted block-encoding constructions over their nonsymmetry-adapted

counterparts (supercell calculations). The double-factorization (DF) representation continues this trend, though the origin of the speedup is different. In the double-factorization circuits each unitary of the LCU is a rank-one one-body operator that can be thought of as the outer product of two vectors of ladder operators, where each vector of ladder operators is obtained by a Givens rotation with multiqubit control based on other indices. First notice that for SF there is  $\mathcal{O}(N_k^2 N^3)$  data to output to specify the Hamiltonian via the Cholesky factors. The factors come from two momentum indices  $\mathbf{k}, \mathbf{Q}$ , two band indices  $p, q$ , and one

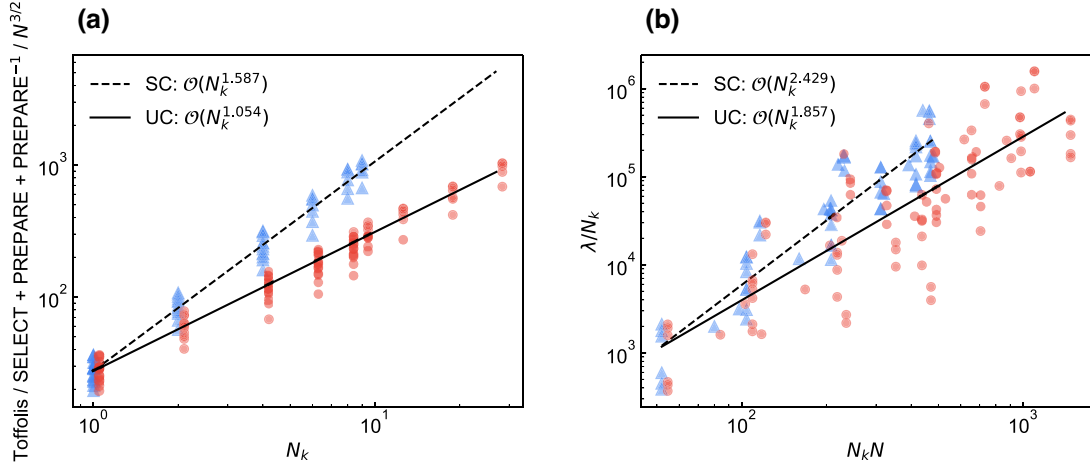


FIG. 4. (a) Number of  $k$  points versus Toffoli cost to implement the block encoding for the single factorization LCU evaluated for the benchmark systems listed in Table I described using the cc-pVDZ and cc-pVTZ basis sets and  $\Gamma$ -centered Monkhorst-Pack grids of size  $[1, 1, 1]$  to  $[3, 3, 3]$ . Each point is a single system described at a particular basis set and  $k$  mesh where the range of the auxiliary index of the Cholesky factorization is selected to produce two-electron integrals corresponding to an MP2 error of one 1 millihartree per unit cell with respect to an untruncated auxiliary index range. We divide the Toffoli complexity for implementing SELECT + PREPARE + PREPARE<sup>-1</sup> by  $N^{3/2}$ , which is the shared scaling in the number of bands. The different scaling in number of  $k$  points becomes clear:  $N_k$  for symmetry-adapted block encodings and  $N_k^{3/2}$  for supercell nonsymmetry-adapted block encodings. We observe similar behavior for qubit count and for plotting oracle Toffoli complexity versus the number of bands. (b) The value of  $\lambda$  per unit cell ( $\lambda/N_k$ ) as a function of the total system size  $NN_k$  for the same systems described with the same cutoffs used in (a).

auxiliary index  $n$ . In this section we demonstrate that by using a workspace register to apply Givens rotations to pairs of band indices,  $\{\mathbf{k}, \mathbf{k} \ominus \mathbf{Q}\}$ , the complexity of the DF LCU can also be improved by a factor of  $\sqrt{N_k}$  over supercell calculations.

To construct the DF LCU, we will separate  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  out into sums over  $\mathbf{k}$ . To express this, instead of having  $\rho_n(\mathbf{Q})$ , we define  $\rho_n(\mathbf{Q}, \mathbf{k})$

$$\begin{aligned} \hat{\rho}_n(\mathbf{Q}, \mathbf{k}) &= \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{pq}^{N/2} L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma} \right), \\ \hat{\rho}_n^\dagger(\mathbf{Q}, \mathbf{k}) &= \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{pq}^{N/2} L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}^* a_{q(\mathbf{k} \ominus \mathbf{Q})\sigma}^\dagger a_{p\mathbf{k}\sigma} \right), \end{aligned} \quad (42)$$

so then the Hermitian one-body operators that are squared to form the two body part of the Hamiltonian are

$$\hat{A}_n(\mathbf{Q}) = \sum_{\mathbf{k}} \frac{1}{2} (\hat{\rho}_n(\mathbf{Q}, \mathbf{k}) + \hat{\rho}_n^\dagger(\mathbf{Q}, \mathbf{k})), \quad (43)$$

$$\hat{B}_n(\mathbf{Q}) = \sum_{\mathbf{k}} \frac{i}{2} (\hat{\rho}_n(\mathbf{Q}, \mathbf{k}) - \hat{\rho}_n^\dagger(\mathbf{Q}, \mathbf{k})). \quad (44)$$

Just as in the single factorization case, we have the two-body part of the Hamiltonian

$$\hat{H}'_2 = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n^{N_k} (\hat{A}_n^2(\mathbf{Q}) + \hat{B}_n^2(\mathbf{Q})). \quad (45)$$

We can write  $\hat{A}_n(\mathbf{Q})$  as

$$\hat{A}_n(\mathbf{Q}) = \sum_{\mathbf{k}} \left[ U_n^A(\mathbf{Q}, \mathbf{k}) \left( \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q}, n, \mathbf{k}, A}} f_p^A(\mathbf{Q}, n, \mathbf{k}) n_{p\mathbf{k}\sigma} \right) U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \right], \quad (46)$$

where the basis rotation unitary  $U_n(\mathbf{Q}, \mathbf{k})$  acts on orbitals indexed by  $\mathbf{k}$  and  $\mathbf{k} \ominus \mathbf{Q}$ ,  $\Xi_{\mathbf{Q}, n, \mathbf{k}, A}$  corresponds to a rank cutoff for  $A$ , and  $f_p^A(\mathbf{Q}, n, \mathbf{k})$  is the eigenvalue of the one body

operator that is diagonalized by  $U_n(\mathbf{Q}, \mathbf{k})$ . The expression for  $\hat{B}_n(\mathbf{Q})$  is similar, and we use  $\Xi_{\mathbf{Q}, n, \mathbf{k}, B}$  to denote the rank cutoff.

In practice, for the implementation we would apply a different basis rotation for each individual value of  $p$ . As explained by Ref. [33], when doing that the number of Givens rotations needed only corresponds to the number of orbitals it is acting upon, instead of the square. Here we have two momentum modes  $\{\mathbf{k}, \mathbf{k} \ominus \mathbf{Q}\}$  with  $N$  orbitals for each, suggesting there should be  $2N$ . However, there is no mixture between the different spin states indexed by  $\sigma$ , so that gives the number of orbitals as  $N$ .

To quantify the amount of information needed to specify the rotations for the Hamiltonian, there is a  $\mathbf{Q}$  summation,  $n$  summation,  $\mathbf{k}$  summation,  $p$  summation, and we need to specify  $N$  Givens rotations for each. In turn, each Givens rotation needs two angles. The total data here therefore scales as

$$\tilde{\mathcal{O}} \left( N \sum_{\mathbf{Q}, n, \mathbf{k}} (\Xi_{\mathbf{Q}, n, k, A} + \Xi_{\mathbf{Q}, n, k, B}) \right), \quad (47)$$

where a factor of  $N$  comes from the number of Givens rotations, and the tilde accounts for the bits of precision given for the rotations. By analogy with the supercell case, it is convenient to define an average rank

$$\Xi := \frac{1}{2N_k M} \sum_{\mathbf{Q}, n, \mathbf{k}} (\Xi_{\mathbf{Q}, n, k, A} + \Xi_{\mathbf{Q}, n, k, B}). \quad (48)$$

This has division by  $N_k$  for the  $\mathbf{Q}$  sum and  $M$  for the  $n$  sum, but no division by a factor accounting for  $\mathbf{k}$ . That is, it is the average rank for each value of  $\mathbf{Q}$  and  $n$ , with the sum over  $\mathbf{k}$  regarded as part of the rank. Then it is most closely analogous to the rank in the supercell case, and it is found that it similarly scales as  $\mathcal{O}(N_k N)$ . In terms of  $\Xi$ , the scaling of the amount of data can be given as  $\tilde{\mathcal{O}}(N_k N^2 \Xi)$ , using  $M = \mathcal{O}(N)$ .

Next we describe in general terms how to perform the block encoding of the linear combination of unitaries, with the full explanation in Appendix C. As in the case of single factorization, the general principle is to perform state preparation over  $\mathbf{Q}$  and  $n$ , then block encode the squares of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  using oblivious amplitude amplification. The difference is that  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$  are now block encoded in a factorized form. In more detail, the key parts are as follows.

1. Perform a state preparation over  $\mathbf{Q}$  and  $n$ , as well as a qubit distinguishing between  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ . Using the advanced QROM, the complexity of this state preparation scales approximately as the square root of the number of items of data, so as  $\tilde{\mathcal{O}}(\sqrt{N_k N})$ . The tilde accounts for logarithmic factors from the size of the output. For convenience here we use a contiguous register for combined values of  $\mathbf{Q}$  and  $n$ .
2. Apply a QROM, which outputs the value of  $\mathbf{Q}$ , as well as an offset needed for the contiguous register needed in the state preparation for  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ .
3. Perform the inner state preparation over  $\mathbf{k}$  and  $p$ . Here the number of items of data is  $\mathcal{O}(N_k N \Xi)$ , accounting for the sums over  $\mathbf{Q}, n, \mathbf{k}, p$ . The complexity via advanced QROM is approximately the square root of this quantity,  $\tilde{\mathcal{O}}(\sqrt{N_k N \Xi})$ .
4. Apply the QROM again to output the rotation angles for the Givens rotations needed for the basis rotation. This time the size of the output scales as  $\mathcal{O}(N)$ , so the complexity of the QROM scales as  $\tilde{\mathcal{O}}(N \sqrt{N_k \Xi})$ . This is the dominating term in the complexity.
5. Use control qubits to swap the system registers into the correct location. This is done first controlled by a qubit labeling the spin,  $\sigma$ , which is similar to what was done in prior work. The new feature here is that registers containing  $\mathbf{k}$  and  $\mathbf{k} \ominus \mathbf{Q}$  are also used to swap system registers into  $N$  target qubits.
6. Apply the Givens rotations on these  $N$  target qubits. The complexity here scales only as  $\tilde{\mathcal{O}}(N)$ , so is smaller than in the other steps.
7. Apply a controlled  $Z$  for part of the number operator. This comes from representing the number operator as  $(\mathbb{1} - Z)/2$  and combining the identity with the one-body part of the Hamiltonian.
8. Invert the Givens rotations, controlled swaps, QROM for the Givens rotations, and state preparation over  $\mathbf{k}, p$ . The complexities here are similar to those in the previous steps, but the complexities for QROM erasure are reduced. This completes the block encoding of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ .
9. Perform a reflection on the ancilla qubits used for the state preparation on  $\mathbf{k}, p$ . This is needed for the oblivious amplitude amplification.
10. Perform steps 3 to 8 again for a second block encoding of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ . This together with the reflection gives a step of oblivious amplitude amplification, and therefore the squares of  $\hat{A}_n(\mathbf{Q})$  and  $\hat{B}_n(\mathbf{Q})$ .
11. Invert the QROM from step 2. This has reduced complexity because it is an erasure.
12. Invert the state preparation from step 1.

A quantum circuit for the procedure is shown in Fig. 5. This is similar to Fig. 16 in [33], except it is including the extra parts needed in order to account for the momentum  $\mathbf{k}$  used here. In particular,  $\ell$  shown here is a contiguous register for  $\mathbf{Q}, n$ , and  $p$  shown in the diagram is actually a contiguous register for  $\mathbf{k}, p$ . The values of  $\mathbf{Q}$  and  $\mathbf{k}$  need to be output via QROM after the state preparations. Then  $\mathbf{k} \ominus \mathbf{Q}$  is computed and  $\mathbf{k}$  and  $\mathbf{k} \ominus \mathbf{Q}$  are used to swap the required part of the system register into  $N$  target qubits where we apply the Givens rotations.

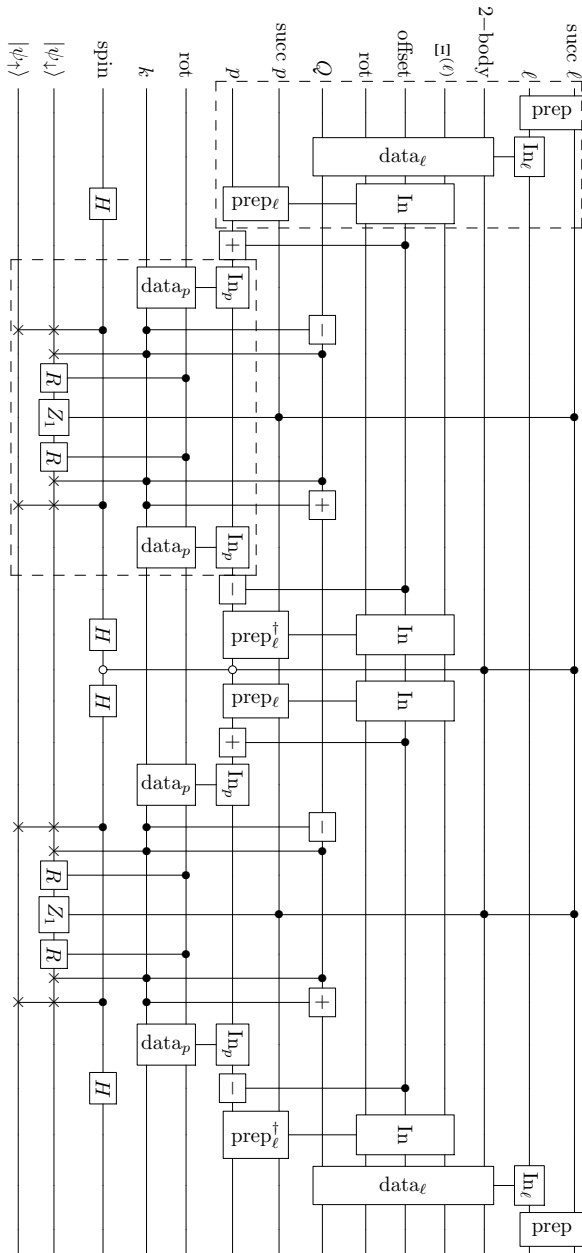


FIG. 5. The circuit for performing the state preparation and controlled operations for the double factorization approach. The register labeled  $\ell$  is a contiguous register for preparing  $\mathbf{Q}, n$ , with  $\mathbf{Q}$  output next in the QROM. The register labeled  $p$  is actually a contiguous register for preparing both  $\mathbf{k}$  and  $p$ . The value of  $\mathbf{k}$  is output in the next step together with the rotations. Then the minus on  $\mathbf{Q}$  controlled by  $\mathbf{k}$  is to compute  $\mathbf{k} \ominus \mathbf{Q}$ . The “ $\times$ ” on  $|\psi_{\downarrow}\rangle$  controlled by the  $\mathbf{Q}$  and  $\mathbf{k}$  registers indicates that these registers are used to swap the required part of the system register into  $N$  target qubits that the Givens rotations  $R$  act upon. The dashed boxes represent the two most expensive parts of the circuit, which implement the bulk of the PREPARE and SELECT steps. The prepare box has three steps, an outer prepare has a  $\tilde{O}(\sqrt{N_k N})$  Toffoli complexity while the inner prepare scales like  $\tilde{O}(\sqrt{N_k N})$ . The second dashed box loads the Givens rotation angles at a cost of  $\tilde{O}(N\sqrt{N_k \Xi})$  and performs the controlled rotations at cost  $\tilde{O}(N)$ .

The  $\lambda$  value for the Hamiltonian can be calculated by determining the total L1 norm of the coefficients of the unitaries used to represent the Hamiltonian. To determine this norm, note first that the number operator is replaced with  $(\mathbb{1} - Z)/2$ , and the identity is combined with the one-body part of the Hamiltonian. For  $\hat{A}_n(\mathbf{Q})$ , what is implemented therefore corresponds to

$$-\frac{1}{2} \sum_{\mathbf{k}} \left[ U_n^A(\mathbf{Q}, \mathbf{k}) \left( \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q}, n, \mathbf{k}, A}} f_p^A(\mathbf{Q}, n, \mathbf{k}) Z_{p\mathbf{k}\sigma} \right) U_n^A(\mathbf{Q}, \mathbf{k})^{\dagger} \right]. \quad (49)$$

Summing the absolute values of coefficients here gives

$$\sum_{\mathbf{k}} \sum_p^{\Xi_{\mathbf{Q}, n, \mathbf{k}, A}} |f_p^A(\mathbf{Q}, n, \mathbf{k})|, \quad (50)$$

where the sum over the spin  $\sigma$  has given a factor of 2, which canceled the factor of 1/2. In implementing the square of  $\hat{A}_n(\mathbf{Q})$  we use oblivious amplitude amplification, which provides a factor of 1/2 to  $\lambda$ . Combining this with the 1/2 in the definition of  $\hat{H}'_2$  gives 1/4, and combining with the contribution from  $\hat{B}_n(\mathbf{Q})$  then gives

$$\lambda_{\text{DF},2} = \frac{1}{4} \sum_{\mathbf{Q}, n} \left[ \left( \sum_{\mathbf{k}, p}^{N_k \Xi_{\mathbf{Q}, n, \mathbf{k}, A}} |f_n^A(p, \mathbf{Q}, \mathbf{k})| \right)^2 + \left( \sum_{\mathbf{k}, p}^{N_k \Xi_{\mathbf{Q}, n, \mathbf{k}, B}} |f_n^B(p, \mathbf{Q}, \mathbf{k})| \right)^2 \right], \quad (51)$$

where the superscript  $B$  on  $f$  indicates the corresponding quantity for  $\hat{B}_n(\mathbf{Q})$ .

The one-body Hamiltonian is adjusted by the one-body term arising from the identity in the representation of the number operator in the two-body Hamiltonian. This yields an effective one-body Hamiltonian (see Appendix C)

$$H'_1 = \sum_{\mathbf{k}, p, q, \sigma} \left( h_{p\mathbf{k}, q\mathbf{k}} + \sum_{\mathbf{k}', r} V_{r\mathbf{k}', r\mathbf{k}', q\mathbf{k}, p\mathbf{k}} \right) a_{p\mathbf{k}\sigma}^{\dagger} a_{q\mathbf{k}\sigma}. \quad (52)$$

We can rewrite this as

$$H'_1 = \sum_{\mathbf{k}, \sigma} \left[ U^C(\mathbf{k}) \left( \sum_p^{N/2} \lambda_{\mathbf{k}, p} n_{p\mathbf{k}\sigma} \right) U^C(\mathbf{k})^{\dagger} \right], \quad (53)$$

where  $\lambda_{\mathbf{k}, p}$  are eigenvalues of the matrix indexed by  $p, q$  in the brackets in Eq. (52). Thus the L1 norm of  $H'_1$  is the



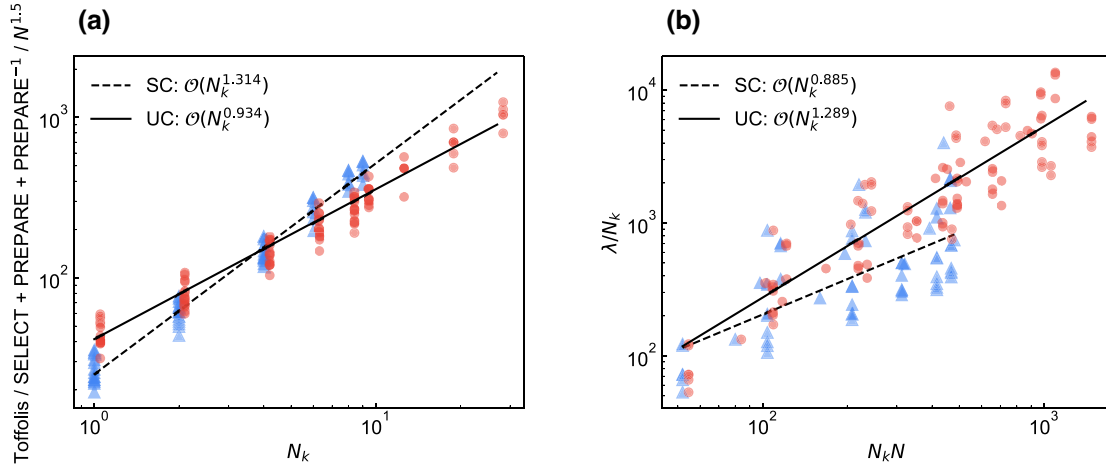


FIG. 6. (a) The number of  $k$  points versus Toffoli cost to implement the block encoding for the double factorization LCU evaluated for the benchmark systems listed in Table I described using the cc-pVDZ and cc-pVTZ basis sets and  $\Gamma$ -centered Monkhorst-Pack grids of size [1, 1, 1] to [3, 3, 3]. Each point is a single system described at a particular basis set and  $k$  mesh where the threshold to keep eigenvalues and vectors of the second factorization is selected to produce two-electron integrals corresponding to an MP2 error of one 1 millihartree with respect to an untruncated double factorization. On average this corresponds to a threshold value of  $1 \times 10^{-4}$  for the benchmark systems. The expected  $\mathcal{O}(\sqrt{N_k})$  scaling improvement for symmetry-adapted walk operators is demonstrated. (b) The value of  $\lambda$  per unit cell as a function of the total system size  $NN_k$  for the same systems described with the same cutoffs used in (a). The reduced variational freedom in compression of the two-electron integral tensors for the symmetry-adapted walk operator construction translates to an increased value of  $\lambda$  at all system sizes.

sum

$$\lambda_{\text{DF},1} = \sum_{\mathbf{k}} \sum_p |\lambda_{\mathbf{k},p}|. \quad (54)$$

Figure 6 demonstrates the improved  $\sqrt{N_k}$  scaling of the block encodings coming from reducing the number of controlled rotations by  $N_k$ . Unlike the SF case  $\lambda$  for DF has worse scaling in the symmetry-adapted setting compared to the supercell case. This is rationalized by the fact that there is a larger degree of variational freedom in the second factorization for supercell calculations (and thus more compression) compared to the symmetry-adapted case. The  $\lambda$  value is basis-set-dependent and can potentially be reduced by orbital optimization [87].

#### D. The tensor hypercontraction Hamiltonian representation

In the tensor hypercontraction (THC) LCU representation the fact that the two-electron integrals can be represented in a symmetric canonical polyadiclike decomposition is used to define a set of nonorthogonal basis functions in which to represent the Hamiltonian, and we use a similar infrastructure to the DF algorithm to implement each term in the factorization (which is in a different nonorthogonal basis) sequentially. In the following section, we describe the Bloch orbital version (symmetry adapted) of the THC decomposition and the resulting LCU,  $\lambda$  calculation, and qubitization complexities. First we review the salient features of tensor hypercontraction for the molecular case

before introducing symmetry labels. Recall that in the molecular THC approach we expand densitylike terms over a grid of  $M$  points (labeled  $\mu$ ) and weight each grid point with a function  $\xi_\mu(\mathbf{r})$

$$\phi_p(\mathbf{r})\phi_q(\mathbf{r}) \approx \sum_{\mu} \xi_{\mu}(\mathbf{r})\phi_p(\mathbf{r}_{\mu})\phi_q(\mathbf{r}_{\mu}), \quad (55)$$

which allows us to write the two-electron integral tensor as

$$V_{pqrs} = \sum_{\mu\nu} \chi_p^{(\mu)} \chi_q^{(\mu)} \zeta_{\mu\nu} \chi_r^{(\nu)} \chi_s^{(\nu)}, \quad (56)$$

where the central tensor is defined as

$$\zeta_{\mu\nu} = \int d\mathbf{r} \int d\mathbf{r}' \frac{\xi_{\mu}(\mathbf{r})\xi_{\nu}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}. \quad (57)$$

In order to incorporate translational symmetry into the THC factorization, the decomposition of the density is performed on the *cell periodic* part of the Bloch orbitals as [88,89]

$$u_{p\mathbf{k}_p}^*(\mathbf{r})u_{q\mathbf{k}_q}(\mathbf{r}) \approx \sum_{\mu} \xi_{\mu}(\mathbf{r})u_{p\mathbf{k}_p}^*(\mathbf{r}_{\mu})u_{q\mathbf{k}_q}(\mathbf{r}_{\mu}), \quad (58)$$

where  $u_{p\mathbf{k}_p}(\mathbf{r}) = e^{-i\mathbf{k}_p\mathbf{r}}\phi_{p\mathbf{k}_p}(\mathbf{r})$ . Then the two-electron integral tensor has the form

$$\begin{aligned} V_{p\mathbf{k}_p,q\mathbf{k}_q,r\mathbf{k}_r,s\mathbf{k}_s} &= \int d\mathbf{r} \int d\mathbf{r}' \phi_{p\mathbf{k}_p}^* \phi_{q\mathbf{k}_q} V(\mathbf{r}, \mathbf{r}') \phi_{r\mathbf{k}_r}^* \phi_{s\mathbf{k}_s} \\ &= \sum_{\mu\nu} u_{p\mathbf{k}_p}^*(\mathbf{r}_\mu) u_{q\mathbf{k}_q} \\ &\quad \times (\mathbf{r}_\mu) \zeta_{\mu\nu}^{\mathbf{k}_p, \mathbf{k}_q, \mathbf{k}_r, \mathbf{k}_s} u_{r\mathbf{k}_r}^*(\mathbf{r}_\nu) u_{s\mathbf{k}_s}(\mathbf{r}_\nu) \\ &= \sum_{\mu\nu} \chi_{p\mathbf{k}_p}^{(\mu)*} \chi_{q\mathbf{k}_q}^{(\mu)} \zeta_{\mu\nu}^{\mathbf{k}_p, \mathbf{k}_q, \mathbf{k}_r, \mathbf{k}_s} \chi_{r\mathbf{k}_r}^{(\nu)*} \chi_{s\mathbf{k}_s}^{(\nu)}, \end{aligned} \quad (59)$$

where  $\chi_{q\mathbf{k}_q}^{(\mu)} = u_{q\mathbf{k}_q}(\mathbf{r}_\mu)$ ,  $V(\mathbf{r}, \mathbf{r}') = |\mathbf{r} - \mathbf{r}'|^{-1}$ , and

$$\begin{aligned} \zeta_{\mu\nu}^{\mathbf{k}_p, \mathbf{k}_q, \mathbf{k}_r, \mathbf{k}_s} &= \int d\mathbf{r} \int d\mathbf{r}' e^{-i(\mathbf{k}_p - \mathbf{k}_q)\cdot\mathbf{r}} \xi_\mu(\mathbf{r}) \\ &\quad \times V(\mathbf{r}, \mathbf{r}') \xi_\nu(\mathbf{r}') e^{i(\mathbf{k}_s - \mathbf{k}_r)\cdot\mathbf{r}'}. \end{aligned} \quad (60)$$

Some care needs to be taken when bringing this into a form similar to Eq. (5). First recall that we have  $\mathbf{k}_p - \mathbf{k}_q + \mathbf{k}_r - \mathbf{k}_s = \mathbf{G}_{pqrs}$ , where  $\mathbf{G}_{pqrs}$  is a reciprocal lattice vector, and we are working with a uniform  $\Gamma$ -point centered momentum grid with dimensions  $\mathbf{N} = [N_x, N_y, N_z]$  and  $N_k = N_x N_y N_z$ . To eliminate one of the four momentum modes, we identify  $\mathbf{Q} = \mathbf{k}_p \ominus \mathbf{k}_q$ , and  $\mathbf{Q} = \mathbf{k}_s \ominus \mathbf{k}_r$ , and set  $\mathbf{k}_p = \mathbf{k}$ ,  $\mathbf{k}_q = \mathbf{k} \ominus \mathbf{Q}$ ,  $\mathbf{k}_s = \mathbf{k}'$  and  $\mathbf{k}_r = (\mathbf{k}' \ominus \mathbf{Q})$ . To evaluate the  $\zeta$  tensor we still need to know the values of  $\mathbf{k}_p - \mathbf{k}_q$  in absolute terms given a value for  $\mathbf{Q}$  and  $\mathbf{k}$ . We note that mapping the difference  $\mathbf{k}_p - \mathbf{k}_q$  back into our  $k$ -point mesh amounts to adding a specific reciprocal lattice vector  $\mathbf{G}_{pq}^{\mathbf{Q}} = (\mathbf{k}_p - \mathbf{k}_q) - \mathbf{Q} = (\mathbf{k} - (\mathbf{k} \ominus \mathbf{Q})) - \mathbf{Q} \equiv \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}}$ , with a similar expression for  $\mathbf{k}'$  (the subtraction here is not modular). Thus, given a  $\mathbf{Q}$  and  $\mathbf{k}$  we can determine  $\mathbf{k} - \mathbf{Q}$  and  $\mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}}$ . With these replacements we can write

$$\begin{aligned} V_{p\mathbf{k}_p,q\mathbf{k}_q,r\mathbf{k}_r,s\mathbf{k}_s} &\rightarrow V_{p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q}),r(\mathbf{k}'\ominus\mathbf{Q}),s\mathbf{k}'} = \sum_{\mu\nu} \chi_{p\mathbf{k}}^{(\mu)*} \chi_{q\mathbf{k}\ominus\mathbf{Q}}^{(\mu)} \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{k}, \mathbf{k}'} \chi_{r(\mathbf{k}'\ominus\mathbf{Q})}^{(\nu)*} \chi_{s\mathbf{k}'}^{(\nu)} \\ &= \sum_{\mu\nu} \chi_{p\mathbf{k}}^{(\mu)*} \chi_{q\mathbf{k}\ominus\mathbf{Q}}^{(\mu)} \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}}, \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}}} \chi_{r(\mathbf{k}'\ominus\mathbf{Q})}^{(\nu)*} \chi_{s\mathbf{k}'}^{(\nu)}, \end{aligned} \quad (61)$$

where we have used

$$\begin{aligned} \zeta_{\mu\nu}^{\mathbf{k}_p, \mathbf{k}_q, \mathbf{k}_r, \mathbf{k}_s} &\rightarrow \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{k}, \mathbf{k}'} = \int d\mathbf{r} \int d\mathbf{r}' e^{-i(\mathbf{Q} + \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}})\cdot\mathbf{r}} \xi_\mu(\mathbf{r}) V(\mathbf{r}, \mathbf{r}') \xi_\nu(\mathbf{r}') e^{i(\mathbf{Q} + \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}})\cdot\mathbf{r}'} \\ &= \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}}, \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}}}. \end{aligned} \quad (62)$$

In practice there are at most eight values of  $\mathbf{G}$ , so we need only to classically determine at most  $8^2 N_k$  values of  $\zeta$ , as opposed to  $N_k^3$ .

We can then write

$$\begin{aligned} H_2 &= \frac{1}{2} \sum_{\mathbf{Q}, \mathbf{k}, \mathbf{k}'} \sum_{pqrs} \sum_{\sigma\tau} V_{p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q}),r(\mathbf{k}'\ominus\mathbf{Q}),s\mathbf{k}'} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\ominus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\ominus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \\ &= \frac{1}{2} \sum_{\mathbf{Q}, \mathbf{k}, \mathbf{k}'} \sum_{pqrs} \sum_{\sigma\tau} \sum_{\mu\nu} \chi_{p\mathbf{k},\mu}^* \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\mu} \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}}, \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}}} \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\nu}^* \chi_{s\mathbf{k}',\nu} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\ominus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\ominus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \\ &= \frac{1}{2} \sum_{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2} \sum_{\mu\nu} \sum_{\sigma\tau} \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2} \left( \sum_{\mathbf{k} | \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}} = \mathbf{G}_1} \sum_{pq} \chi_{p\mathbf{k},\mu}^* \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\mu} a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\ominus\mathbf{Q})\sigma} \right) \\ &\quad \times \left( \sum_{\mathbf{k}' | \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}} = \mathbf{G}_2} \sum_{rs} \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\nu}^* \chi_{s\mathbf{k}',\nu} a_{r(\mathbf{k}'\ominus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \right), \end{aligned} \quad (63)$$

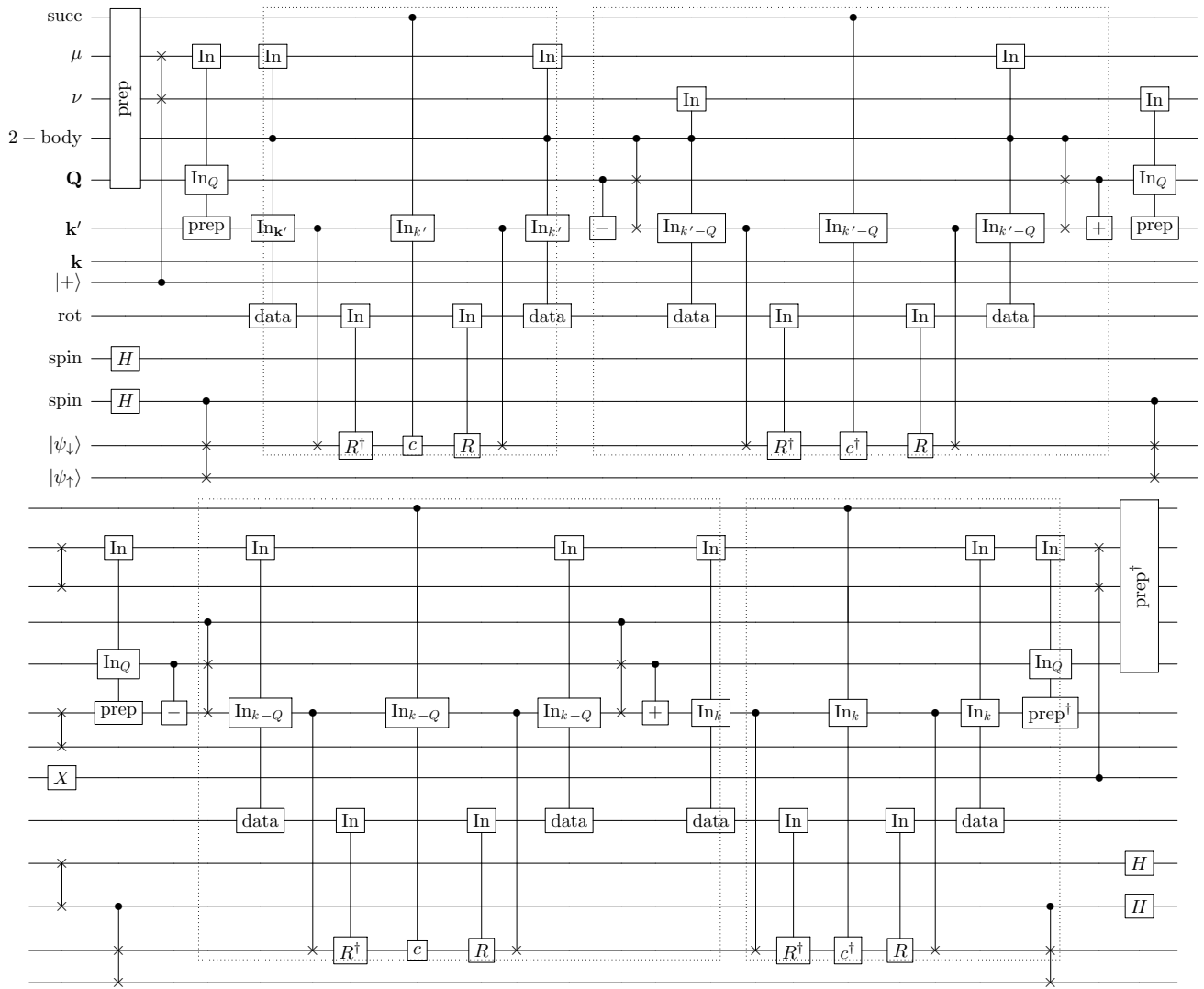


FIG. 7. The quantum circuit for the block encoding of the THC representation, split into two parts with the right half at the bottom. The top shows the portion of the circuit for the first part controlled by  $\mathbf{k}'$  and  $\nu$ , and the bottom shows the (right) part of the circuit where it is controlled by  $\mathbf{k}$  and  $\mu$ . The dotted rectangles show the regions for implementing the  $c$  and  $c^\dagger$  operators together with the Givens rotations needed to change the basis. These steps eventually dominate the cost of the algorithm, with the asymptotic cost scaling like  $O(N_k N)$ . The swaps controlled by the  $\mathbf{k}'$  and  $\mathbf{k}$  registers are to move the appropriate qubits into target registers in order to apply the Givens rotations. The  $c$  and  $c^\dagger$  are applied using a superposition of  $X$  and  $iY$  applied using an ancilla qubit (not shown for simplicity), together with a string of  $Z$  gates for the Jordan-Wigner representation. The preparation at the beginning includes an inequality test between  $\mu$  and  $\nu$  to give a qubit flagging whether the real or imaginary part is produced. To make the implementation self-inverse, the  $\mu$ ,  $\nu$  and  $\mathbf{k}$ ,  $\mathbf{k}'$  pairs of registers are swapped in the middle (the left of the lower half). Also, an  $X$  gate is applied to the qubit that controls the swaps at the beginning and end.

where in going from the second to the third line of Eq. (63) we have rewritten the sum over  $\mathbf{k}$  and  $\mathbf{k}'$  as a double sum over all  $8^2$  values of  $\mathbf{G}_1$  and  $\mathbf{G}_2$ , and a restricted sum on  $\mathbf{k}$  such that for a given  $\mathbf{G}_1$  and  $\mathbf{Q}$  we only sum over those  $\mathbf{k}$ , which satisfy  $\mathbf{G}_{\mathbf{k}, \mathbf{k}-\mathbf{Q}} = \mathbf{G}_1$ . Here the notation  $\mathbf{G}_{\mathbf{k}_p, \mathbf{k}_q}$  is used as equivalent to  $\mathbf{G}_{p,q}$  above. The fourfold symmetry of the two-electron integrals carries over to analogous symmetries in  $\zeta$ , which are listed in Appendix D.

We will then define  $\tilde{\chi}$ , which are individually normalized for each  $\mathbf{k}$  and  $\mu$  so  $\sum_p \tilde{\chi}_{p\mathbf{k},\mu}^* \tilde{\chi}_{p\mathbf{k},\mu} = 1$

and

$$\mathcal{N}_{\mathbf{k},\mu} \tilde{\chi}_{p\mathbf{k},\mu} = \chi_{p\mathbf{k},\mu} \quad (64)$$

with  $\mathcal{N}_{\mathbf{k},\mu} := \sqrt{\sum_p |\chi_{p\mathbf{k},\mu}|^2}$ . We then use these normalized  $\tilde{\chi}$  to give transformed annihilation and creation operators

$$c_{\mu\mathbf{k}\sigma} = \sum_p \tilde{\chi}_{p\mathbf{k},\mu} a_{p\mathbf{k}\sigma}, \quad c_{\mu\mathbf{k}\sigma}^\dagger = \sum_p \tilde{\chi}_{p\mathbf{k},\mu}^* a_{p\mathbf{k}\sigma}^\dagger. \quad (65)$$

We can then write the two-body Hamiltonian as

$$\hat{H}_2 = \frac{1}{2} \sum_{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2} \sum_{\mu, \nu} \sum_{\sigma \tau} \zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2} \times \left( \sum_{\mathbf{k} | \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}} = \mathbf{G}_1} \mathcal{N}_{\mathbf{k}, \mu} \mathcal{N}_{\mathbf{k} \ominus \mathbf{Q}, \mu} c_{\mu \mathbf{k} \sigma}^\dagger c_{\mu (\mathbf{k} \ominus \mathbf{Q}) \sigma} \right) \times \left( \sum_{\mathbf{k}' | \mathbf{G}_{\mathbf{k}', \mathbf{k}' - \mathbf{Q}} = \mathbf{G}_2} \mathcal{N}_{\mathbf{k}' \ominus \mathbf{Q}, \nu} \mathcal{N}_{\mathbf{k}', \nu} c_{\nu (\mathbf{k}' \ominus \mathbf{Q}) \tau}^\dagger c_{\nu \mathbf{k}' \tau} \right). \quad (66)$$

A complication for the implementation is that we would like to be able to choose the relative weighting between  $\zeta$  and  $\chi$  such that

$$\sum_{\mathbf{k} | \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}} = \mathbf{G}} \mathcal{N}_{\mathbf{k}, \mu} \mathcal{N}_{\mathbf{k} \ominus \mathbf{Q}, \mu} = 1. \quad (67)$$

The difficulty here is that the values of  $\mathcal{N}_{\mathbf{k}, \mu}$  depend only on  $\mathbf{k}, \mu$ , because they are based on  $\chi_{p\mathbf{k}, \mu}$ . This sum is also dependent on  $\mathbf{Q}$  and  $\mathbf{G}$ , so for this normalization condition to hold it would mean we need to have  $\chi_{p\mathbf{k}, \mu}$  also dependent on  $\mathbf{Q}$  and  $\mathbf{G}$  in a multiplicative factor (so a non- $\mu$ -dependent way). That will leave the normalized  $\tilde{\chi}_{p\mathbf{k}, \mu}$  unaffected, but means that the values of  $\mathcal{N}_{\mathbf{k}, \mu}$  need to have dependence on  $\mathbf{Q}, \mathbf{G}$ , which will need to be taken account of in the state preparation.

The form in Eq. (66) then gives us a recipe for block encoding the Hamiltonian as a linear combination of unitaries.

1. First prepare a superposition state proportional to

$$\sum_{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2, \mu, \nu} \sqrt{|\zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2}|} |\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2, \mu, \nu\rangle. \quad (68)$$

This state may be prepared via the coherent alias sampling approach with a complexity dominated by the complexity of the QROM. Accounting for symmetry the dimension is about  $32N_k M^2$  and the size of the QROM output is approximately the log of that plus the number of bits for the keep probability. That gives a Toffoli complexity scaling as

$$\sqrt{32N_k M^2 [\log(32N_k M^2) + \aleph]}. \quad (69)$$

2. For each of the two expressions in brackets in Eq. (66), a preparation over  $\mathbf{k}$  or  $\mathbf{k}'$  is needed to give

a state of the form

$$\sum_{\mathbf{k} | \mathbf{G}_{\mathbf{k}, \mathbf{k} - \mathbf{Q}} = \mathbf{G}} \sqrt{\mathcal{N}_{\mathbf{k}, \mu} \mathcal{N}_{\mathbf{k} \ominus \mathbf{Q}, \mu}} |\mathbf{k}\rangle. \quad (70)$$

As explained above, the values of  $\mathcal{N}_{\mathbf{k}, \mu}$  need to be chosen with (implicit) dependence on  $\mathbf{Q}, \mathbf{G}$  for this to be a normalised state. This means that the amplitudes here need to be indexed by  $\mathbf{k}, \mathbf{Q}, \mathbf{G}_1$ , and  $\mu$ . The restricted range of values in the sum over  $\mathbf{k}$  means that the indexing over  $\mathbf{k}, \mathbf{Q}, \mathbf{G}_1$  gives  $N_k^2$  items of data, which is multiplied by  $M$  for the indexing over  $\mu$ . So there are  $N_k^2 M$  items of data needed, which is smaller than that in the first step, because it is missing the factor of 32 and typically  $N_k < M$ . Given that the output size is approximately  $\log(N_k) + \aleph$ , the Toffoli complexity is approximately

$$\sqrt{N_k^2 M [\log(N_k) + \aleph]}. \quad (71)$$

This cost is incurred twice, once for each of the factors in brackets in Eq. (66).

3. For each of the  $c$  annihilation and creation operators we perform a rotation of the basis from  $a$ . This is done in the following way.

- (a) First use the spin  $\sigma$  or  $\tau$  to control a swap of the system registers. This is done once and inverted for each of the two  $c^\dagger c$  factors. Each of these four swaps has cost  $N_k N / 2$ .
- (b) Then use  $\mathbf{k}$  or  $\mathbf{k} \ominus \mathbf{Q}$  to control the swap of the registers we wish to act on into working registers. The value of  $\mathbf{k} \ominus \mathbf{Q}$  is used for  $c_{\mu (\mathbf{k} \ominus \mathbf{Q}) \sigma}$ , and needs to be computed to use as a control. Each of these eight swaps may be done with a Toffoli complexity approximately as half the number of system registers  $N_k N / 2$ .
- (c) Next  $\mathbf{k}$  (or  $\mathbf{k} \ominus \mathbf{Q}$ ) and  $\mu$  (or  $\nu$ ) are used as a control for a QROM to output the angles for Givens rotations. There are two angles for each of  $N/2$  Givens rotations, so if they have  $\beth$  each the size of the output is  $N \beth$ . Then the QROM complexity is about

$$\sqrt{N_k N^2 \beth}. \quad (72)$$

This must be done 4 times (and has a smaller erasure cost).

- (d) The sequence of  $N/2$  Givens rotations is performed, each with four individual rotations on  $\beth$ , for a cost of  $2N \beth$ . This cost is incurred 8 times, twice for each of the annihilation and creation operators.



- After the rotation of the basis, we simply need to perform the linear combination of  $\vec{Z}X$  and  $\vec{Z}Y$  for  $c^\dagger$  and  $c$ . The  $X$  or  $Y$  is applied in a fixed location, but the  $\vec{Z}$  needs to be applied on a range of qubits chosen by  $\mathbf{k}$  or  $\mathbf{k} \ominus \mathbf{Q}$ . We therefore have approximately  $N_k$  for the unary iteration for each  $\vec{Z}$  for a total cost of about  $4N_k$ .

Lastly we would perform reflections on control ancillas as usual to construct a qubitized quantum walk from the block encoding. This cost is trivial compared to that in the other steps. For a more detailed explanation, see the circuit diagram in Fig. 7 and the discussion in Appendix D.

The  $\lambda_{\text{THC}}$  value has a one-body component and two-body component. Unlike molecular THC where the two-body component is reduced because we evolve by number operators in the nonorthogonal basis, in this version of the THC algorithm we will evolve by ladder operators in a nonorthogonal basis, and thus there is no one-body part to remove. The one-body contribution to  $\lambda_{\text{THC}}$ ,  $\lambda_{\text{THC},1}$ , is computed in a similar way as for the double factorization algorithm but noting that the extra factor of 1/2 coming from the  $Z$  operator is no-longer present to cancel the factor of 2 from spin summing. The one-body contribution to  $\lambda_{\text{THC}}$  is

$$\lambda_{\text{THC},1} = 2 \sum_{\mathbf{k}} \sum_p |\lambda_{p,\mathbf{k}}|. \quad (73)$$

The two-body contribution to  $\lambda_{\text{THC}}$ ,  $\lambda_{\text{THC},2}$ , is determined by summing over all unitaries in the LCU. This summation can be rewritten in the form

$$\begin{aligned} \lambda_{\text{THC},2} = & 2 \sum_{\mathbf{Q}} \sum_{\mu,\nu} \sum_{\mathbf{G}_1,\mathbf{G}_2} (|\text{Re}[\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2}]| + |\text{Im}[\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2}]|) \\ & \times \left( \sum_{\mathbf{k}|\mathbf{G}_{\mathbf{k},\mathbf{k}\ominus\mathbf{Q}}=\mathbf{G}_1} \mathcal{N}_{\mathbf{k},\mu} \mathcal{N}_{\mathbf{k}\ominus\mathbf{Q},\mu} \right) \\ & \times \left( \sum_{\mathbf{k}'|\mathbf{G}_{\mathbf{k}',\mathbf{k}'\ominus\mathbf{Q}}=\mathbf{G}_2} \mathcal{N}_{\mathbf{k}'\ominus\mathbf{Q},\nu} \mathcal{N}_{\mathbf{k}',\nu} \right) \end{aligned} \quad (74)$$

using the expression for  $\zeta$  described in Eq. (62).

To obtain resource estimates for THC with  $k$  points we follow a similar procedure to previous molecular work [33] and first compress the rank of the THC factors ( $M = c_{\text{THC}}N/2$ , where  $c_{\text{THC}}$  is the THC rank parameter). In particular, we use the interpolative separable density fitting (ISDF) approach [88,90,91] as a starting point before subsequently reoptimizing these factors in order to compress the THC rank while regularizing  $\lambda$  [33,46], which we will call  $k$ -THC. Further details of this procedure are provided in Appendix G. In Fig. 8 we demonstrate that a  $c_{\text{THC}} = 8$  is sufficient to obtain MP2 correlation energies

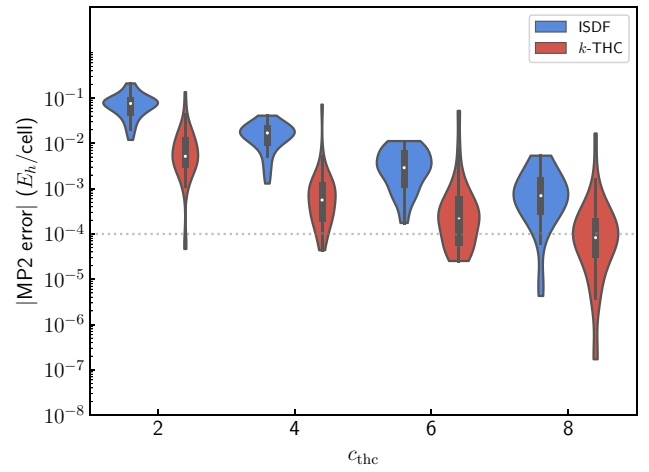


FIG. 8. Violin plot of absolute errors in the  $k$ -THC-MP2 energy per cell for the benchmark set in Table I. Here we compare the MP2 errors as a function of the THC rank parameter  $c_{\text{THC}}$  using ISDF or subsequent reoptimization to generate the THC factors.

within approximately 0.1 mHa/cell for a subset of the systems considered in the benchmark set. We note that the equivalent ISDF rank may be on the order of 10-15 for comparable accuracy, which would correspond to a much larger value for  $\lambda$ .

Figure 9(a) demonstrates a  $\sqrt{N_k}$  scaling improvement of the block encodings in the symmetry-adapted case. Note that this  $\sqrt{N_k}$  speedup for the block encodings is partially a finite-size effect. In Fig. 10 we plot the Toffoli complexity per step as a function of  $N_k$  using artificially generated data to explore the large  $N_k$  behavior. We see that depending on the fitting range employed, the extracted scaling trends towards the expected linear regime. It is interesting to note that although asymptotically we expect the THC algorithm to scale linearly with the system size, this is not observed in practice, with the cost of controlled rotations dominating over the cost of controlled swaps. While ultimately both the symmetry-adapted and supercell encodings should scale linearly with the system size due to the cost of unary iteration over all basis states, there are several factors that yield a  $\sqrt{N_k}$  saving in the symmetry-adapted case, and the relative size of the prefactors becomes important. Similar to DF, we find from Fig. 9(b) that  $\lambda$  in the symmetry-adapted setting exhibits slightly worse scaling than for supercell calculations. This worsening of  $\lambda$  in the symmetry-adapted case can be understood again as a reduction in variational freedom in the symmetry-adapted case, leading to smaller compression. Note that while Eq. (74) nominally scales cubically with  $N_k$ , we expect each individual matrix element to decay like  $N_k^{-1}$ , which yields the expected quadratic dependence of  $\lambda$ , or a linear dependence of  $\lambda$  when targeting the total energy per cell. In the supercell case, there are simply  $M^2 = (N_k N)^2$  elements

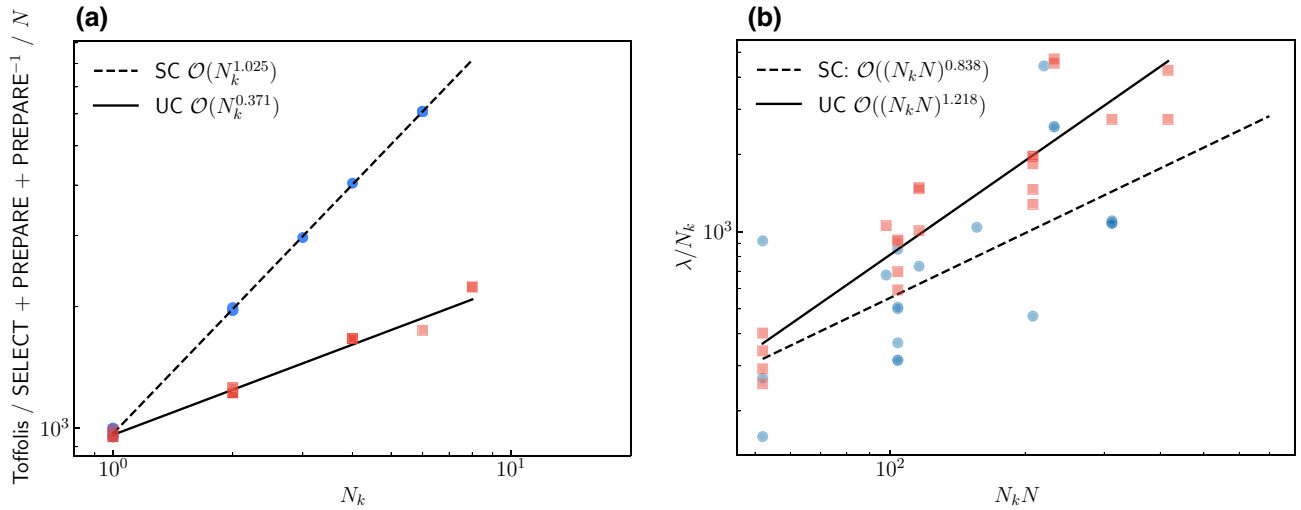


FIG. 9. (a) The number of  $k$  points versus Toffoli cost to implement the block encoding for the THC factorization LCU evaluated for the benchmark systems listed in Table I described using the cc-pVDZ and cc-pVTZ basis sets and  $\Gamma$ -centered Monkhorst-Pack grids of size [1, 1, 1] to [3, 3, 3]. Each point is a single system described at a particular basis set and  $k$  mesh where the range of the auxiliary index of the THC factorization is selected to produce two-electron integrals corresponding to an MP2 error of one millihartree with respect to an untruncated auxiliary index range. This corresponds to an auxiliary index that is 8 times the number of orbitals in the primitive cell for symmetry adapted THC, and eight times the total number of orbitals ( $N_k N$ ) for supercell THC. We divide the Toffoli complexity for implementing SELECT + PREPARE + PREPARE<sup>-1</sup> by  $N$ , which is the shared scaling in the number of bands. While we observe a  $\sqrt{N_k}$  scaling improvement for symmetry-adapted walk operations, we believe this is a finite-size effect and both methods should scale linear with  $N_k$  for sufficiently large  $N_k$ . The value of  $\lambda$  as a function of the total system size  $NN_k$  for the same systems described with the same cutoffs used in (a). The reduced variational freedom in compression of the two-electron integral tensors for the symmetry-adapted walk operator construction translates to an increased value of  $\lambda$  at all system sizes.

in the central tensor, which in turn controls the scaling of  $\lambda$ . From Table II and Fig. 9 we can conclude that there is asymptotically no advantage to incorporating symmetry in the THC factorization for the Toffoli complexity, with both the supercell and symmetry-adapted methods exhibiting approximately quadratic scaling with system size for a fixed target accuracy of the total energy per cell.

#### IV. SCALING COMPARISON AND RUNTIMES FOR DIAMOND

We now compare runtimes and estimate total physical requirements to simulate diamond as a representative material. In Fig. 11 we plot the total Toffoli complexity for the sparse, SF, DF, and THC LCUs using symmetry-adapted block encodings and supercell calculations for diamond with cc-pVDZ and cc-pVTZ basis sets at various Monkhorst-Pack samplings. In sparse and SF there is a clear asymptotic separation between supercell and symmetry-adapted Toffoli counts. This is expected from the fact that both block-encoding constructions are asymptotically improved and  $\lambda$  does not increase. For the DF case, total Toffoli complexity for supercell and symmetry-adapted cases is similar due to the larger  $\lambda$  for the symmetry-adapted algorithm. For THC, the total Toffoli complexity is similar in the supercell and symmetry-adapted case, but the asymptotic scaling is identical for the

supercell and symmetry-adapted algorithms. This is due to the increase in  $\lambda$  for the symmetry-adapted algorithm.

In Table III we tabulate the quantum resource requirements and estimated runtimes after compiling into a surface code using physical qubits with error rates of 0.01% and a 1  $\mu$ s cycle time. We assume four Toffoli factories similar to Refs. [33,46] and observe that for systems with 52-1404 spin orbitals the quantum resource estimates are roughly in line with extrapolated estimates from the molecular algorithms.

It is important to note that while the THC resource requirements look competitive for these small systems, in its current form it is not a practical way to simulate materials at scale. This is due to the prohibitive cost of reoptimizing the THC factors, which significantly limits the system sizes that can be simulated. Moreover, as discussed in Sec. III D, we caution that the THC trend lines are only valid within the fitting range, and we expect that asymptotic THC Toffoli count will trend more towards  $\mathcal{O}(N_k^2)$  in the thermodynamic limit.

#### V. CLASSICAL AND QUANTUM SIMULATIONS OF LNO

In this section, we compare modern classical computational methods with quantum resource estimates in the

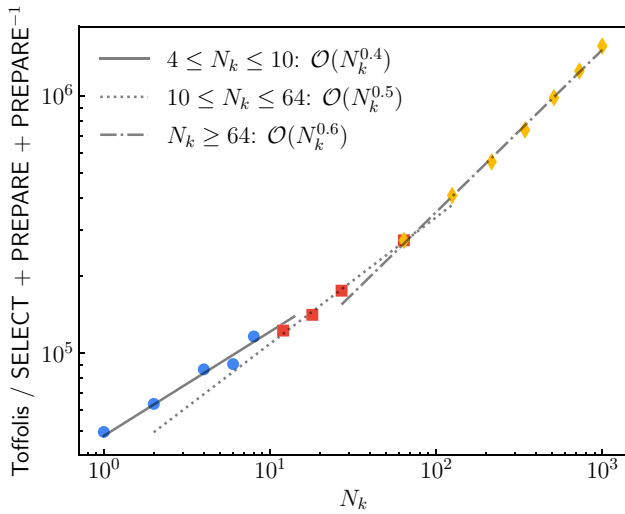


FIG. 10. Synthetic data for the number of Toffolis required to implement the qubitization oracles with the  $k$ -THC factorization demonstrating the challenge of extracting the correct asymptotic scaling with limited finite-size data. To generate the data we used the system parameters of carbon diamond in the cc-pVDZ basis set ( $N = 52$ ,  $M = 208$ ).

context of a challenging problem of industrial interest: the ground state of  $\text{LiNiO}_2$ .

### A. LNO background

Layered oxides have been the most popular cathode active materials for Li-ion batteries since their commercialization in the early 1990s. While  $\text{LiCoO}_2$  is still the material of choice in the electronics industry, the increasing human, environmental, and financial cost of cobalt

spells out the need for cobalt-free cathode active materials, especially for automotive applications [92,93].

The isostructural compound  $\text{LiNiO}_2$  (LNO) had been identified as an ideal replacement for  $\text{LiCoO}_2$  already in the 1990s, due to its comparably high theoretical capacity at a lower cost [94,95]. Despite its numerous drawbacks, LNO still serves as the perfect model system for many derivative compounds, such as lithium nickel-cobalt-manganese (NCM) and lithium nickel-cobalt-aluminum oxides (NCA) that are nowadays the gold standard in the automotive industry [47]. Moreover, the constant demand for better performing materials pushes the amount of substituted Ni to the dilute regime and the research trend is approaching the asymptotic  $\text{LiNiO}_2$  limit, making  $\text{LiNiO}_2$  a system of interest in battery research [47].

Even the nature of the ground state of LNO is still under debate. The universally observed rhombohedral  $R\bar{3}m$  symmetry [47], with Ni being octahedrally coordinated to six oxygen atoms through six equivalent Ni—O bonds conflicts with the renowned Jahn-Teller (JT) activity of low-spin trivalent Ni, which has been experimentally proven on a local scale [47]. In a recent DFT study [48], we argued that this apparent discrepancy might be resolved by the dynamics and low spatial correlation of Jahn-Teller distortions. In that work, the energy distance between Jahn-Teller distorted and nondistorted candidates (Fig. 12) compared to zero-point vibrational energies makes a strong argument in favor of the dynamic Jahn-Teller effect. A non-JT distorted structure resulting from the disproportionation of  $\text{Ni}^{3+}$  has also been reported as a ground-state candidate [49] despite the 1:1 ratio between long and short Ni—O bonds, which conflicts with the experimentally determined 2:1 ratio. In the original study, the stability of this structure has been found to depend heavily on the

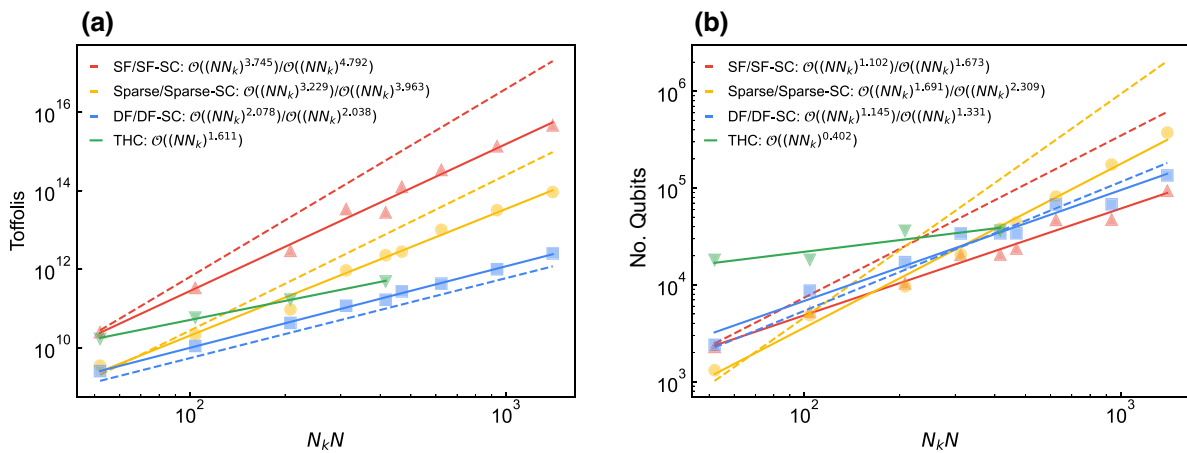


FIG. 11. (a) Total Toffoli requirements for diamond in a cc-pVDZ basis at various Monkhorst-Pack samplings of the Brillouin zone with  $\Gamma$ -point centered grids of size  $[1, 1, 1]$  to  $[3, 3, 3]$ . Dashed lines are fits to the supercell data that is not plotted. Solid lines are fits to the symmetry-adapted data shown as data points. (b) Total logical qubits for symmetry-adapted oracles and supercell (dotted lines). All values are estimated from 0.1 mHa per unit cell thresholds on the MP2 energy. In the case of THC we plot only the symmetry-adapted data due to limited THC data arising from difficulty in optimizing the supercell THC factors.

TABLE III. Diamond represented in a cc-pVDZ basis (52 spin orbitals in the primitive cell) at various  $k$ -mesh sizes and the associated quantum resource requirements to compute the total energy per cell to within 1 kcal/mol. The surface code runtime is estimated using four T-factories, a physical error rate of 0.01%, and a cycle time of 1  $\mu$ s [33,46]. The physical qubit count is given in millions.

LCU	$k$ mesh	Toffolis	Logical qubits	Physical qubits (M)	Surface code runtime (days)
Sparse	[1, 1, 1]	$4.84 \times 10^9$	2478	2.20	$9.10 \times 10^{-1}$
	[2, 2, 2]	$2.66 \times 10^{12}$	75287	90.57	$5.77 \times 10^2$
	[3, 3, 3]	$1.06 \times 10^{14}$	374274	543.76	$2.61 \times 10^4$
SF	[1, 1, 1]	$3.20 \times 10^9$	2283	2.05	$6.02 \times 10^{-1}$
	[2, 2, 2]	$3.27 \times 10^{12}$	20567	24.91	$7.11 \times 10^2$
	[3, 3, 3]	$1.13 \times 10^{15}$	47665	69.52	$3.10 \times 10^5$
DF	[1, 1, 1]	$9.61 \times 10^8$	2396	1.55	$1.81 \times 10^{-1}$
	[2, 2, 2]	$6.74 \times 10^{10}$	18693	18.47	$1.27 \times 10^1$
	[3, 3, 3]	$1.09 \times 10^{12}$	68470	82.39	$2.37 \times 10^2$
THC	[1, 1, 1]	$1.67 \times 10^{10}$	18095	14.20	3.14
	[2, 2, 2]	$4.85 \times 10^{11}$	36393	35.60	$1.05 \times 10^2$

value of the on-site Hubbard correction applied to the PBE functional. With the SCAN-rVV10 functional (with and without on-site Hubbard correction) [48], this candidate is consistently less stable than the JT-distorted models; it is also worth mentioning that the on-site Hubbard correction considerably increases the stability of the JT-distorted models. The dependence of Jahn-Teller stabilization energies on the functional had already been observed by Radin [96] and is ascribed to the difficulty to adequately describe the doubly degenerate high-symmetry, undistorted state.

In light of previous studies, we will focus on four candidate structures for the LNO ground state. These structures are shown in Fig. 12. We will furthermore focus only on the energetics of the problem. The goal is to compute the relative energies of these different crystal structures without the uncertainty of DFT.

## B. Correlated $k$ -point calculations

Local-basis quantum chemistry methods for electron correlation have been increasingly applied to periodic systems as an alternative to DFT with more controllable accuracy. Here we apply two such methods, second-order Møller-Plesset perturbation theory (MP2) [97,98] and coupled cluster singles and doubles (CCSD) [99,100], to the three distorted structures of LNO (Fig. 12). Local basis methods like these can be directly compared to quantum algorithms described in this work, since both are formulated within the same framework of a crystalline Gaussian one-particle basis. While these methods cannot be easily applied to the symmetric structure, which is metallic at the mean-field level, they should provide accurate results for the distorted structures provided that the finite-size and finite-basis errors can be controlled. All mean field, MP2

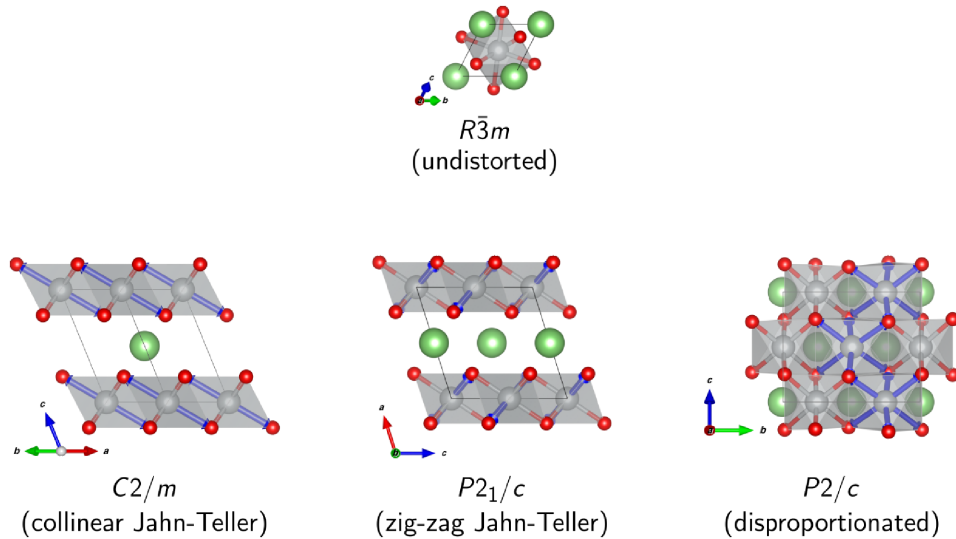


FIG. 12. The four known  $\text{LiNiO}_2$  polymorphs: high-symmetry  $R\bar{3}m$ , collinear JT-distorted  $C2/m$ , zig-zag JT-distorted  $P2_1/c$ , and disproportionated  $P2/c$ . Green spheres represent Li, gray polyhedra are  $\text{NiO}_6$  octahedra, and elongated Ni—O bonds are depicted as bold blue arrows.

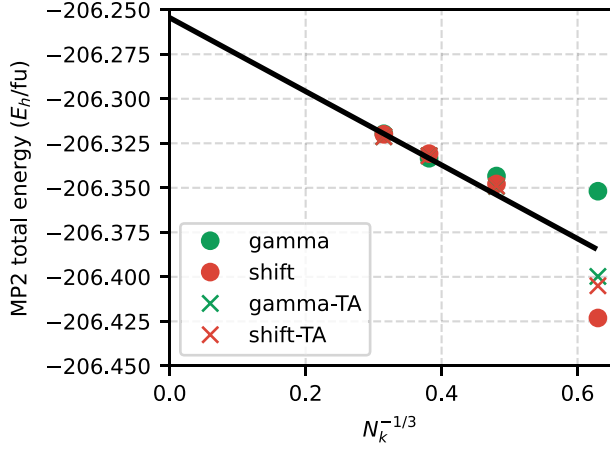


FIG. 13. Convergence of the total MP2 energy for the  $C2/m$  structure using the minimal basis set as a function of  $n_k^{-1/3}$ , where  $n_k$  is the number of  $k$  points.  $\Gamma$ -centered and shifted  $k$ -point grids are compared with and without twist averaging (TA).

and CCSD calculations were performed with the PySCF program package [101,102]. QMCPACK [103,104] was used to perform the ph-AFQMC calculations, where we used at least 600 walkers and a timestep of  $0.005 \text{ Ha}^{-1}$ . The population control bias was found to be negligible. In all calculations, we use separable, norm-conserving GTH pseudopotentials [74,105] that have been recently optimized for Hartree-Fock [106]. In all calculations on LNO we use the GTH basis sets [50,107] (GTH-SZV and GTH-DZVP specifically) that are distributed with the CP2K [51] and PySCF [102] packages. In Fig. 13 we show the convergence of the minimal-basis MP2 energy as a function of effective cell size for increasingly large  $k$ -point calculations. This demonstrates the essential difficulty in converging to the bulk limit for correlated calculations: the finite-size error will converge slowly with the system size. Though the MP2 energy should eventually scale with the inverse volume ( $N_k^{-1}$ ) [108], the apparent scaling is more like  $N_k^{-1/3}$  for a small number of  $k$  points. Shifting the  $k$ -point grid to  $(1/8, 1/8, 1/8)$  and/or twist averaging (TA) does not change the asymptotic behavior of the energy. In all other LNO calculations, we use  $\Gamma$ -centered  $k$ -point grids. In all calculations, the density of  $k$  points along each reciprocal lattice vector was chosen so that the density of  $k$  points is as close to constant as possible.

TABLE IV. ROHF total energy and MP2, CCSD, and ph-AFQMC correlation energies computed with a double- $\zeta$  basis set and a small  $k$  mesh, the equivalent of four primitive formula units (16 atoms total) for each distorted structure. All units are Hartrees per formula unit.

Structure	$k$ points	ROHF	MP2	CCSD	ph-AFQMC
$C2/m$	$2 \times 2 \times 1$	-206.557491	-0.750524	-0.767350	-0.7997(5)
$P2_1/c$	$1 \times 2 \times 1$	-206.567049	-0.747717	-0.765445	-0.7982(5)
$P2/c$	$1 \times 1 \times 1$	-206.551767	-0.811386	-0.780580	-0.8078(4)

TABLE V. Energies in meV relative to the energy of the  $P2_1/c$  structure for each method. The  $P2_1/c$  structure is lowest in energy for all methods except MP2.

Structure	ROHF	MP2	CCSD	ph-AFQMC
$C2/m$	260	184	208	218(18)
$P2/c$	416	-1317	4	155(17)

While a minimal basis is useful for a qualitative understanding of the finite-size error, it does not provide sufficient accuracy to resolve the different LNO structures examined in this work. The double- $\zeta$  basis set (GTH-DZVP) is large enough to provide qualitative accuracy, but converging the result to the bulk limit is prohibitively expensive for the systems considered here. We can nonetheless provide some estimates of the ground-state CCSD and MP2 (DZVP) energies as shown in Table IV and V. Since there is no evidence of particularly “strong correlation” in any of these systems (see Appendix E for a more detailed discussion), MP2 and CCSD should provide qualitatively correct estimates of the ground state energy. The unusually large MP2 correlation energy for the  $P2/c$  structure suggests it may not be as reliable for this structure, and this suspicion is confirmed by the CCSD and ph-AFQMC calculations. For CCSD and ph-AFQMC, the  $P2_1/c$  structure is lowest in energy, which agrees qualitatively with the DFT calculations in Ref. [48]. However, this prediction carries with it a great deal of uncertainty due to the small simulation size, small one-particle basis set, and error in the MP2/CCSD/ph-AFQMC approximations.

### C. Single-shot density-matrix embedding theory

Another way to apply high-level correlated methods to periodic solids is through quantum embedding methods in which a local *impurity* is treated with a high-level method and the remainder of the system, the *bath*, is treated at a lower level of theory. For periodic solids, dynamical mean-field theory (DMFT) is perhaps the most widely successful such method [21,109–112]. Density-matrix embedding theory (DMET) is an efficient quantum embedding method for the ground state of quantum systems [113,114], and it has recently been applied to periodic solids with a fully *ab initio* Hamiltonian [21].



Although very large impurities are necessary to converge to the bulk limit of the correlated method used for the impurity, a fixed impurity size provides a local, systematically improvable approximation to the correlation energy. This is particularly useful in cases where a local treatment of correlation is sufficient for a qualitatively correct solution. Here we apply DMET with a CCSD impurity solver to the distorted structures in a minimal basis set (GTH-SZV). The libdmnet code [20] was used for the DMET calculations with the PySCF program package [101,102] used in the impurity solver.

Figure 14 shows the minimal-basis DMET results for each of the three distorted structures. In the context of DMET, we can effectively converge the mean-field part of the problem. Unfortunately, the small one-particle basis set and modest impurity size make it unable to meaningfully resolve these three structure of  $\text{LiNiO}_2$ . Quantum simulation can potentially overcome some of these limits by acting as a lower scaling, unbiased impurity solver [115].

#### D. Quantum resource estimates for LNO

Quantum resource estimates for LNO using the SF and DF LCUs are reported in Table VI. THC is not reported due to the difficulty of reoptimizing the THC tensors to have low L1 norm as discussed in Refs. [33,46,87]. For the

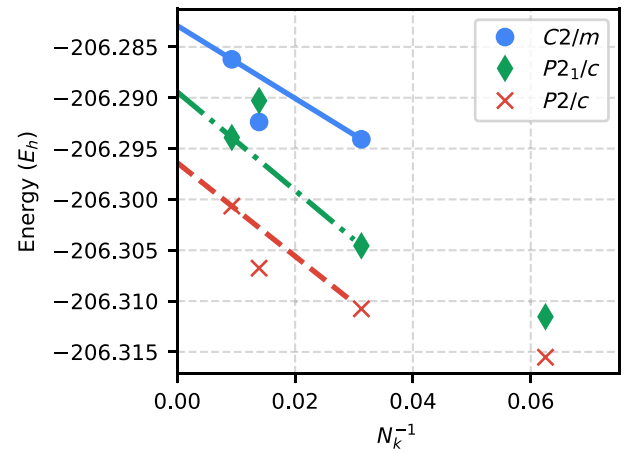


FIG. 14. Convergence of the total DMET energy for a four-formula unit (16-atom) impurity with respect to effective size of the mean-field calculation for the different distorted structures.

sparse LCU, a threshold of  $1 \times 10^{-4}$  was determined by averaging the thresholds for the systems in Table I required to achieve 1  $mE_H$  per unit cell. For the SF LCU, the truncation of the auxiliary basis was set to 8 times the number of molecular orbitals, which was determined by requiring the error in the MP2 energy for the smallest  $C2/m$

TABLE VI. Quantum resource estimates for all four LNO structures normalized by the number of formula units represented in each simulation cell.  $R\bar{3}m$  and  $C2/m$  are both one formula unit while  $P2/c$  is four formula units and  $P2_1/c$  is two formula units. The sparse threshold is selected to be  $1.0 \times 10^{-4}$ , the SF the auxiliary index is truncated at eight times the number of molecular orbitals, and the DF the second factorization is truncated at  $1.0 \times 10^{-4}$ .

System	LCU	$k$ mesh	$\lambda$	No. spin orbs.	Toffolis	Logical qubits	Physical qubits (M)	Runtime (days)
$R\bar{3}m$	Sparse	[2, 2, 2]	120382.037	116	$6.16 \times 10^{13}$	166946	242.72	$1.51 \times 10^4$
		[3, 3, 3]	718377.133	116	$3.57 \times 10^{15}$	1625295	2808.82	$9.82 \times 10^5$
	SF	[2, 2, 2]	183778.821	116	$7.86 \times 10^{13}$	89162	129.77	$1.93 \times 10^4$
		[3, 3, 3]	2966279.293	116	$4.60 \times 10^{15}$	404723	699.68	$1.27 \times 10^6$
	DF	[2, 2, 2]	10730.422	116	$4.97 \times 10^{12}$	149939	180.16	$1.08 \times 10^3$
		[3, 3, 3]	44794.803	116	$7.28 \times 10^{13}$	598286	869.02	$1.79 \times 10^4$
$C2/m$	Sparse	[2, 2, 1]	58422.522	116	$1.03 \times 10^{13}$	83532	100.47	$2.53 \times 10^3$
		[4, 4, 2]	893339.394	116	$5.37 \times 10^{15}$	3051285	5272.93	$1.48 \times 10^6$
	SF	[2, 2, 1]	95803.204	116	$2.05 \times 10^{13}$	44657	53.90	$5.05 \times 10^3$
		[4, 4, 2]	2899609.300	116	$5.23 \times 10^{15}$	405310	700.69	$1.44 \times 10^6$
	DF	[2, 2, 1]	4873.648	116	$1.18 \times 10^{12}$	75178	90.44	$2.56 \times 10^2$
		[4, 4, 2]	51416.281	116	$9.82 \times 10^{13}$	598736	869.68	$2.41 \times 10^4$
$P2/c$	Sparse	[1, 1, 1]	84977.359	464	$2.06 \times 10^{13}$	99918	120.21	$5.07 \times 10^3$
		[2, 2, 2]	1627121.892	464	$1.67 \times 10^{16}$	3182362	6454.14	$4.59 \times 10^6$
	SF	[1, 1, 1]	201894.726	464	$8.74 \times 10^{13}$	92786	135.04	$2.15 \times 10^4$
		[2, 2, 2]	5666363.179	464	$2.07 \times 10^{16}$	839487	1450.95	$5.68 \times 10^6$
	DF	[1, 1, 1]	2753.901	464	$9.72 \times 10^{11}$	75834	91.23	$2.11 \times 10^2$
		[2, 2, 2]	40788.113	464	$1.40 \times 10^{14}$	1192900	1732.40	$3.44 \times 10^4$
$P2_1/c$	Sparse	[1, 2, 1]	105584.297	232	$3.39 \times 10^{13}$	182864	265.83	$8.34 \times 10^3$
		[2, 4, 2]	1714723.913	232	$1.50 \times 10^{16}$	3116825	6321.24	$4.12 \times 10^6$
	SF	[1, 2, 1]	271178.934	232	$8.92 \times 10^{13}$	96882	140.98	$2.19 \times 10^4$
		[2, 4, 2]	7798992.981	232	$2.13 \times 10^{16}$	438080	757.32	$5.85 \times 10^6$
	DF	[1, 2, 1]	3958.111	232	$1.27 \times 10^{12}$	75383	90.69	$2.76 \times 10^2$
		[2, 4, 2]	46189.645	232	$1.23 \times 10^{14}$	1192758	1732.20	$3.02 \times 10^4$

system to be less than  $1 mE_h$  per formula unit. For DF, the same requirement was used to determine a cutoff for the second factorization of  $1 \times 10^{-3}$ . The trends are consistent with what was observed in Sec. IV: DF is consistently more efficient than either sparse or SF LCUs.

## VI. CONCLUSION

In this work we developed the theory of symmetry-adapted block encodings for extended system simulation using four different representations of the Hamiltonian as LCUs in order to improve quantum resource costs for reaching the thermodynamic limit when simulating solids. In order to realize an asymptotic speedup due to symmetry, we substantially modify the block encodings compared with their molecular counterparts. To demonstrate these asymptotic improvements we compiled constant factors for all four LCUs and compared their performance on a suite of benchmark systems and a realistic problem in materials simulation. We find that despite a clear asymptotic speedup for walk operator construction there are competing factors (such as lower compression in Hamiltonian tensor factorizations) that make it difficult to observe a large speedup using symmetry. It was recently shown that variationally constructing tensor compressions for Hamiltonian simulation can improve quantum resource requirements [46,87,116] and thus we believe the compressions can be improved to ultimately demonstrate a speedup for these types of simulations.

For the sparse and SF LCUs we derive a  $\mathcal{O}(\sqrt{N_k})$  speedup in constructing SELECT and PREPARE by ensuring only the minimal amount of symmetry unique information is accessed by the quantum circuit through QROM. In both cases a speedup is observable, though it is much clearer in the SF case. Observing the sparse LCU speedup is more challenging due to the difficulty of converging the  $N_k$  and  $N$  dependence of the two-electron integrals. Compared with the molecular case where sparse was competitive with the DF and THC algorithms [33,79], largely due to the simplicity of SELECT, we find that sparse is not viable for converging to the thermodynamic limit of solids.

The DF and THC tensor factorizations yield LCUs as unitaries in nonorthogonal bases and lead to much higher compression than sparse and SF LCUs. In the DF case we derive an asymptotic  $\mathcal{O}(\sqrt{N_k})$  improvement in Toffoli complexity and qubit cost when constructing the qubitization walk operator. Unfortunately,  $\lambda$  is increased in these cases. The increase is attributed to the lower variational freedom in constructing nonorthogonal bases when representing the two-electron integral tensor in factorized form compared with the nonsymmetry-adapted setting. For the THC case, no asymptotic speedup is formally possible. This stems from the linear cost of unary iteration over all basis states. Nevertheless, due to competing

prefactors between unary iteration and state preparation, we do observe a  $\sqrt{N_k}$  scaling improvement in the Toffoli per step and logical qubit cost for the range of systems studied. This is likely a finite-size effect, but may be practically important when considering which algorithm to choose in the future. Thus, improving the  $\lambda$  value of THC through more sophisticated and affordable means is worth further investigation.

Reaching the thermodynamic and complete basis-set limit is very challenging, even for classical wavefunction methods like CCSD and ph-AFQMC. Previous ph-AFQMC results for simple insulating solids with two-atom unit cells suggest that at least a  $3 \times 3 \times 3$  and  $4 \times 4 \times 4$  sampling of the Brillouin zone is required to extrapolate correlation energies to the thermodynamic limit [117]. Similarly, it has been found that quadruple- $\zeta$  quality basis sets are required to converge the cohesive energy to less than 0.1 eV/atom, while a triple- $\zeta$  quality basis is likely sufficient for quantities such as the lattice constant and bulk modulus [118]. Similar system sizes and basis sets were found to be required for CCSD simulations of metallic systems [19]. Although the theory of finite-size corrections [119–122] is still an area of active research [123,124], the simulation of bulk systems even with these corrections typically requires on the order of 50 atoms, which in turn corresponds to hundreds of electrons and thousands of orbitals. For excited-state properties, particularly those concerning charged excitations, even larger system sizes may be required without the use of sophisticated finite-size correction schemes [125]. Thus, we suspect that simulating large system sizes will continue to be necessary in order to obtain high accuracy for condensed phase simulations. It is important to note that high-accuracy classical wavefunction methods are often considered too expensive for practical materials simulation, and DFT is still the workhorse of the field. Appendix F shows that simulating even simple solids with coarse  $k$  meshes can take on the order of hours, which would otherwise take seconds for a modern DFT code. From the quantum computing perspective, it is clear that several orders of magnitude of improvement in resources are necessary before practical materials simulation is possible. Despite this, the fairly low scaling of phase estimation as a function of system size serves as encouragement to pursue quantum simulation for materials further.

Our estimates begin to demarcate the boundary for quantum advantage in the context of *ab initio* materials simulation. In particular, taking ph-AFQMC as a proxy for a high-accuracy classical method, we see comparing Table III and Fig. 16 that we would need roughly 2 orders of magnitude runtime improvements in the quantum algorithm for it to become competitive from a wall-time perspective. With that being said, it is important to caveat this estimate with the fact that one would not necessarily need to use a quantum computer (or QMC) to simulate

a largely single reference system like carbon, and, more importantly, for strongly correlated systems this boundary is likely much closer as the runtime of methods like QMC will, in principle, scale exponentially if more sophisticated trial wave functions are required in order to obtain a fixed accuracy. Failing any further algorithmic breakthroughs, it is likely that quantum computers will be most effective for simulating a fragment of the whole system with dynamical correlation being accounted for using some approximate (classical) approach or through embedding.

The aforementioned convergence difficulties are demonstrated in our classical calculations on the LNO system when attempting to resolve the discrepancy between band-theory predictions and experimental observations of the ground-state geometry. Furthermore, the variance in energy between CCSD, MP2, ph-AFQMC, and DMET (and their expenses) make it difficult to select an efficient method for determining Hamiltonian parameter cutoffs to use in quantum resource estimation. If anything, this highlights the need for high-accuracy classical computation when performing quantum resource estimates and ultimately picking an algorithm for quantum simulation. The quantum resource estimates for LNO simulations are exorbitantly expensive even at small  $k$  mesh; estimated to run in  $\mathcal{O}(10^2) - \mathcal{O}(10^3)$  days using the DF LCU and requiring  $\mathcal{O}(10^6)$  logical qubits and  $\mathcal{O}(10^{14})$  Toffoli gates. Just as resource estimates for chemistry fell drastically with algorithmic developments clearly further algorithmic improvements are needed to make a LNO-sized problem feasible on a quantum computer.

Qubitization is a general tool for Hamiltonian simulation and there may be other simulation scenarios when the improved walk operators yield faster simulations. There are also areas to further improve the quantum algorithms by taking advantage of space-group symmetry along with translational symmetry. In classical calculations this can lead to substantial computational savings even at the mean-field level. Just as in the case of quantum algorithms for molecular simulations, we expect the quantum resource costs to fall with further exploration of algorithmic improvements.

## ACKNOWLEDGMENTS

The authors thank Yuan Su for helpful conversations. F.D.M. thanks Miguel Morales for helpful discussions on the form of the  $k$ -point THC factorization. D.W.B. worked on this project under a sponsored research agreement with Google Quantum AI. D.W.B. is also supported by Australian Research Council Discovery Projects DP190102633 and DP210101367.

## APPENDIX A: SPARSE REPRESENTATION DERIVATIONS

### 1. The Pauli-operator representation of the one-body term

Here we derive the Pauli-operator form of the one-body operator amenable to implementation as a Majorana select operation. The one-body operator is rewritten as

$$\begin{aligned}
& \sum_{p,q=1}^{N/2} h_{p\mathbf{k},q\mathbf{k}} a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma} \mapsto \frac{1}{4} \sum_{p,q=1}^{N/2} h_{p\mathbf{k},q\mathbf{k}} [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} + iY_{q\mathbf{k}\sigma})] \\
&= \frac{1}{8} \sum_{p,q=1}^{N/2} h_{p\mathbf{k},q\mathbf{k}} [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} + iY_{q\mathbf{k}\sigma})] + \frac{1}{8} \sum_{p,q=1}^{N/2} h_{q\mathbf{k},p\mathbf{k}} [\vec{Z}(X_{q\mathbf{k}\sigma} - iY_{q\mathbf{k}\sigma})][\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})] \\
&= \frac{1}{8} \sum_{p \neq q=1}^{N/2} h_{p\mathbf{k},q\mathbf{k}} [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} + iY_{q\mathbf{k}\sigma})] - \frac{1}{8} \sum_{p \neq q=1}^{N/2} h_{p\mathbf{k},q\mathbf{k}}^* [\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} - iY_{q\mathbf{k}\sigma})] \\
&+ \frac{1}{4} \sum_{p=1}^{N/2} h_{p\mathbf{k},p\mathbf{k}} [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})] \\
&= \frac{1}{8} \sum_{p \neq q=1}^{N/2} \text{Re}(h_{p\mathbf{k},q\mathbf{k}}) \{ [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} + iY_{q\mathbf{k}\sigma})] - [\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} - iY_{q\mathbf{k}\sigma})] \} \\
&+ \frac{i}{8} \sum_{p,q=1}^{N/2} \text{Im}(h_{p\mathbf{k},q\mathbf{k}}) \{ [\vec{Z}(X_{p\mathbf{k}\sigma} - iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} + iY_{q\mathbf{k}\sigma})] + [\vec{Z}(X_{p\mathbf{k}\sigma} + iY_{p\mathbf{k}\sigma})][\vec{Z}(X_{q\mathbf{k}\sigma} - iY_{q\mathbf{k}\sigma})] \}
\end{aligned}$$

$$\begin{aligned}
 & + \frac{1}{2} \sum_{p=1}^{N/2} h_{p\mathbf{k},p\mathbf{k}} (\mathbb{1}_{p\mathbf{k}\sigma} - Z_{p\mathbf{k}\sigma}) \\
 & = \frac{1}{4} \sum_{p \neq q=1}^{N/2} \text{Re}(h_{p\mathbf{k},q\mathbf{k}}) \{ -i\bar{Z}Y_{p\mathbf{k}\sigma} \bar{Z}X_{q\mathbf{k}\sigma} + i\bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} \} \\
 & + \frac{i}{4} \sum_{p,q=1}^{N/2} \text{Im}(h_{p\mathbf{k},q\mathbf{k}}) \{ \bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}X_{q\mathbf{k}\sigma} + \bar{Z}Y_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} \} + \frac{1}{2} \sum_{p=1}^{N/2} h_{pp}(\mathbf{k}) (\mathbb{1}_{p\mathbf{k}\sigma} - Z_{p\mathbf{k}\sigma}) \\
 & = \frac{i}{4} \sum_{p \neq q=1}^{N/2} \text{Re}(h_{p\mathbf{k},q\mathbf{k}}) \{ \bar{Z}X_{q\mathbf{k}\sigma} \bar{Z}Y_{p\mathbf{k}\sigma} + \bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} \} \\
 & + \frac{i}{4} \sum_{p,q=1}^{N/2} \text{Im}(h_{p\mathbf{k},q\mathbf{k}}) \{ \bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}X_{q\mathbf{k}\sigma} + \bar{Z}Y_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} \} + \frac{1}{2} \sum_{p=1}^{N/2} h_{pp}(\mathbf{k}) (\mathbb{1}_{p\mathbf{k}\sigma} - Z_{p\mathbf{k}\sigma}) \\
 & = \frac{i}{2} \sum_{p,q=1}^{N/2} \text{Re}(h_{p\mathbf{k},q\mathbf{k}}) \bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} + \frac{i}{4} \sum_{p,q=1}^{N/2} \text{Im}(h_{p\mathbf{k},q\mathbf{k}}) \{ \bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}X_{q\mathbf{k}\sigma} + \bar{Z}Y_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma} \} + \frac{1}{2} \sum_{p=1}^{N/2} h_{p\mathbf{k},p\mathbf{k}} \mathbb{1}. \quad (\text{A1})
 \end{aligned}$$

In the last line we have used the symmetry of  $\text{Re}(h_{p\mathbf{k},q\mathbf{k}})$  to combine  $\bar{Z}X_{q\mathbf{k}\sigma} \bar{Z}Y_{p\mathbf{k}\sigma}$  and  $\bar{Z}X_{p\mathbf{k}\sigma} \bar{Z}Y_{q\mathbf{k}\sigma}$ , then used the fact that  $iXY = -Z$  to combine the sum with  $p \neq q$  with that for  $p$ . The complete expression for the Hamiltonian has the sum over  $\sigma$  and  $\mathbf{k}$ , which we have left out for simplicity here. Including those gives the expression in Eq. (??).

## 2. One-body correction for sparse case

Next we derive the effective one-body term from the two-electron part of the Hamiltonian. In the case  $p = q$  and  $\mathbf{Q} = 0$ , the second term in square brackets in Eq. (21) can be written as

$$\begin{aligned}
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & = V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* (a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} \\
 & + a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & = V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger. \quad (\text{A2})
 \end{aligned}$$

In the last line we have used the fact that for  $p = q$  and  $\mathbf{Q} = 0$ ,  $a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}$  is just the identity, so this becomes a one-body operator.

Similarly, if  $r = s$  and  $\mathbf{Q} = 0$  (but  $p \neq q$ ), the second term in square brackets in Eq. (21) can be written as

$$\begin{aligned}
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & = V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} (a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & + a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger) \\
 & = V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} a_{r(\mathbf{k}' \oplus \mathbf{Q})\tau} a_{s\mathbf{k}'\tau}^\dagger \\
 & - V_{p\mathbf{k},q(\mathbf{k} \oplus \mathbf{Q}),r(\mathbf{k}' \oplus \mathbf{Q}),s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}. \quad (\text{A3})
 \end{aligned}$$

Thus we see that in either case ( $p = q$  or  $r = s$ ), we have the same expression as in Eq. (22), plus a one-body operator. Moreover, because of the symmetry of  $V$  (in swapping the  $pq$  pair with the  $rs$  pair), these corrections are equal. Note also that we can relabel swapping  $p$  with  $q$  and  $r$  with  $s$  to replace  $V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}$  with (now explicitly taking  $\mathbf{Q} = 0$ )

$$V_{q\mathbf{k},p\mathbf{k},s\mathbf{k}',r\mathbf{k}'}^* a_{q\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} = -V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',s\mathbf{k}'}^* a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \quad (\text{A4})$$

This means that the contribution of these corrections is

$$\sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p,q=1}^{N_k/2} \left( \sum_{r=1}^{N_k/2} \sum_{\mathbf{k}'} V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',r\mathbf{k}'} \right) a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \quad (\text{A5})$$

In this expression the constant factor is determined as follows. There is a factor of  $1/4$  in Eq. (21). Next, there is a

factor of 2 because we have the contribution from  $p = q$  as well as that from  $r = s$ . Last, there is the factor of 2 from the summation over the spin  $\tau$ . As a result, these factors cancel to give 1 above. Therefore, for  $p \neq q$ , we can combine this one-body term with  $h_{pq}$  as

$$h'_{pq} = h_{pq} + \sum_{r=1}^{N/2} \sum_{\mathbf{k}'}^{N_k} V_{p\mathbf{k},q\mathbf{k},r\mathbf{k}',r\mathbf{k}'}. \quad (\text{A6})$$

Next we consider the case where  $p = q$ ,  $r = s$ , and  $\mathbf{Q} = 0$ . Then the second term in square brackets in Eq. (21) can be written as

$$\begin{aligned} & V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}^\dagger a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\oplus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \\ &= V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}^\dagger a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\oplus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \\ &+ V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}^\dagger (a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\oplus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau} \\ &- a_{p\mathbf{k}\sigma}^\dagger a_{q(\mathbf{k}\oplus\mathbf{Q})\sigma} a_{r(\mathbf{k}'\oplus\mathbf{Q})\tau}^\dagger a_{s\mathbf{k}'\tau}). \end{aligned} \quad (\text{A7})$$

The operators in brackets in the final line can be written as, taking  $p = q$ ,  $r = s$ , and  $\mathbf{Q} = 0$ ,

$$\begin{aligned} & a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} + a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} \\ & - a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} - a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} \\ &= a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} - a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} \\ &= a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} + a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} - \mathbb{1}. \end{aligned} \quad (\text{A8})$$

By symmetry of swapping  $p$  and  $q$ , and swapping  $r$  and  $s$ , we must be able to simplify the final line of Eq. (A7) to

$$V_{p\mathbf{k},p\mathbf{k},r\mathbf{k}',r\mathbf{k}'}^\dagger (a_{r\mathbf{k}'\tau}^\dagger a_{r\mathbf{k}'\tau} + a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} - \mathbb{1}). \quad (\text{A9})$$

That is, this value of  $V$  is real. We can also relabel  $r$  and  $p$  and use symmetry to show the contribution from the first term in Eq. (A9) is equivalent to

$$V_{r\mathbf{k}',r\mathbf{k}',p\mathbf{k},p\mathbf{k}}^\dagger a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} = V_{p\mathbf{k},p\mathbf{k},r\mathbf{k}',r\mathbf{k}'}^\dagger a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma}. \quad (\text{A10})$$

Hence the contribution of these corrections is

$$\sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p=1}^{N/2} \left( \sum_{r=1}^{N/2} \sum_{\mathbf{k}'}^{N_k} V_{p\mathbf{k},p\mathbf{k},r\mathbf{k}',r\mathbf{k}'}^\dagger \right) (a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} - \mathbb{1}/2). \quad (\text{A11})$$

In this case, the constant factor comes from 1/4 in Eq. (21), and a factor of 2 from the sum over  $\tau$ . As a result the expression in Eq. (A9) is divided by 2 here, and apart from the identity we have the same expression as that accounting for only one of the pairs  $p, q$  and  $r, s$  being equal. The operator proportional to the identity can be ignored in the implementation of the Hamiltonian because it just gives a global shift in the eigenvalues.

### 3. Complexity for sparse implementation

The fundamental operator we are aiming to implement is in the form of Eq. (20) for the one-body term and Eq. (24) for the two-body term. In both we have a real part and an imaginary part; for the one-body term this is  $h_{pq}$ , and for the two-body term this is  $V_{p\mathbf{k},q(\mathbf{k}\oplus\mathbf{Q}),r(\mathbf{k}'\oplus\mathbf{Q}),s\mathbf{k}'}$ . We need to perform a state preparation that provides *real* amplitudes for the real and imaginary parts of  $h$  and  $V$  on separate basis states (not just real and imaginary parts of an amplitude on each basis state). This means the number of items of data to output is doubled in order to give the real and imaginary parts. The state preparation is otherwise essentially unchanged from that in Ref. [79], as described in Eq. (48) of that work and the accompanying explanation.

Recall that in the sparse state preparation procedure, we use a register indexing the nonzero entries [see Eq. (43) of Ref. [79]]. That is used to output “ind”, “alt,” and “keep” values via QROM [see Eq. (44) of Ref. [79]]. The “ind” values are values of  $p, q, r, s$ , as well as the sign needed, and a qubit distinguishing between the one- and two-body terms. The “alt” values are alternate values of these quantities, and “keep” governs the probability of swapping these registers for the state preparation via coherent alias sampling. Since we need a bit to flag whether the amplitude being produced is for the real or imaginary part, that would indicate we need two extra bits output, one for the “ind” value and one for the “alt” value. However, we can use one bit in the register indexing the nonzero entries to flag between real and imaginary parts. It is most convenient to make this register the least significant bit. Then we just need to produce “alt” values of this register, so the output size is only increased by 1 bit instead of 2. A requirement for this approach is that the nonzero entries of  $V$  that are retained are the same for the real and imaginary parts.

A further increase in the size of the output register is because we need to output values of  $\mathbf{k}, \mathbf{k}'$ , and  $\mathbf{Q}$ . The number of bits needed to store  $\mathbf{k}$  is not simply  $\lceil \log N_k \rceil$  because  $\mathbf{k}$  is a vector. The number of bits will be denoted  $n_k$ . If we assume that the number of values is given by the product of numbers in the three dimensions  $N_k = N_x N_y N_z$ , then

$$n_k = \lceil \log N_x \rceil + \lceil \log N_y \rceil + \lceil \log N_z \rceil. \quad (\text{A12})$$

Therefore,  $\mathbf{k}, \mathbf{k}'$ , and  $\mathbf{Q}$  increase the size of both the ind and alt registers by  $3n_k$ , for a total of  $6n_k$ . The size of the output is given in Eq. (A13) of Ref. [33] as  $m = \aleph + 8\lceil \log(N/2) \rceil + 4$ , and would here be increased to

$$\aleph + 8\lceil \log(N/2) \rceil + 6n_k + 5, \quad (\text{A13})$$

where we have also increased the size of the output by 1 to account for selecting between real and imaginary parts, as discussed above. The quantity  $\aleph$  is the number of bits for the “keep” register.



The remaining consideration for the sparse state preparation is the symmetry. In prior work there were three symmetries, with a swap of  $p, q$  with  $r, s$  as well as swaps within the  $p, q$  and  $r, s$  pairs. The method to take advantage of this was described from Eq. (49) in Ref. [79]. There you perform only the preparation for a restricted range of  $p, q, r, s$ , then use three qubits to control swaps to generate the symmetries.

Here we have the symmetry with a swap of  $p, q$  with  $r, s$ , but we can only swap the  $p, q$  and  $r, s$  pairs simultaneously. We also need to take the complex conjugate when performing that swap. In order to implement the symmetries here, we will have two control qubits. One qubit will be in a  $|+\rangle$  state and control swap of the  $p, q$  with  $r, s$  as before, except we now have the registers containing  $\mathbf{k}, \mathbf{k}', \mathbf{k} \ominus \mathbf{Q}, \mathbf{k}' \ominus \mathbf{Q}$  to swap. That qubit is only set to  $|+\rangle$  for the two-body term, since that symmetry does not make sense for the one-body term. The second qubit is used to simultaneously swap the  $p, q$  and  $r, s$  pairs, as well as the registers containing  $\mathbf{k}$ , etc. It will also be used as a control for a  $Z$  phase gate on a qubit flagging imaginary components. That is a Clifford gate and is not included in the Toffoli count.

The net result is that the cost of the swaps to produce these symmetries is unchanged from that in Ref. [79], except in that we are counting the qubits needed to store  $\mathbf{k}$ , etc, as well as  $p, q, r, s$ . Since a controlled swap of two qubits can be performed with a single Toffoli (and Clifford gates), the Toffoli cost of the two controlled swaps of registers is the total number of qubits used to store  $p, q, r, s$  as well as  $\mathbf{k}, \mathbf{k}', \mathbf{k} \ominus \mathbf{Q}, \mathbf{k}' \ominus \mathbf{Q}$ , which is  $4\lceil \log(N/2) \rceil + 4n_k$ . Note that in the state preparation we will be producing  $\mathbf{k}, \mathbf{k}', \mathbf{Q}$ , and need to compute  $\mathbf{k} \ominus \mathbf{Q}$  and  $\mathbf{k}' \ominus \mathbf{Q}$  before performing the swaps for these symmetries.

Assuming for the moment that  $N_x, N_y, N_z$  are all powers of 2, then the number of Toffolis needed for the modular subtractions of the three components will be  $n_k - 3$ , unless one or more of  $N_x, N_y, N_z$  is equal to 1. It is simpler to give the cost as  $n_k$  Toffolis, to avoid needing to address special cases. A further complication is when one or more of  $N_x, N_y, N_z$  are not powers of 2. In this case, the subtraction can be performed in the usual way for two's complement binary. Then you can check if the result for any component is negative, and if it is then add the appropriate  $N_x, N_y, N_z$  to make it non-negative. The controlled addition of a classically given number has complexity  $n_k$ , so this at worst doubles the complexity to  $2n_k$  for the modular subtraction.

The other major feature that we need to account for is the modified SELECT operation needed. The basic circuit primitive was given in Fig. 13 of Ref. [33], in order to apply  $\vec{Z}Y_{p,\sigma}$  followed by  $\vec{Z}X_{q,\sigma}$ . A more complicated circuit primitive was given in Fig. 1 of Ref. [79], which included testing  $p = q$ , which is not needed in the approach of Ref. [33]. Here the scheme is more complicated, because instead of having a fixed sequence where we need to apply  $Y$  followed by  $X$  we have every combination. This can be

achieved by simply performing each twice; once with a controlled  $\vec{Z}Y$  and once with a controlled  $\vec{Z}X$ , with a doubling of the Toffoli complexity. That can be seen easily from the diagram in Fig. 9 of Ref. [2]. There a control qubit is used, so that can be used to control application of this circuit with  $Y$ , then to control application of this circuit with  $X$ .

To understand how  $X$  versus  $Y$  is selected, note that there are effectively five bits controlling here. Let us call the bit selecting between the one- and two-body terms  $b_0$ ; this is created in the sparse state preparation. Let us call the bit selecting real versus imaginary parts  $b_1$ ; this is again created in the state preparation. There also needs to be a bit  $b_2$  for selecting between the two lines for real and the two lines for imaginary in the expression in Eq. (24). Then we have  $b_3$  to select between the two terms in the first set of square brackets in each line of Eq. (24), and a bit  $b_4$  selecting between the two terms in the second set of square brackets.

Now, considering the operators indexed by  $r, s$  first, these are applied for the two-body terms but not the one-body term. This control of the operations adds only one Toffoli to the cost. For the first operation,  $\vec{Z}X_{s\mathbf{k}'\tau}$  or  $\vec{Z}Y_{s\mathbf{k}'\tau}$ , we can see that the selection between  $X$  and  $Y$  depends only on bit  $b_4$ . For the second operation, the selection is independent of whether we have the real or imaginary part. We select  $X$  if we have  $b_4 = 0$  (the first term) and  $b_2 = 0$  (the first line), or if we have  $b_4 = b_2 = 1$ . To create a bit selecting between  $X$  and  $Y$  we can simply perform a CNOT between these bits, with no Toffoli cost.

Next, consider the operators indexed by  $p, q$ . For simplicity we will first consider just the two-body terms. Again the first operation can select between  $X$  and  $Y$  just by using the bit  $b_3$  selecting between the terms. Then for the second operation, we select  $X$  if we have  $b_1, b_2, b_3$  equal to 0, 0, 0, or 0, 1, 1, or 1, 0, 1, or 1, 1, 0. It is easily seen that if we apply CNOTs with  $b_1$  then  $b_2$  as control and  $b_3$  as target, then we should apply  $X$  if we have  $b_3 = 0$ . This selection can be performed without Toffolis again.

Now to take account of how the one-body terms are applied, it is convenient to rewrite the first line of Eq. (20) as

$$-\frac{i}{4} \sum_{p,q=1}^{N/2} \text{Re}(h_{p\mathbf{k},q\mathbf{k}}) \{ \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma} - \vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma} \}. \quad (\text{A14})$$

Then the selection between the operations is identical to that for  $b_2 = 1$  (second lines) for the two-body part. Therefore, for the above analysis of the two-body implementation, we can replace  $b_2$  with a bit that is 1 if  $b_2 = 1$  OR  $b_0 = 0$ . This operation requires one more Toffoli.

Another modification we need to make is to compute  $\mathbf{k}' \ominus \mathbf{Q}$  and  $\mathbf{k} \ominus \mathbf{Q}$  to use in the selection for the two-body

operations. As explained above, these modular subtractions have complexity at worst  $2n_k$ . The calculation  $\mathbf{k} \ominus \mathbf{Q}$  needs to be controlled on the bit  $b_0$  selecting between the one- and two-body terms, which increases its complexity by  $n_k$ . Therefore, the complexity of this arithmetic is  $3\lceil \log N_k \rceil$  Toffolis. We can keep the working qubits in order to uncompute this arithmetic with Clifford gates.

Finally, we should account for the phase factors needed in the implementation. The phase factors needed are as follows.

1. We should apply an  $i$  phase factor on the one-body term. That can be implemented with an  $S$  gate, which is Clifford.
2. If we have the one-body term (flagged by  $b_0 = 0$ ) we should flip the sign of the real part (flagged by  $b_1 = 0$ ). This can be done with a controlled phase, which is again Clifford.
3. For the two-body term ( $b_0 = 1$ ), real ( $b_1 = 0$ ), and second line ( $b_2 = 1$ ) we should flip the sign. This doubly controlled phase has a cost of one Toffoli.
4. We should flip the sign with  $b_3 = 1$  if we have the two-body term ( $b_0 = 1$ ) and the second line for real ( $b_1 = 0, b_2 = 1$ ) or the first line for imaginary ( $b_1 = 1, b_2 = 0$ ). We should also flip the sign with  $b_3 = 1$  if we have the one-body term ( $b_0 = 0$ ) and real ( $b_1 = 0$ ). To achieve this we can first perform a CNOT with  $b_1$  as the control and  $b_2$  as the target. Then, if  $b_0 = 0, b_1 = 0$  OR  $b_0 = 1, b_2 = 1$  we should apply a  $Z$  gate to the qubit containing  $b_3$ . This can be achieved with two double controlled phase gates, so has Toffoli cost 2.
5. We should flip the sign for  $b_4 = 1$  if we have the second line  $b_2 = 1$ . That is just a controlled phase with no non-Clifford cost.

As a result, the total complexity of implementing these phase factors is three Toffoli gates.

The total additional complexity is therefore two Toffolis for the selection of  $X$  versus  $Y$  when we account for needing to perform the one-or two-body term,  $3\lceil \log N_k \rceil$  Toffolis for subtractions, three Toffolis for phase factors, and doubling the selection cost to select between  $X$  and  $Y$ . The two Toffolis to account for the one-body term were one for selecting performing the operators indexed by  $r, s$ , and another Toffoli to perform an OR between  $b_0$  and  $b_2$  for the operators indexed by  $p, q$ .

A further complication arises where the  $h$  and  $V$  are dependent on the spins  $\sigma$  and  $\tau$ . This is easily accounted for by outputting the values of  $\sigma, \tau$  as part of the state preparation. This means that the size of both the “ind” and “alt” outputs are increased by 2, making the total size of the output increase by 4 to be

$$\aleph + 8\lceil \log(N/2) \rceil + 6n_k + 9. \quad (\text{A15})$$

Often there is the symmetry that for  $V$  the value with  $\sigma = \uparrow, \tau = \downarrow$  are the same as for  $\sigma = \downarrow, \tau = \uparrow$ . This means that we can omit the case  $\sigma = \downarrow, \tau = \uparrow$ , and use a swap of these two qubits controlled by an ancilla qubit in the usual way for obtaining symmetries. In the detailed costing below, we give results for the case where  $h$  and  $V$  are not dependent on spin for simplicity.

The QROM output size is

$$m = \aleph + 8n_N + 6n_k + 5, \quad (\text{A16})$$

where  $n_N = \lceil \log(N/2) \rceil$ . This output size is increased above that analyzed in Ref. [33]. Then, using that output size, the formula for the cost of the preparation with  $d$  unique nonzero entries is

$$\lceil d/k_1 \rceil + m(k_1 - 1) \quad (\text{A17})$$

and of the inverse preparation is

$$\lceil d/k_2 \rceil + k_2. \quad (\text{A18})$$

Here  $k_1$  and  $k_2$  must be chosen as powers of 2. This formula is the same as in Ref. [33], but with the modified value of  $m$ .

To begin the state preparation, we need to prepare an equal superposition state over  $d$  basis states. The analysis is described in Ref. [33], which gives the costing  $3\lceil \log d \rceil - 3\eta + 2b_r - 9$  Toffoli gates. Here  $\eta$  is a number such that  $2^\eta$  is a factor of  $d$ , and  $b_r$  is a number of bits used for rotation of an ancilla qubit to improve the amplitude of success. This is a cost needed both for the preparation and inverse preparation.

Other minor Toffoli costs are as follows. We use extra ancillas to save cost, because a large number of ancillas were used for the QROM, and can be reused here without increasing the maximum number of ancillas needed. In the following, we use the notation  $n_N = \lceil \log(N/2) \rceil$ .

1. Perform SELECT as shown in Fig. 13 of Ref. [33] twice, but controlling between  $X$  and  $Y$ . This complexity is  $4NN_k - 6$ , since we have 8 times a complexity of  $NN_k/2 - 1$  for each of the selected operations, plus two Toffolis to generate the qubits we need for the control. There were two Toffolis needed to account for selecting between one- and two-body terms, and otherwise the selection to account for the various terms can be performed using Clifford gates. In addition to this, we need to perform swaps controlled by spin qubits twice for each of the two spin qubits, with a complexity  $2NN_k$ . That then gives a total complexity of this step  $6NN_k - 6$ .
2. The state preparation needs an inequality test on  $\aleph$  qubits, as well as controlled swaps. The controlled swaps are on  $n_N + 3n_k + 2$  qubits. Here

- $4\lceil\log(N/2)\rceil$  are for the values of  $p$ ,  $q$ ,  $r$ , and  $s$ , the  $3n_k$  is for the  $\mathbf{k}$ ,  $\mathbf{k}'$ , and  $\mathbf{Q}$  values, a  $+1$  is for the qubit that distinguishes between the one- and two-electron terms, and a further  $+1$  comes from the qubit for selecting between the real and imaginary parts. There are also ind and alt values of the sign, but the correct phase can be applied with Clifford gates, so this does not add to the Toffoli cost. The cost of the inequality test on  $\aleph$  qubits is  $\aleph$ . As in Ref. [33], we can eliminate the non-Clifford cost of the inverse preparation using ancillas and measurements, so the Toffoli cost is  $\aleph + 4n_N + 3n_k + 2$ .
3. The controlled swaps used to generate the symmetries have a cost of  $4n_N + 4n_k$ . This is increased by  $4n_k$  over that in Ref. [33], since we need to swap  $\mathbf{k}$  registers as well. Although there are only two controlled swaps rather than three in Ref. [33], two of the controlled swaps in Ref. [33] together act on as many qubits as one controlled swap here, so the factor of 4 is the same as in Ref. [33]. A further  $4n_k$  cost is for computing  $\mathbf{k} \ominus \mathbf{Q}$  and  $\mathbf{k}' \ominus \mathbf{Q}$  (or  $2n_k$  if  $N_x, N_y, N_z$  are powers of 2), and an extra  $n_k$  is needed to make the computation of  $\mathbf{k} \ominus \mathbf{Q}$  controlled. Again these controlled swaps can be inverted for the inverse preparation with measurements and Clifford gates. Thus the total Toffoli cost here is  $4n_N + 9n_k$ .
  4. For the qubitization construction a reflection on the ancilla is needed as well. The qubits that need to be reflected on are as follows:
    - (a) the  $\lceil\log d\rceil$  qubits for preparing the state;
    - (b)  $\aleph$  qubits for the equal superposition state in coherent alias sampling;
    - (c) two qubits that are used for controlled swaps to generate the symmetries of the state;
    - (d) the two spin qubits;
    - (e) the ancilla qubit that is rotated to produce the equal superposition state;
    - (f) and the qubits storing  $b_2, b_3, b_4$ , which are also used in the linear combination of unitaries.

There is no non-Clifford Toffoli cost for the preparation on  $b_2, b_3, b_4$ , since an equal superposition may be prepared with a Hadamard. They are control qubits that need to be reflected upon for the qubitization, so add a cost of three Toffolis to the reflection giving a total cost  $\lceil\log d\rceil + \aleph + 6$ .

5. As before, the control for the phase estimation uses unary iteration on the control registers, with one more Toffoli for each step. The control by these registers is implemented simply by controlling the reflection, which needs just one Toffoli per step.

6. An extra three Toffolis are needed for the phase factors.

Adding all these minor costs together gives, in the spin-independent case

$$\begin{aligned}
 & 2(3\lceil\log d\rceil - 3\eta + 2b_r - 9) + (6NN_k - 6) + (\aleph + 4n_N \\
 & \quad + 3n_k + 2) + 4n_N + 9n_k + \lceil\log d\rceil + \aleph + 6 + 2 + 3 \\
 & = 6NN_k + 8n_N + 10\lceil\log N_k\rceil + 2\aleph + 7\lceil\log d\rceil \\
 & \quad - 6\eta + 4b_r - 8. \tag{A19}
 \end{aligned}$$

The total cost for a single step is then

$$\begin{aligned}
 & \left\lceil \frac{d}{k_1} \right\rceil + m(k_1 - 1) + \left\lceil \frac{d}{k_2} \right\rceil + k_2 + 6NN_k + 8n_N \\
 & \quad + 12n_k + 2\aleph + 7\lceil\log d\rceil - 6\eta + 4b_r - 8, \tag{A20}
 \end{aligned}$$

with  $m = \aleph + 8n_N + 6\lceil\log N_k\rceil + 5$ ,  $n_N = \lceil\log(N/2)\rceil$ ,  $\eta$  an integer such that  $2^\eta$  is a factor of  $d$ , and  $b_r$  the number of bits used for rotation of an ancilla qubit.

We may count the qubit costs by considering the maximum used during the QROM, as the advanced QROM has a high qubit usage that will not be exceeded in other parts of the algorithm. The qubit costs are therefore as follows.

1. The control register for the phase estimation uses  $\lceil\log(\mathcal{I} + 1)\rceil$  qubits, and there are  $\lceil\log(\mathcal{I} + 1)\rceil - 1$  qubits for the unary iteration.
2. The system uses  $NN_k$  qubits.
3. The  $\lceil\log d\rceil + \aleph + 8$  qubits that need to be reflected upon listed above.
4. A qubit is needed to flag success of the equal superposition state preparation.
5. The phase gradient state uses  $b_r$  qubits.
6. The QROM uses qubits (including the output)  $mk_1 + \lceil\log(d/k_1)\rceil$ .

This gives a total number of logical qubits

$$\begin{aligned}
 & 2\lceil\log(\mathcal{I} + 1)\rceil + NN_k \\
 & \quad + \lceil\log d\rceil + b_r + \aleph + mk_1 + \lceil\log(d/k_1)\rceil + 8, \tag{A21}
 \end{aligned}$$

with  $m = \aleph + 8n_N + 6\lceil\log N_k\rceil + 5$ .

## APPENDIX B: SINGLE-FACTORIZATION DERIVATIONS

### 1. One-body correction for single factorization

For the single factorized form of the Hamiltonian, we may use the same expressions for  $\hat{A}$  and  $\hat{B}$  for the case  $\mathbf{Q} = 0$  as for  $\mathbf{Q} \neq 0$ , with an additional correction proportional to the identity. This yields a one-body correction in the case of  $\hat{A}$  but not  $\hat{B}$ . For  $\hat{A}_n(\mathbf{Q} = 0)$  we obtain a term proportional to the identity, as follows:

$$\begin{aligned}
\hat{A}_n(\mathbf{Q} = 0) &= \frac{1}{2} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p \neq q} \left( L_{p\mathbf{k}, q\mathbf{k}, n} a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma} + L_{p\mathbf{k}, q\mathbf{k}, n}^* a_{q\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} \right) + \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_p L_{p\mathbf{k}p\mathbf{k}, n} a_{p\mathbf{k}\sigma}^\dagger a_{p\mathbf{k}\sigma} \quad (\text{B1}) \\
&= \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{p \neq q} \left( \frac{i\text{Re}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right. \\
&\quad \left. + \frac{i\text{Im}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right) + \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_p \frac{L_{p\mathbf{k}p\mathbf{k}, n}}{2} (\mathbb{1} - Z) \\
&= \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_{pq} \left( \frac{i\text{Re}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right. \\
&\quad \left. + \frac{i\text{Im}[L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}]}{4} (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q(\mathbf{k} \oplus \mathbf{Q})\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q(\mathbf{k} \oplus \mathbf{Q})\sigma}) \right) + \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}} \sum_p \frac{L_{p\mathbf{k}p\mathbf{k}, n}}{2} \mathbb{1}. \quad (\text{B2})
\end{aligned}$$

Here we have used the symmetry of  $L$ , so  $L_{p\mathbf{k}p\mathbf{k}, n}$  is real. This derivation is similar to that for the one-body term in Appendix A 1.

Because  $\hat{A}_n(\mathbf{Q} = 0)$  is squared, the identity term gives rise to a one-body correction

$$\begin{aligned}
&\frac{i}{4} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_n^M \sum_{\mathbf{k}}^{N_k} \sum_{p, q}^{N/2} (\text{Re}[L_{p\mathbf{k}q\mathbf{k}, n}] (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma}) \text{Im}[L_{p\mathbf{k}q\mathbf{k}, n}] (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma})) \sum_{\mathbf{k}'}^{N_k} \sum_{r=1}^{N/2} L_{r\mathbf{k}'r\mathbf{k}', n} \\
&= \frac{i}{4} \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{\mathbf{k}}^{N_k} \sum_{p, q}^{N/2} \sum_{\mathbf{k}'}^{N_k} \sum_{r=1}^{N/2} (\text{Re}[V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'}] (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma} - \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma}) \text{Im}[V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'}] \\
&\quad + (\vec{Z}X_{p\mathbf{k}\sigma} \vec{Z}X_{q\mathbf{k}\sigma} + \vec{Z}Y_{p\mathbf{k}\sigma} \vec{Z}Y_{q\mathbf{k}\sigma})). \quad (\text{B3})
\end{aligned}$$

Here there was a factor of 1/2 on the square of  $\hat{A}_n(\mathbf{Q} = 0)$ , a factor of 2 from the cross term in the square, a factor of 2 from the sum over the spin on the identity, and so a factor of 1/2 has been canceled. The form of this correction is identical to that for the one-body term, except  $h_{p\mathbf{k}, q\mathbf{k}}$  is replaced with

$$\sum_{r=1}^{N/2} \sum_{\mathbf{k}'}^{N_k} V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'}. \quad (\text{B4})$$

For  $\hat{B}_n(\mathbf{Q} = 0)$ , it is easily seen that the symmetry  $L_{p\mathbf{k}q\mathbf{k}, n} = L_{q\mathbf{k}p\mathbf{k}, n}^*$  implies that  $\hat{B}_n(\mathbf{Q} = 0) = 0$ . If we use the form for  $\hat{B}_n(\mathbf{Q} = 0)$  in terms of Pauli operators given in Eq. (38), then it will be proportional to the identity due to the case  $p = q$ . Squaring then just gives a correction proportional to the identity (which can be ignored in the implementation because it is just an energy shift), and it gives no one-body correction. As a result we add the expression in Eq. (B4) to  $h_{pq}$  to obtain the complete one-body Hamiltonian given in Eq. (39).

## 2. Complexity for single-factorized representation

To see the changes we need to make to the algorithm for the single-factorized representation, recall that the two-body term was of the form [33]

$$W' = \frac{1}{8} \sum_{\ell=1}^L \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{p, q=1}^{N/2} W_{pq}^{(\ell)} Q_{pq\sigma} \right)^2, \quad (\text{B5})$$

where  $Q_{pq\sigma}$  was an individual Pauli string. So the changes in the representation are as follows:

- The sum over  $\ell$  up to  $L$  has been replaced with a sum over  $\mathbf{Q}$  and  $n$ , as well as a sum over the squares of  $A$  and  $B$ .
- Inside the square, the sum over just  $\sigma, p, q$  now also has a sum over  $\mathbf{k}$ .
- Inside the sum, instead of just having a single Pauli string, we have a sum over 4, with real and imaginary parts of  $L_{p\mathbf{k}q(\mathbf{k} \oplus \mathbf{Q}), n}$ .

The amendments we will make to the original algorithm (according to the description in Ref. [33]) to implement the block encoding are as follows.

- (a) For the sum over  $\mathbf{Q}$  and  $n$  we can combine them into  $\ell$ , and use the same state-preparation method as before. The value of  $\mathbf{Q}$  will need to be used in the SELECT operation, so needs to be output as part of that state preparation.
- (b) In the preparation for the block encoding of  $A$  and  $B$ , the index  $\mathbf{k}$  will be needed as well as  $p$  and  $q$ .
- (c) We no longer take advantage of  $p, q$  symmetry.
- (d) We need to perform arithmetic to compute  $\mathbf{k} \ominus \mathbf{Q}$  and  $\mathbf{k}' \ominus \mathbf{Q}$ , with a cost of  $4n_k$  (or  $2n_k$  if  $N_x, N_y, N_z$  are powers of 2).
- (e) A number of qubits can be used for selecting between the parts of the linear combination of unitaries, similar to the sparse case. We have  $b_0$  to select between the one- and two-body terms,  $b_1$  for selecting between the real and imaginary parts, and  $b_3$  selecting between the two terms in one application of  $A$  or  $B$ . The qubit  $b_2$  can be used for selecting between  $A$  and  $B$ , which is a change from the sparse case, where it was used for selecting between lines.

We do not need  $b_4$  because we are implementing  $A$  or  $B$  twice (and creating the bit  $b_3$  both times).

- (f) There needs to be a doubling of the selection cost to select between  $X$  and  $Y$  as in the sparse case.
- (g) The creation of the qubits for controlling between  $X$  and  $Y$  can be performed with one additional Toffoli. Note first that the terms in  $A$  are equivalent to the one-body part, and the terms in  $B$  are the same except with the real and imaginary lines swapped around. This means that we can use  $b_0$  and  $b_2$  as a control to flip  $b_1$ , which effectively swaps the real and imaginary parts for  $B$  so it can be implemented in the same way. Now, for the first selection of  $X$  versus  $Y$ , we can apply a CNOT with  $b_1$  as control and  $b_3$  as target, and use that as control. For the second selection we can simply use  $b_3$  as control.
- (h) For the phase factors, we just need a sign flip if  $b_1 = 0$  and  $b_3 = 1$ , which is a Clifford controlled phase.

To explain the modifications needed for the costings, here we give the sequence of steps with the same numbering as in Ref. [33], explaining the differences.

1. We first prepare a state as

$$\frac{1}{\sqrt{\lambda}} \left( |0, 0, 0, 0\rangle \sqrt{\sum_{p,q} (|\operatorname{Re}(h'_{pq})| + |\operatorname{Im}(h'_{pq})|)} + \frac{1}{\sqrt{2}} \sum_{\mathbf{Q}, n} |\ell, \mathbf{Q}, n, 1\rangle \sum_{\mathbf{k}, pq} (|\operatorname{Re}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]| + |\operatorname{Im}[L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n}]|) \right), \quad (\text{B6})$$

where  $|\ell, \mathbf{Q}, n\rangle$  indicates  $\ell$ , which starts from 1 indexing values of  $\mathbf{Q}, n$ , but  $\mathbf{Q}$  and  $n$  are also output in registers. That is, we will be preparing  $\ell$  while outputting values of  $\mathbf{Q}, n$ . We are assuming the more difficult case where the number of values of  $\mathbf{Q}$  or  $n$  are not powers of 2, but if they are then further simplifications are possible. This has complexity as follows.

- (a) Preparing an equal superposition on  $MN_k + 1$  basis states has complexity  $3n_{MN} + 2b_r - 9$ , where  $b_r$  is the number of bits used for the rotation on the ancilla,

$$n_{MN} = \lceil \log(MN_k + 1) \rceil. \quad (\text{B7})$$

- (b) A QROM is applied with output size

$$b_{MN} = \aleph_1 + n_{MN} + 2n_k + 2, \quad (\text{B8})$$

with  $\aleph_1$  being the number of bits used for the keep values (which govern the precision of the state preparation via the inequality test). Here  $n_{MN}$  and  $2n_k$  are for  $\ell$  and  $\mathbf{Q}$ , with the factor of 2 accounting for ind and alt values of  $\mathbf{Q}$ . The extra two qubits are for outputting a qubit showing if  $\ell = 0$  (for selecting between the one- and two-body parts). The complexity is

$$\left\lceil \frac{MN_k + 1}{k_{MN}} \right\rceil + b_{MN}(k_{MN} - 1). \quad (\text{B9})$$

- (c) An inequality test is performed with complexity  $\aleph_1$ .
- (d) A controlled swap is performed with complexity  $n_k + \lceil \log M \rceil + 1$ .

2. Next, we prepare a state on the second register as



$$\begin{aligned}
& \frac{1}{\sqrt{\lambda}} \left( |0, 0, 0, 0\rangle \sum_{p,q} \left[ \sqrt{2|\operatorname{Re}(h'_{pq})|} |\theta_{pq0}^{(0)}\rangle |0, p, q, 0\rangle + \sqrt{2|\operatorname{Im}(h'_{pq})|} |\theta_{pq1}^{(0)}\rangle |0, p, q, 1\rangle \right] \right. \\
& + \frac{1}{\sqrt{2}} \sum_{\mathbf{Q}, n} |\ell, \mathbf{Q}, n, 1\rangle \sqrt{\sum_{\mathbf{k}, r, s} (|\operatorname{Re}[L_{r\mathbf{k}s(\mathbf{k} \ominus \mathbf{Q}), n}]| + |\operatorname{Im}[L_{r\mathbf{k}s(\mathbf{k} \ominus \mathbf{Q}), n}]|)} \\
& \left. \times \sum_{\mathbf{k}, p, q} \left[ \sqrt{|\operatorname{Re}(L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n})|} |\theta_{\mathbf{k}pq0}^{(\ell)}\rangle |\mathbf{k}, p, q, 0\rangle + \sqrt{|\operatorname{Im}(L_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q}), n})|} |\theta_{\mathbf{k}pq1}^{(\ell)}\rangle |\mathbf{k}, p, q, 1\rangle \right] \right) |+\rangle |+\rangle, \quad (\text{B10})
\end{aligned}$$

where  $\theta_{\mathbf{k}pq0}^{(\ell)}$ ,  $\theta_{\mathbf{k}pq1}^{(\ell)}$  are used to obtain the correct signs on the terms, and the  $|+\rangle$  states at the end are used to select the spin and control the swap between the  $p$  and  $q$  registers.

Now we have a distinction from Ref. [33] in that we have separate real and imaginary parts, and a separate prepared qubit to flag between the real and imaginary parts. Because of the large number of variables, we will again use a single variable for iteration, and use it to output  $\mathbf{k}, p, q$ . The complexity of this state preparation is then as follows.

- (a) First, prepare an equal superposition over the variable for iteration. There are  $P = N_k N^2 / 2$  values to take, which includes a factor of 2 for the real and imaginary parts,  $N_k$  for  $\mathbf{k}$ , and  $N^2 / 4$  for the values of  $p, q$ . Then the complexity of preparing the equal superposition is  $3n_p - 3\eta + 2b_r - 9$ , where  $n_p = \lceil \log P \rceil$ , with  $\eta$  being the largest number such that  $2^\eta$  is a factor of  $P$ .
- (b) The size of the QROM output is

$$b_p = 2n_k + 4n_N + \aleph_2 + 3, \quad (\text{B11})$$

where the first term is for the three components of  $\mathbf{k}$ , the second is for  $p$  and  $q$ . The third is for ind and alt values of the qubit to store the correct sign, as well as an alt value of the extra qubit for selecting between the real and imaginary parts. We do not include an ind value for that qubit, because it is part of the register we are iterating over. The complexity of this QROM will be

$$\left\lceil \frac{MN_k + 1}{k_{p1}} \right\rceil \left\lceil \frac{P}{k_{p2}} \right\rceil + b_p (k_{p1} k_{p2} - 1), \quad (\text{B12})$$

where we are accounting for the cost to select based on both the index from the factorization and the index for  $\mathbf{k}, p, q$ , and using the result for the complexity of QROM on two registers from Appendix G of Ref. [33].

- (c) Perform the inequality test with cost  $\aleph_2$ , which is the bits of precision for this state preparation.
- (d) Perform the controlled swap with the alt values with cost  $n_k + 2n_N + 1$ . Here we are swapping the ind and alt values of  $\mathbf{k}, p, q$ , as well as the qubit selecting between real and imaginary parts. The sign required for the sign qubits can be implemented with Cliffords as in Ref. [33], so does not add to this Toffoli cost.

3. We no longer perform swaps of  $p$  and  $q$  for symmetry, but we do need to perform arithmetic to compute  $\mathbf{k} \ominus \mathbf{Q}$  and  $\mathbf{k}' \ominus \mathbf{Q}$ , with a cost of  $4n_k$ .
4. Perform SELECT by performing the sequence of four controlled  $\bar{Z}X_{p,\sigma}$  or  $\bar{Z}Y_{p,\sigma}$  operations. The cost is  $4(NN_k/2 - 1)$  Toffolis since it must be controlled, and there is a cost of one more Toffoli to create the qubits to control on. In order to select the spin we also perform a swap controlled by the spin selection qubit before and after, with a cost of  $NN_k$  Toffolis.
5. Reverse steps 2 and 3, where the complexities are the same except the QROM complexity, which is changed to

$$\left\lceil \frac{MN_k + 1}{k'_{p1}} \right\rceil \left\lceil \frac{P}{k'_{p2}} \right\rceil + k'_{p1} k'_{p2}. \quad (\text{B13})$$

6. Reflect on the qubits that were prepared in step 2. The qubits we need to reflect on are as follows.
  - (a) The  $n_p$  qubits for the variable of iteration.
  - (b) We need to reflect on the  $\aleph_2$  registers that are used for the equal superposition state for the state preparation.
  - (c) One that is rotated for the preparation of the equal superposition state.
  - (d) One for the spin.
  - (e) One for controlling the swap between the  $p$  and  $q$  registers.
  - (f) One for selecting between the real and imaginary part.
  - (g) One for selecting between  $A$  and  $B$ .

That gives a total of  $n_P + \aleph_2 + 5$  qubits. The reflection needs to be controlled on the success of the preparation on the  $\ell$  register, and  $\ell \neq 0$ , making the total cost  $n_P + \aleph_2 + 5$  Toffolis.

7. Perform steps 2 to 5 again, but this time  $MN_k + 1$  is replaced with  $MN_k$  in Eq. (B12) and Eq. (B13). Also, the SELECT operation needs to be controlled on  $\ell \neq 0$ , which flags the one-body term. That requires another four Toffolis.
8. Invert the state preparation on the  $\ell$  register, where the complexity of the QROM is reduced to

$$\left\lceil \frac{MN_k + 1}{k'_p} \right\rceil + k'_p. \quad (\text{B14})$$

9. To complete the step of the quantum walk, perform a reflection on the ancillas used for the state preparation. There are  $n_{MN} + n_P + \aleph_1 + \aleph_2 + 5$ , where the qubits we need to reflect on are as follows.

- (a) The  $n_{MN}$  qubits for the  $\ell$  register.
- (b) The  $n_P$  qubits for the registers in the state preparation for  $A$  and  $B$ .
- (c) The  $\aleph_1$  qubits for the equal superposition state used for preparing the state on the  $\ell$  register using the coherent alias sampling.
- (d) The  $\aleph_2$  qubits for the equal superposition state for preparing the state for  $A$  and  $B$ .
- (e) Two qubits rotated for the boosting the success probability for the equal superposition states.
- (f) One qubit for the spin.
- (g) One qubit for controlling the swap of the  $p$  and  $q$  registers.
- (h) One for selecting between the real and imaginary part.
- (i) One for selecting between  $A$  and  $B$ .

This reflection has cost  $n_{MN} + n_P + \aleph_1 + \aleph_2 + 4$ .

10. The steps of the walk are made controlled by using unary iteration on an ancilla used for the phase estimation. Each step requires another two Toffolis for the unary iteration and making the reflection controlled.

In this list of steps we have not explicitly included the part for applying the phase factors, but that has no non-Clifford cost.

Next we consider the total number of logical qubits needed for the simulation via this method.

1. The control register for the phase estimation, and the ancillas for the unary iteration, together need  $2\lceil \log \mathcal{I} \rceil - 1$  qubits.
2. There are  $NN_k$  qubits for the target system.
3. There are  $n_{MN} + 2$  qubits for the  $\ell$  register, the qubit rotated in preparing the equal superposition, and

the qubit flagging success of preparing the equal superposition.

4. The state preparation on the  $\ell$  register uses  $b_{MN} = 2n_k + 2\lceil \log M \rceil + 2\aleph_1 + 2$  qubits. Here  $2n_k + 2\lceil \log M \rceil$  is for the ind and alt values of  $\mathbf{Q}$  and  $n$ ,  $\aleph_1$  are for keep values,  $\aleph_1$  are for the equal superposition state, 1 is for the output of the inequality test, and 2 are for the qubit flagging  $\ell \neq 0$  and its alternate value.
5. There are  $n_P + 2$  qubits needed for the register preparing  $p, q, \mathbf{k}$  values, a qubit that is rotated for the equal superposition, and a qubit flagging success of preparing the equal superposition.
6. The equal superposition state used for the second preparation uses  $\aleph_2$  qubits.
7. The phase gradient register uses  $b_r$  qubits.
8. The qubits for the spin, controlling the swap of  $p$  and  $q$ , selection between the real and imaginary parts, and selection between  $A$  and  $B$  for a total of 4.
9. The QROM needs a number of qubits  $b_p k_{p1} k_{p2} + \lceil \log[(MN_k + 1)/k_{p1}] \rceil + \lceil \log[L/k_{p2}] \rceil$ .

The QROM for the state preparation on the second register uses a large number of temporary ancillas, which can be reused by later parts of the algorithm, so those later parts of the algorithm do not need the number of qubits counted. The total number of qubits used is then

$$2\lceil \log \mathcal{I} \rceil + NN_k + n_{MN} + n_P + 2n_k + 2\lceil \log M \rceil + 2\aleph_1 + \aleph_2 + b_r + 9 + b_p k_{p1} k_{p2} + \lceil \log[(MN_k + 1)/k_{p1}] \rceil + \lceil \log[L/k_{p2}] \rceil \quad (\text{B15})$$

with  $b_p = 2n_k + 2n_N + \aleph_2 + 3$ ,  $n_N = \lceil \log(N/2) \rceil$ ,  $n_P = \lceil \log P \rceil$ ,  $L = N_k N(N + 2)/4$ . This completes the costing of the low-rank factorization method.

## APPENDIX C: DOUBLE-FACTORIZATION DERIVATIONS

### 1. One-body correction

Here we derive the correction for the one-body Hamiltonian as given in Eq. (52). The  $\lambda$  value for the Hamiltonian can be calculated by determining the total L1 norm using the second factorization

$$\hat{H}'_2 = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n^{N_k} \sum_n^M (\hat{A}_n^2(\mathbf{Q}) + \hat{B}_n^2(\mathbf{Q})) \quad (\text{C1})$$

with

$$\begin{aligned} 2\hat{A}_n(\mathbf{Q}) &= \sum_{\mathbf{k}} \left[ U_n^A(\mathbf{Q}, \mathbf{k}) \left( \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},A}} f_p^A(\mathbf{Q}, n, \mathbf{k})(\mathbb{1} - Z_{p\mathbf{k}\sigma}) \right) U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \right] \\ &= \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger - \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \end{aligned} \quad (\text{C2})$$

where  $\hat{\mathbb{1}}_{\mathbf{k}}^A = \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},A}} f_p^A(\mathbf{Q}, n, \mathbf{k}) \mathbb{1}$  and  $\hat{Z}_{\mathbf{k}}^A = \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},A}} f_p^A(\mathbf{Q}, n, \mathbf{k}) Z_{p\mathbf{k}\sigma}$ , and

$$\begin{aligned} 2\hat{B}_n(\mathbf{Q}) &= \sum_{\mathbf{k}} \left[ U_n^B(\mathbf{Q}, \mathbf{k}) \left( \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},B}} f_p^B(\mathbf{Q}, n, \mathbf{k})(\mathbb{1} - Z_{p\mathbf{k}\sigma}) \right) U_n^B(\mathbf{Q}, \mathbf{k})^\dagger \right] \\ &= \sum_{\mathbf{k}} U_n^B(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^B U_n^B(\mathbf{Q}, \mathbf{k})^\dagger - \sum_{\mathbf{k}} U_n^B(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^B U_n^B(\mathbf{Q}, \mathbf{k})^\dagger \end{aligned} \quad (\text{C3})$$

where  $\hat{\mathbb{1}}_{\mathbf{k}}^B = \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},B}} f_p^B(\mathbf{Q}, n, \mathbf{k}) \mathbb{1}$  and  $\hat{Z}_{\mathbf{k}}^B = \sum_{\sigma} \sum_p^{\Xi_{\mathbf{Q},n,\mathbf{k},B}} f_p^B(\mathbf{Q}, n, \mathbf{k}) Z_{p\mathbf{k}\sigma}$ . The factor of 1/2 from the Jordan-Wigner transform is squared to 1/4, which is moved outside each term and combined with the prefactor 1/2 to produce a prefactor of 1/8. We note that  $\hat{A}_n(\mathbf{Q})^2$  can be written as

$$\begin{aligned} 4\hat{A}_n(\mathbf{Q})^2 &= \left( \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger - \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \right) \\ &\quad \times \left( \sum_{\mathbf{k}'} U_n^A(\mathbf{Q}, \mathbf{k}') \hat{\mathbb{1}}_{\mathbf{k}'}^A U_n^A(\mathbf{Q}, \mathbf{k}')^\dagger - \sum_{\mathbf{k}'} U_n^A(\mathbf{Q}, \mathbf{k}') \hat{Z}_{\mathbf{k}'}^A U_n^A(\mathbf{Q}, \mathbf{k}')^\dagger \right) \\ &= 2 \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \hat{A}_n(\mathbf{Q}) + 2\hat{A}_n(\mathbf{Q}) \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \\ &\quad + \sum_{\mathbf{k}, \mathbf{k}'} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger U_n^A(\mathbf{Q}, \mathbf{k}') \hat{Z}_{\mathbf{k}'}^A U_n^A(\mathbf{Q}, \mathbf{k}')^\dagger \\ &\quad - \sum_{\mathbf{k}} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{\mathbb{1}}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \sum_{\mathbf{k}'} U_n^A(\mathbf{Q}, \mathbf{k}') \hat{\mathbb{1}}_{\mathbf{k}'}^A U_n^A(\mathbf{Q}, \mathbf{k}')^\dagger. \end{aligned} \quad (\text{C4})$$

The last term in the above equation is proportional to the identity and is ignored. A similar expression can be derived for  $\hat{B}_n(\mathbf{Q})^2$  and thus the component of the two-body term involving two Pauli  $Z$  operators is written as

$$\begin{aligned} V &= \frac{1}{8} \sum_{\mathbf{Q}, n, \mathbf{k}, \mathbf{k}'} U_n^A(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^A U_n^A(\mathbf{Q}, \mathbf{k})^\dagger U_n^A(\mathbf{Q}, \mathbf{k}') \hat{Z}_{\mathbf{k}'}^A U_n^A(\mathbf{Q}, \mathbf{k}')^\dagger \\ &\quad + \frac{1}{8} \sum_{\mathbf{Q}, n, \mathbf{k}, \mathbf{k}'} U_n^B(\mathbf{Q}, \mathbf{k}) \hat{Z}_{\mathbf{k}}^B U_n^B(\mathbf{Q}, \mathbf{k})^\dagger U_n^B(\mathbf{Q}, \mathbf{k}') \hat{Z}_{\mathbf{k}'}^B U_n^B(\mathbf{Q}, \mathbf{k}')^\dagger, \end{aligned} \quad (\text{C5})$$

which implies the two-body L1 norm,  $\lambda_{\text{DF},2}$ , is

$$\lambda_{\text{DF},2} = \frac{1}{4} \sum_{\mathbf{Q}, n} \left[ \left( \sum_{\mathbf{k}, p}^{N_k \Xi_{\mathbf{Q},n,\mathbf{k},A}} |f_n^A(p, \mathbf{Q}, \mathbf{k})| \right)^2 + \left( \sum_{\mathbf{k}, p}^{N_k \Xi_{\mathbf{Q},n,\mathbf{k},B}} |f_n^B(p, \mathbf{Q}, \mathbf{k})| \right)^2 \right], \quad (\text{C6})$$

where the factor of 1/8 becomes a factor of 1/2 accounting for spin. This factor of 1/2 is further divided by two because we perform oblivious amplitude amplification—i.e., the inner step of qubitization evolving by  $2\hat{A}_n(\mathbf{Q})^2 - \mathbb{1}$  and  $2\hat{B}_n(\mathbf{Q})^2 - \mathbb{1}$ .

Next, the one-body terms in the third line of Eq. (C4) can be rewritten as

$$2 \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A \hat{A}_n(\mathbf{Q}) + 2 \hat{A}_n(\mathbf{Q}) \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A. \quad (\text{C7})$$

This expression needs to be divided by 8 to give the contribution to the Hamiltonian, and there is a similar contribution from  $\hat{B}_n(\mathbf{Q})^2$  to give the overall contribution to the one-body Hamiltonian

$$\frac{1}{2} \sum_{n, \mathbf{Q}} \left( \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A \hat{A}_n(\mathbf{Q}) + \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^B \hat{B}_n(\mathbf{Q}) \right). \quad (\text{C8})$$

Taking the trace of Eqs. (C2) and (C3) then implies

$$\sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A = \mathbb{1} \text{Tr}(\hat{A}_n(\mathbf{Q})) = \mathbb{1} \frac{1}{2} [\text{Tr}(\hat{\rho}_n(\mathbf{Q})) + \text{Tr}(\hat{\rho}_n^\dagger(\mathbf{Q}))], \quad (\text{C9})$$

$$\sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^B = \mathbb{1} \text{Tr}(\hat{B}_n(\mathbf{Q})) = \mathbb{1} \frac{i}{2} [\text{Tr}(\hat{\rho}_n(\mathbf{Q})) - \text{Tr}(\hat{\rho}_n^\dagger(\mathbf{Q}))]. \quad (\text{C10})$$

The trace of  $\hat{\rho}_n(\mathbf{Q})$  is nonzero only for  $\mathbf{Q} = 0$ . In that case

$$\text{Tr}(\hat{\rho}_n(0)) = 2 \sum_{\mathbf{k}} \left( \sum_r^{N/2} L_{r\mathbf{k}r\mathbf{k},n} \right), \quad (\text{C11})$$

which is real. Moreover, it is easily seen that  $\hat{\rho}(0)$  is Hermitian using the symmetry  $L_{p\mathbf{k}q\mathbf{k},n} = L_{q\mathbf{k}p\mathbf{k},n}^*$ , so  $\hat{A}_n(0) = \hat{\rho}_n(0)$  and  $\hat{B}_n(0) = 0$ . Therefore,

$$\sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A \hat{A}_n(0) + \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^B \hat{B}_n(0) = 2 \sum_{\mathbf{k}, p, q, \sigma} \left( \sum_{\mathbf{k}', r} L_{p\mathbf{k}q\mathbf{k},n} L_{r\mathbf{k}'r\mathbf{k}',n} \right) a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \quad (\text{C12})$$

Therefore, the contribution to the one-body Hamiltonian becomes

$$\begin{aligned} & \frac{1}{2} \sum_{n, \mathbf{Q}} \left( \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^A \hat{A}_n(\mathbf{Q}) + \sum_{\mathbf{k}} \hat{\mathbb{1}}_{\mathbf{k}}^B \hat{B}_n(\mathbf{Q}) \right) \\ &= \sum_{\mathbf{k}, p, q, \sigma} \left( \sum_{\mathbf{k}', r} V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'} \right) a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \end{aligned} \quad (\text{C13})$$

As a result, the complete one-body Hamiltonian is

$$H'_1 = \sum_{\mathbf{k}, p, q, \sigma} \left( h_{p\mathbf{k}, q\mathbf{k}} + \sum_{\mathbf{k}', r} V_{p\mathbf{k}, q\mathbf{k}, r\mathbf{k}', r\mathbf{k}'} \right) a_{p\mathbf{k}\sigma}^\dagger a_{q\mathbf{k}\sigma}. \quad (\text{C14})$$

This is identical to the result that was obtained in the single-factorization case as in Eq. (B4). Thus the L1 norm of  $H'_1$  is the sum

$$\lambda_{\text{DF},1} = \sum_{\mathbf{k}} \sum_p |\lambda_{\mathbf{k},p}|, \quad (\text{C15})$$

where  $\lambda_{\mathbf{k},p}$  is an eigenvalue of the matrix representing  $H'_1(\mathbf{k})$ , which are the coefficients in the parenthesis of Eq. (C14).

## 2. Complexity of the double-factorized representation

Our form of the two-body part of the Hamiltonian is

$$\hat{H}'_2 = \frac{1}{2} \sum_{\mathbf{Q}} \sum_n^{N_k} \left( \hat{A}_n^2(\mathbf{Q}) + \hat{B}_n^2(\mathbf{Q}) \right), \quad (\text{C16})$$

with

$$\hat{A}_n(\mathbf{Q}) = \sum_{\mathbf{k}} \left[ U_n^A(\mathbf{Q}, \mathbf{k}) \left( \sum_{\sigma} \sum_r^{\Xi_{\mathbf{Q},n,\mathbf{k},A}} f_r^A(\mathbf{Q}, n, \mathbf{k}) n_{r,\mathbf{k},\sigma} \right) U_n^A(\mathbf{Q}, \mathbf{k})^\dagger \right] \quad (\text{C17})$$

and similarly for  $\hat{B}_n(\mathbf{Q})$ . In comparison, the double-factorized Hamiltonian from Refs. [33,45] is

$$F' = \frac{1}{8} \sum_{\ell=1}^L U_\ell \left( \sum_{\sigma \in \{\uparrow, \downarrow\}} \sum_{p=1}^{\Xi^{(\ell)}} f_p^{(\ell)} Z_{p,\sigma} \right)^2 U_\ell^\dagger. \quad (\text{C18})$$

So, in contrast to the decomposition before, instead of a sum over  $\ell$ , we have a sum over  $\mathbf{Q}, n$ , and a qubit indexing over  $\hat{A}, \hat{B}$ . This difference can be accounted for easily in the method as presented in Ref. [33]. That method may be summarized as follows.

1. Perform a state preparation over  $\ell$  for the first factorisation.
2. Use a QROM on  $\ell$  to output some parameters needed for the state preparation for the second factorization (the operator that is squared).
3. Perform the inner state preparation over  $p$ .
4. Apply a QROM to output the sequence of rotations dependent on  $\ell$  and  $p$ .
5. Apply the Givens rotations.
6. Apply a controlled Z.

7. Invert the Givens rotations, QROM, and state preparation over  $p$ .
8. Perform a reflection on the ancilla qubits used for the state preparation over  $p$ .
9. Perform steps 3 to 7 again.
10. Invert the QROM from step 2.
11. Invert the state preparation from step 1.

Note that this is distinct from the procedure in Ref. [45], which combined the  $\ell$  and  $p$  preparations.

To account for the changes here, the index  $\ell$  can be used to iterate through all possible values of  $\mathbf{Q}$ ,  $n$ , and the qubit indexing over  $\hat{A}$ ,  $\hat{B}$ . Most of the steps can be performed ignoring these values, but we will need to know  $\mathbf{Q}$  before performing the Givens rotations. It is convenient to output this value in the QROM used in step 2, which slightly increases the output size of this QROM. We will also need to output  $\mathbf{k}$  values, and these will be given in the second state preparation used in step 3. But, that preparation will produce a joint index of  $p$  and  $\mathbf{k}$  without giving  $\mathbf{k}$  explicitly (similar to our preparation over  $\ell$  not giving  $\mathbf{Q}$  explicitly). This can be output by the QROM in step 4.

In order to apply the Givens rotations, we will need to perform controlled swaps of system registers  $\mathbf{k}$ ,  $\mathbf{k} \ominus \mathbf{Q}$  into working registers, then apply the Givens rotations on those working registers. Since  $\mathbf{k} \ominus \mathbf{Q}$  is not given directly by the state preparation, it needs to be computed with cost  $2n_k$  (or  $n_k$  if  $N_x, N_y, N_z$  are powers of 2). The controlled swaps have a Toffoli cost of  $2n_k$  for the unary iteration, and  $NN_k$  for the controlled swaps. The cost of  $NN_k$  is because we need to run through  $NN_k/2$  system qubits twice. These controlled swaps are performed 4 times, because they need to be performed before and after each application of the Givens rotations. That gives a total cost from this part

$$4NN_k + 12n_k. \quad (\text{C19})$$

Then for the QROM outputting the Givens rotations, the number of items of data can be given as

$$\sum_{\mathbf{Q}, n, \mathbf{k}} (\Xi_{\mathbf{Q}, n, \mathbf{k}, A} + \Xi_{\mathbf{Q}, n, \mathbf{k}, B}), \quad (\text{C20})$$

where  $\Xi_{\mathbf{Q}, n, \mathbf{k}, A}$  and  $\Xi_{\mathbf{Q}, n, \mathbf{k}, B}$  are the cutoffs in the sums for  $\hat{A}$  and  $\hat{B}$ . As per Eq. (48), we define  $\Xi$  as this quantity divided by  $L = 2N_k M$ , so we can write the number of items of data as  $L\Xi$ .

The Givens rotations need to be on  $2N$  orbitals, so there are  $2N$  Givens rotations. For each of these rotations two angles need to be specified, in contrast to one in Refs. [33, 45]. The size of the data output for the QROM for the Givens rotations is increased to  $2N\lceil \rceil$ , because there are two registers of size  $N/2$ , and there are two rotations of  $\lceil \rceil$  of precision for each Givens rotation. The total complexity of applying the Givens rotations is increased to

$16N(\lceil \rceil - 2)$ . This is an increase of a factor of 4 over that in Ref. [33], with a factor of 2 from using two working registers, and a factor of 2 because there are two rotations for each Givens rotation.

The other changes in the cost are relatively trivial. There is a swap on the system registers controlled on the spin register. Since this is now on  $NN_k$  qubits instead of  $N$ , the cost is multiplied by  $N_k$ .

So, to summarize the complexity using the same numbering of steps as in Ref. [33], we have the following.

1. The cost of the state preparation over  $\ell$  is

$$(3n_L - 3\eta + 2b_r - 9) + \left\lceil \frac{L+1}{k_{p1}} \right\rceil + b_{p1}(k_{p1} - 1) + \aleph_1 + n_L, \quad (\text{C21})$$

where  $L$  is now  $2N_k M$  and as before  $b_{p1} = n_L + \aleph_1$ ,  $n_L = \lceil \log L \rceil$ .

2. The complexity of the QROM on  $\ell$  is now

$$\left\lceil \frac{L+1}{k_o} \right\rceil + b_o(k_o - 1), \quad (\text{C22})$$

with

$$b_o = n_k + n_{\Xi} + n_{L, \Xi} + b_r + 1, \quad (\text{C23})$$

with the extra  $n_k$  being to output  $\mathbf{Q}$ . Here  $n_{\Xi}$  is the number of bits needed for  $\Xi_{\mathbf{k}, p}$  values of  $p$ , and  $n_{L, \Xi}$

$$n_{L, \Xi} = \lceil \log(L\Xi + N_k N/2) \rceil \quad (\text{C24})$$

is the number of bits needed for the offset.

3. The cost of the second stage of state preparation is

$$4(7n_{\Xi} + 2b_r - 6) + 4(n_{L, \Xi} - 1) + \left( \left\lceil \frac{L\Xi + NN_k/2}{k_{p2}} \right\rceil + \left\lceil \frac{L\Xi}{k_{p2}} \right\rceil + 2b_{p2}(k_{p2} - 1) \right) + 4(\aleph_2 + n_{\Xi}), \quad (\text{C25})$$

where the brackets are used to indicate the cost of parts (a) to (d) of step 3. As well as using our modified definition of  $\Xi$ , the only change over the costing in Ref. [33] is replacing  $N/2$  with  $NN_k/2$  for the range of values for the one-body term. In this cost we are including the second use of the preparation in part 7.



4. The cost of the number operators via QROM is

$$\begin{aligned} & \left\lceil \frac{L\Xi + NN_k/2}{k_r} \right\rceil + \left\lceil \frac{L\Xi}{k_r} \right\rceil + (4N\lceil \rceil + n_k)(k_r - 1) \\ & + \left\lceil \frac{L\Xi + NN_k/2}{k'_r} \right\rceil + \left\lceil \frac{L\Xi}{k'_r} \right\rceil \\ & + 2k'_r + 4(n_{L,\Xi} - 1) + 16N(\lceil - 2) + 2NN_k + 2. \end{aligned} \quad (\text{C26})$$

Here the term  $4N\lceil(k_r - 1)$  has been increased by a factor of 4 over that in Ref. [33], because we have 2 times as many qubits that the Givens rotations need to act on, and there are twice as many rotations needed for each Givens rotation. (This term is corresponding to the output size for the QROM.) We have also added  $n_k$  for the output size so we can output the value of  $\mathbf{k}$  needed to select the register. Again  $N/2$  is replaced with  $NN_k/2$  for the one-body term. The quantity  $16N(\lceil - 2)$  is for the cost of the Givens rotations, and is also multiplied by a factor of 4 over that in Ref. [33]. The  $2NN_k$  for the controlled swaps for spin, and is increased over  $2N$  in Ref. [33] because we now have  $\mathbf{k}$ .

5. The inversion of the state preparation has cost

$$\begin{aligned} & 2(7n_\Xi + 2b_r - 6) + 2(n_{L,\Xi} - 1) \\ & + \left( \left\lceil \frac{L\Xi + NN_k/2}{k'_{p2}} \right\rceil + \left\lceil \frac{L\Xi}{k'_{p2}} \right\rceil + 2k'_{p2} \right) \\ & + 2(\aleph_2 + n_\Xi). \end{aligned} \quad (\text{C27})$$

This cost is the same as in part 3, except the cost of erasing the QROM is reduced. We are again including both uses (with the second described in step 7).

6. The reflection for the oblivious amplitude amplification has an unchanged cost

$$n_\Xi + \aleph_2 + 2. \quad (\text{C28})$$

7. The cost of the second use of the block encoding to give the square are already accounted for above.

8. The cost of inverting step 1 is

$$(3n_L - 3\eta + 2b_r - 9) + \left\lceil \frac{L+1}{k'_{p1}} \right\rceil + k'_{p1} + \aleph_1 + n_L, \quad (\text{C29})$$

and for inverting step 2 is

$$\left\lceil \frac{L+1}{k'_o} \right\rceil + k'_o, \quad (\text{C30})$$

where we are using the improved cost for erasing QROM.

9. The reflection cost is unchanged at

$$n_L + n_\Xi + \aleph_1 + \aleph_2 + 1. \quad (\text{C31})$$

10. The extra cost of unary iteration on the control register and of controlling the reflection on that register is two Toffolis.

11. The new costs of performing controlled swaps into working registers and arithmetic to compute  $\mathbf{k} \ominus \mathbf{Q}$  and  $\mathbf{k}' \ominus \mathbf{Q}$  are

$$4NN_k + 12n_k. \quad (\text{C32})$$

Adding all these costs together gives the total cost for block encoding the Hamiltonian.

The cost in terms of logical qubits is very similar to that for the original double-factorized approach. The differences are as follows.

1. There are registers needed to store  $\mathbf{k}, \mathbf{Q}, \mathbf{k} \ominus \mathbf{Q}$ . Because  $\mathbf{k} \ominus \mathbf{Q}$  can be computed in place in the  $\mathbf{Q}$  register, we need only storage for 2. Moreover, because  $\mathbf{k}$  is given in the QROM output in part 4 above, it does not need to be added to that qubit costing.
2. There are  $N$  qubits used for the working registers (2 of size  $N/2$ ).
3. A number of parameters are changed, in particular, the number of system qubits is now  $NN_k$ , and  $L$  is computed from the number of values of  $\mathbf{Q}$  and  $n$ .
4. The size of the output for the Givens rotations is multiplied by a factor of 4.

## APPENDIX D: TENSOR HYPERCONTRACTION DERIVATIONS

### 1. THC symmetries

In this section we derive the symmetry relationships for the central tensor based on the fourfold symmetry of the two-electron integral tensor as used in Eq. (63). Recall an element of the two-electron integral tensor can be represented in THC form as

$$\begin{aligned} V_{p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q}),r\mathbf{k}'\ominus\mathbf{Q},s\mathbf{k}'} &= \sum_{\mu,\nu} \chi_{p\mathbf{k},\mu}^* \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\mu} \zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} \\ & \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\nu} \chi_{s\mathbf{k}',\nu}, \end{aligned} \quad (\text{D1})$$

where  $\mathbf{G}_1$  is shorthand for  $\mathbf{G}_{\mathbf{k},\mathbf{k}\ominus\mathbf{Q}}$  and  $\mathbf{G}_2$  is shorthand for  $\mathbf{G}_{\mathbf{k}',\mathbf{k}'\ominus\mathbf{Q}}$ . The fourfold symmetry of the complex valued two-electron integral tensor is reflected in the central tensor  $\zeta$ . We recover the symmetry by first noting the four equivalent two-electron integrals

$$\begin{aligned}
V_{p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q}),r\mathbf{k}'-\mathbf{Q},s\mathbf{k}'} &= \sum_{\mu,\nu} \chi_{p\mathbf{k},\mu}^* \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\mu} \zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\nu} \chi_{s\mathbf{k}',\nu}, \\
V_{q(\mathbf{k}\ominus\mathbf{Q}),p\mathbf{k},s\mathbf{k}',r\mathbf{k}'-\mathbf{Q}}^* &= \left( \sum_{\mu,\nu} \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\mu}^* \chi_{p\mathbf{k},\mu} \zeta_{\mu\nu}^{(\ominus\mathbf{Q}),!\mathbf{G}_1,!\mathbf{G}_2} \chi_{s\mathbf{k}',\nu}^* \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\nu} \right)^*, \\
V_{r(\mathbf{k}'\ominus\mathbf{Q}),s\mathbf{k}',p\mathbf{k},q(\mathbf{k}\ominus\mathbf{Q})} &= \sum_{\mu,\nu} \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\mu}^* \chi_{s\mathbf{k}',\mu} \zeta_{\mu\nu}^{(\ominus\mathbf{Q}),!\mathbf{G}_2,!\mathbf{G}_1} \chi_{p\mathbf{k},\nu}^* \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\nu}, \\
V_{s\mathbf{k}',r(\mathbf{k}'\ominus\mathbf{Q}),q(\mathbf{k}\ominus\mathbf{Q}),p\mathbf{k}} &= \left( \sum_{\mu,\nu} \chi_{s\mathbf{k}',\mu}^* \chi_{r(\mathbf{k}'\ominus\mathbf{Q}),\mu} \zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_2,\mathbf{G}_1} \chi_{q(\mathbf{k}\ominus\mathbf{Q}),\nu}^* \chi_{p\mathbf{k},\nu} \right)^*,
\end{aligned}$$

which implies

$$\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} = (\zeta_{\mu\nu}^{(\ominus\mathbf{Q}),!\mathbf{G}_1,!\mathbf{G}_2})^* = \zeta_{\nu\mu}^{(\ominus\mathbf{Q}),!\mathbf{G}_2,!\mathbf{G}_1} = (\zeta_{\nu\mu}^{\mathbf{Q},\mathbf{G}_2,\mathbf{G}_1})^*. \quad (\text{D2})$$

Here  $(\ominus\mathbf{Q})$  is used to indicate a modular negative of  $\mathbf{Q}$ , similar to modular subtraction. In the above expression the complement of  $\mathbf{G}_1$ ,  $!\mathbf{G}_1$ , is defined through

$$\begin{aligned}
\mathbf{k}_p - \mathbf{k}_q &= \mathbf{Q} + \mathbf{G}_1, \\
\mathbf{k}_q - \mathbf{k}_p &= (\ominus\mathbf{Q}) + !\mathbf{G}_1, \\
!\mathbf{G}_1 &= -(\mathbf{Q} + \mathbf{G}_1 + (\ominus\mathbf{Q})), \quad (\text{D3})
\end{aligned}$$

and it is important to note that  $(\ominus\mathbf{Q})$  is defined to be in the original set of  $k$  points and it is useful as we build only  $\zeta^{\mathbf{Q}}$ . A similar expression can be derived for  $!\mathbf{G}_2$ . It is helpful to consider some concrete examples, which are given in Table VII.

## 2. Complexity of the tensor hypercontraction representation

The following is a detailed costing for the qubitization oracles using the THC LCU. In the initial state preparation, we need to prepare a superposition over  $\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2, \mu, \nu$  with weights  $\sqrt{|\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2}|}$ . The state can be prepared via the coherent alias sampling procedure, starting with QROM to output keep and alt values. One option here is to produce an equal superposition over  $\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2, \mu, \nu$ , then calculate a contiguous register from these values to use for the QROM. That procedure is fairly complicated, because it requires preparing equal superpositions over three components of  $\mathbf{Q}$  as well as  $\mathbf{G}_1, \mathbf{G}_2, \mu$  and  $\nu$ , then arithmetic for the contiguous register. To simplify the procedure we give the complexity for giving ind values like for sparse state preparation. That is, we prepare the contiguous register, and use the QROM to output both ind (index) and alt values of  $\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2, \mu, \nu$ .

There is the symmetry  $\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} = (\zeta_{\nu,\mu}^{\mathbf{Q},\mathbf{G}_2,\mathbf{G}_1})^*$ , which indicates only half the range of  $\mu, \nu, \mathbf{G}_1, \mathbf{G}_2$  values need

be prepared. It is convenient to prepare the full range, but use part of the range for real and part for imaginary components. If we only were considering  $\mu, \nu$ , we could use  $\mu \leq \nu$  for real components and  $\mu > \nu$  for imaginary components. To account for  $\mathbf{G}_1, \mathbf{G}_2$  as well, we can combine them with  $\mu, \nu$  as least-significant bits for combined integers to use in inequality tests. This inequality test between  $\mu, \mathbf{G}_1$  and  $\nu, \mathbf{G}_2$  is used to give a qubit flagging that the component should be imaginary. A further qubit in a  $|+\rangle$  state is used to control a swap of  $\mu, \mathbf{G}_1$  with  $\nu, \mathbf{G}_2$  registers, and a controlled  $Z$  gate on the qubit flagging the imaginary component gives the desired complex conjugate. As a result the range for  $\mu, \nu$  is  $M^2$  taking account of giving real and imaginary components.

For  $\mathbf{Q}$  there is also the symmetry where  $\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} = (\zeta_{\mu\nu}^{(\ominus\mathbf{Q}),!\mathbf{G}_1,!\mathbf{G}_2})^*$ , so it is only necessary to produce approximately half as many values of  $\mathbf{Q}$ . This is complicated by the cases where  $\mathbf{Q} = \ominus\mathbf{Q}$ . If  $N_x, N_y, N_z$  are odd, then the only case where this can be true is that  $\mathbf{Q} = 0$ , so the number of values of  $\mathbf{Q}$  that need be considered is  $(N_x N_y N_z + 1)/2$ . If one of  $N_x, N_y, N_z$  is even and the other two are odd, then for the one that is even there will be a second value of that component of  $\mathbf{Q}$  that is equal to its negative. That means there are two value of  $\mathbf{Q}$  overall satisfying  $\mathbf{Q} = \ominus\mathbf{Q}$ , and the number of unique values is  $N_x N_y N_z / 2 + 1$ . Similarly, if there are two even values of  $N_x, N_y, N_z$ , then there are four values of  $\mathbf{Q}$  satisfying  $\mathbf{Q} = \ominus\mathbf{Q}$ , and the number of unique values is  $N_x N_y N_z / 2 + 2$ . For all three of  $N_x, N_y, N_z$  even the number of unique values is  $N_x N_y N_z / 2 + 4$ . We also need  $NN_k/2$  values for the one-body term. The number of values is then

$$d = 32[N_x N_y N_z + 2^v]M^2 + NN_k/2, \quad (\text{D4})$$

where  $v$  is the number of even values of  $N_x, N_y, N_z$ .

The size of the output is then

$$m = 2(2\lceil \log M \rceil + n_k + 8) + \aleph. \quad (\text{D5})$$

where  $\aleph$  is the number of bits for the keep register. There is a factor of 2 at the front to account for ind and alt values,

TABLE VII. Some examples of the values that the different momentum labels can take in Eq. (D3). We restrict  $\mathbf{k}, \mathbf{Q}, (\ominus\mathbf{Q})$  to be in the original  $k$ -point set.

$k$ mesh	$\mathbf{k}_p$	$\mathbf{k}_q$	$\mathbf{k}_p - \mathbf{k}_q$	$\mathbf{Q}$	$\mathbf{G}$	$\mathbf{k}_q - \mathbf{k}_p$	$(\ominus\mathbf{Q})$	$!\mathbf{G}$
[1, 1, 4]	(0, 0, 3)	(0, 0, 1)	(0, 0, 2)	(0, 0, 2)	(0,0,0)	(0, 0, -2)	(0, 0, 2)	(0, 0, -4)
[1, 4, 4]	(0, 2, 1)	(0, 3, 1)	(0, -1, 0)	(0, 3, 0)	(0, -4, 0)	(0, 1, 0)	(0, 1, 0)	(0, 0, 0)
[1, 4, 4]	(0, 2, 1)	(0, 3, 3)	(0, -1, -2)	(0, 3, 2)	(0, -4, -4)	(0, 1, 2)	(0, 1, 2)	(0, 0, 0)
[1, 4, 4]	(0, 1, 2)	(0, 1, 3)	(0, 0, -1)	(0, 0, 3)	(0, 0, -4)	(0, 0, 1)	(0, 0, 2)	(0, 0, 0)
[1, 4, 4]	(0, 1, 3)	(0, 1, 2)	(0, 0, 1)	(0, 0, 1)	(0,0,0)	(0, 0, -1)	(0, 0, 3)	(0, 0, -4)
[4, 4, 4]	(2, 1, 3)	(3, 1, 2)	(-1, 0, 1)	(3, 0, 1)	(-4, 0, 0)	(1, 0, -1)	(0, 0, 3)	(0, 0, -4)
[4, 4, 4]	(2, 1, 2)	(3, 3, 3)	(-1, -2, -1)	(3, 2, 3)	(-4, -4, -4)	(1, 2, 1)	(1, 2, 1)	(0, 0, 0)

then  $\lceil \log M \rceil$  for each of  $\mu$  and  $\nu$ , and  $n_k$  for the components of  $\mathbf{Q}$ . There is a further qubit distinguishing between the one- and two-electron terms, a qubit giving the sign of the real or imaginary component of  $\zeta$ , and six qubits for  $\mathbf{G}_1, \mathbf{G}_2$ , for a total +8.

1. There is a cost of  $N_k N / 2$  for controlled swaps for the spin. In principle, this is performed 4 times, because it is performed before and after the two  $c^\dagger c$  operators. The middle pair can be combined, with the single controlled swap being controlled by the parity of the two spin qubits, for a total cost of  $3N_k N / 2$ .
2. Before the state preparation, we need to prepare an equal superposition over  $d$  basis states, with costing  $3\lceil \log d \rceil - 3\eta + 2b_r - 9$  Toffoli gates. As before,  $\eta$  is a number such that  $2^\eta$  is a factor of  $d$ , and  $b_r$  is a number of bits used for rotation of an ancilla qubit to improve the amplitude of success. This cost is incurred twice, once for the preparation and once for the inverse preparation.
3. The complexity of the QROM being used for the state preparation is

$$\left\lceil \frac{d}{k_p} \right\rceil + m(k_p - 1), \quad (\text{D6})$$

with  $k_p$  being a power of 2. The inverse preparation then has a cost

$$\left\lceil \frac{d}{k'_p} \right\rceil + k'_p. \quad (\text{D7})$$

4. We perform an inequality test with cost  $\aleph$ . Accounting for the inverse of the preparation gives a total cost  $2\aleph$ .
5. The controlled swap based on the result of the inequality test is on

$$2\lceil \log M \rceil + n_k + 7 \quad (\text{D8})$$

pairs of qubits, so has this Toffoli cost. Note that we have +7 here rather than +8. This is because we do not need to swap the sign qubits; the sign

can be applied with  $Z$  gates controlled on the result of the inequality test, not adding to the Toffoli cost (as usual). This cost is incurred again in the inverse preparation for a total of  $4\lceil \log M \rceil + 2n_k + 14$ .

6. As described above, we perform an inequality test between  $\mu, \mathbf{G}_1$  and  $\nu, \mathbf{G}_2$  to give the qubit flagging whether we have a real or imaginary component. Then we perform a controlled swap of  $\mu, \mathbf{G}_1$  with  $\nu, \mathbf{G}_2$  to generate one symmetry for the state preparation, with the complex conjugate applied using a Clifford gate. This part therefore has Toffoli cost  $2\lceil \log M \rceil + 6$ . This cost is incurred again in the inverse preparation giving a total cost  $4\lceil \log M \rceil + 12$ . In addition to this controlled swap, we perform a controlled swap in the middle, but it is not controlled so does not add to the Toffoli complexity.
7. For the symmetry where  $\zeta_{\mu\nu}^{\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2} = (\zeta_{\mu\nu}^{(\ominus\mathbf{Q}), !\mathbf{G}_1, !\mathbf{G}_2})^*$ , we can use a second control qubit to flip the sign on  $\mathbf{Q}$ , negate  $\mathbf{G}_1$  and  $\mathbf{G}_2$ , and apply the complex conjugate. The complex conjugate again can be applied with a Clifford gate, and so can the controlled-NOT gates on  $\mathbf{G}_1, \mathbf{G}_2$ . A controlled sign flip of  $\mathbf{Q}$  can be performed with  $2n_k$  Toffolis, simply by flipping the sign in usual two's complement binary, then controlling addition of  $N_x, N_y, N_z$  in each component.
8. Next we need to prepare a superposition over allowed values of  $\mathbf{k}$ , because  $\mathbf{k} - \mathbf{Q} - \mathbf{G}$  needs to be in the allowed range of  $\mathbf{k}$  values (using  $\mathbf{G}$  to indicate  $\mathbf{G}_1$  or  $\mathbf{G}_2$  depending on which part we are performing). In particular, for the  $x$  component we have an allowed range for  $k_x$  from  $Q_x$  to  $N_x - 1$  when  $G_x = 0$ , or 0 to  $Q_x - 1$  when  $G_x \neq 0$ . It is similar for the other two components. We can therefore prepare a superposition over the appropriate range then add  $Q_x$  if  $G_x = 0$ .

Creating an equal superposition requires Hadamards on the appropriate subset of qubits, as well as a  $Q_x, G_x$ -dependent rotation to give a high success probability for the amplitude amplification. This information can be output with Toffoli cost  $2N_x - 2$  on the qubits representing  $Q_x, G_x$ . The complexity of the controlled Hadamards is then  $\lceil \log N_x \rceil$  Toffolis, assuming we use a catalytic  $T$  state as in Ref. [33].

The complexity of preparing the equal superposition is then  $6\lceil\log N_x\rceil + 2b_r - 6$ , including  $3\lceil\log N_x\rceil$  for three rounds of  $\lceil\log N_x\rceil$  controlled Hadamards. The reason why there is  $-6$  rather than  $-9$  is the inequality test is with a value in a quantum register (in each of three tests), which requires one more Toffoli than an inequality test with a classically given value. The controlled addition of  $Q_x$  has complexity  $2\lceil\log N_x\rceil$ . The total complexity of the preparation of the superposition for the three components of  $\mathbf{k}$  is therefore

$$N_x + N_y + N_z + 8n_k + 6b_r - 24. \quad (\text{D9})$$

This cost is incurred 4 times for the preparation and inverse preparation of  $\mathbf{k}$  and  $\mathbf{k}'$ .

9. In order to account for the one-body term, we note that the one-body term has a single  $\mu$  and  $\mathbf{k}$  rather than  $\mathbf{Q}$ . We also do not want the operations we perform in the two-body part for the symmetry to affect the one-body part. We can therefore output  $\mu = \nu$  for the one-body part in the QROM, so the swap of  $\mu$  and  $\nu$  has no effect. The value of  $\mathbf{k}$  for the one-body part can be stored in the same register as used for  $\mathbf{Q}$  for the two-body part. To prevent the operations used to generate the symmetry  $\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_1,\mathbf{G}_2} = (\zeta_{\mu\nu}^{(\ominus\mathbf{Q}),\mathbf{G}_1,\mathbf{G}_2})^*$  being applied for the one-body part, we can simply apply a Toffoli to produce a new control qubit. The remaining part above is the preparation of the superposition over the  $\mathbf{k}$  values controlled on  $\mathbf{Q}$ ; this does not need to be amended to account for the one-body part because there we will not be using this value in the extra register.
10. Now that we have prepared the register that is in an equal superposition over the appropriate range of  $\mathbf{k}$ , we need to use that in combination with  $\mathbf{Q}$  and  $\mu$  to prepare a superposition with the correct weights. To do this, we will use coherent alias sampling in the usual way, but will need to construct an appropriate register to iterate over from registers  $\mathbf{k}, \mathbf{Q}, \mathbf{G}, \mu$ . First we compute  $\mathbf{k} - \mathbf{Q} - \mathbf{G}$  in an ancilla register. These two subtractions have cost  $2n_k$ . Since it needs to be computed and uncomputed for each of the two factors in the Hamiltonian, the total cost is  $8n_k$ . Now, because  $\mathbf{k}$  and  $\mathbf{k} \ominus \mathbf{Q}$  uniquely specify  $\mathbf{Q}, \mathbf{G}$ , these two registers can be used for the iteration instead of  $\mathbf{Q}$ , with the additional advantage that they are both over the full range of the Brillouin zone. Now we need to compute a contiguous register

$$\begin{aligned} & (((\mathbf{k}_x N_y + \mathbf{k}_y) N_z + \mathbf{k}_z) N_x \\ & + \mathbf{k}'_x N_y + \mathbf{k}'_y N_z + \mathbf{k}'_z) M + \mu, \end{aligned} \quad (\text{D10})$$

where we are using  $\mathbf{k}'$  for  $\mathbf{k} \ominus \mathbf{Q}$ . This contiguous register includes many multiplications by classically chosen constants, which has complexity depending on how many ones are in these constants. The worst case is where these numbers are all ones, so we will give the cost for that case even though it is rare.

As discussed in Ref. [126] the cost of multiplying two integers when one is given classically is no more than the product of the numbers of bits. For the additions, the cost is no more than the number of bits on the larger number. So, we have a cost as follows.

- (a) For multiplying  $\mathbf{k}_x N_y$ , a cost of  $\lceil\log N_x\rceil \lceil\log N_y\rceil$ . Here  $N_y$  would have more bits if it were a power of 2, but then the multiplication cost would be zero.
- (b) For adding  $+\mathbf{k}_y$ , a cost of  $\lceil\log N_x N_y\rceil$ .
- (c) For multiplying  $\times N_z$  the cost is  $\lceil\log N_x N_y\rceil \lceil\log N_z\rceil$ .
- (d) For adding  $+\mathbf{k}_z$  the cost is  $\lceil\log N_k\rceil$ .
- (e) For multiplying  $\times N_x$  the cost is  $\lceil\log N_k\rceil \lceil\log N_x\rceil$ .
- (f) For adding  $+\mathbf{k}'_x$  the cost is  $\lceil\log N_x N_k\rceil$ .
- (g) For multiplying  $\times N_y$  the cost is  $\lceil\log N_x N_k\rceil \lceil\log N_y\rceil$ .
- (h) For adding  $+\mathbf{k}'_y$  the cost is  $\lceil\log N_x N_y N_k\rceil$ .
- (i) For multiplying  $\times N_z$  the cost is  $\lceil\log N_x N_y N_k\rceil \lceil\log N_z\rceil$ .
- (j) For adding  $+\mathbf{k}'_z$  the cost is  $\lceil\log N_k^2\rceil$ .
- (k) For multiplying  $\times M$  the cost is  $\lceil\log N_k^2\rceil \lceil\log M\rceil$ .
- (l) For finally adding  $+\mu$  the cost is  $\lceil\log N_k^2 M\rceil$ .

We need to add all these items together to give the total cost, and it needs to be multiplied by 4 because we compute and uncompute for each of the two factors in the Hamiltonian.

Next we have a QROM on this contiguous register with cost

$$\left\lceil \frac{N_k^2 M}{k_{\text{nm}}} \right\rceil + (k_{\text{nm}} - 1)(n_k + \aleph). \quad (\text{D11})$$

with  $k_{\text{nm}}$  a power of 2. This is because there are  $N_k^2 M$  items to iterate over, and we need to output  $n_k$  bits for the alternate value of  $k$  and  $\aleph$  for the keep value. We have twice this cost because of the two factors in the Hamiltonian, but the erasure cost for each factor is

$$\left\lceil \frac{N_k^2 M}{k_{\text{era}}} \right\rceil + k_{\text{era}}. \quad (\text{D12})$$

The last two steps of the coherent alias sampling are an inequality test with cost  $\aleph$  and a controlled swap

with cost  $n_k$ . These costs are incurred 4 times, once for preparation and once for inverse preparation for each of the two factors for the Hamiltonian.

11. We will need to prepare a register that is  $\mathbf{k} - \mathbf{Q} - \mathbf{G}$  again. We previously computed this, but we need to compute it again because we have performed a state preparation on  $\mathbf{k}$ . This has a cost of  $2n_k$  again, and needs to be done 4 times for a total cost of  $8n_k$ . A further subtlety is that we are storing the value of  $\mathbf{k}$  to use in the  $\mathbf{Q}$  register in the one-body case. We can perform a controlled swap into the working register for  $\mathbf{k}$  or  $\mathbf{k} \ominus \mathbf{Q}$ , which has a total cost of  $4n_k$ . Combined with the arithmetic cost this is  $12n_k$ .
12. To use the register with  $\mathbf{k}$  or  $\mathbf{k} \ominus \mathbf{Q}$  to control the swap of system registers into working registers, we can use each qubit to control swaps of the system registers in a similar way as is used for advanced QROM. The cost for selecting each qubit out of  $N_k$  is  $N_k - 1$ , similar to the use in advanced QROM, despite the use of multiple components. In particular, we can perform swaps of system registers based on the  $x$  component of  $\mathbf{k}$  with cost  $(N_x - 1)N_yN_z$ . Then swapping the registers based on the  $y$  component out of the subset of  $N_yN_z$  has cost  $(N_y - 1)N_z$ . Then the cost of swapping based on the  $z$  component has cost  $N_z - 1$ . Adding these three costs together gives  $N_xN_yN_z - 1 = N_k - 1$ . This is performed for each of the  $N/2$  qubits we need, for cost  $N(N_k - 1)/2$ . We need to swap and inverse swap 8 times on  $NN_k/2$  system registers, for a total cost of  $4N(N_k - 1)$ .
13. Next, we consider the output of the rotations for the  $c$  modes. These will be controlled by the registers with  $\mu$  and  $\mathbf{k}$  (or  $\mathbf{k} \ominus \mathbf{Q}$ ), as well as the qubit selecting between the one- and two-body terms. A difficulty is that we would need a contiguous register in order to be able to effectively apply the advanced QROM. A method around this is to use a QROM on the selection qubit and  $\mathbf{k}$  to output an offset, then add  $\mu$ . The complexity of that QROM is  $2N_k$ , then the complexity of the addition is  $\lceil \log N_k(M + N/2) \rceil$ . That is in the case where we need to apply the one-body term as part of the implementation. Recall that we need only to do that once when we are applying  $c^\dagger c$  twice for the two-body term. In the part where we are not applying the one-body term we instead have complexity  $N_k + \lceil \log N_k M \rceil$ . The size of the QROM output for the rotations is then  $N\lceil \rceil$ . That is again because we need Givens rotations on  $N/2$  qubits, and need two angles for each Givens rotation with  $\lceil \rceil$  each. The complexity is then, in the case with the one-body term,

$$\left\lceil \frac{N_k(M + N/2)}{k_r} \right\rceil + N\lceil \rceil(k_r - 1), \quad (\text{D13})$$

and for the case where we do not need the one-body term

$$\left\lceil \frac{N_k M}{k_r} \right\rceil + N\lceil \rceil(k_r - 1), \quad (\text{D14})$$

with  $k_r$  being a power of 2. The cost of the Givens rotations is  $2N(\lceil \rceil - 2)$ , because we have  $N/2$  qubits and two angles for each Givens rotation.

The cost of erasing the QROM for the two cases is

$$\left\lceil \frac{N_k(M + N/2)}{k'_r} \right\rceil + k'_r, \quad (\text{D15})$$

$$\left\lceil \frac{N_k M}{k'_r} \right\rceil + k'_r. \quad (\text{D16})$$

Lastly we note that we need to apply the sequence of Givens rotations 8 times to account for the four  $c^\dagger$  and  $c$  operators, and similarly we need to apply the QROM and invert it 4 times. That gives a total complexity for the QROM-based basis rotations

$$\begin{aligned} & 2 \left\lceil \frac{N_k(M + N/2)}{k_r} \right\rceil + 2N\lceil \rceil(k_r - 1) + 2 \left\lceil \frac{N_k M}{k_r} \right\rceil \\ & + 2N\lceil \rceil(k_r - 1) + 2 \left\lceil \frac{N_k(M + N/2)}{k'_r} \right\rceil + 2k'_r \\ & + 2 \left\lceil \frac{N_k M}{k'_r} \right\rceil + 2k'_r + 16N(\lceil \rceil - 2) \\ & + 12N_k + 4\lceil \log N_k(M + N/2) \rceil + 4\lceil \log N_k M \rceil, \end{aligned} \quad (\text{D17})$$

where  $16N(\lceil \rceil - 2)$  is for the Givens rotations themselves, and the terms from  $12N_k$  on are for creating and erasing contiguous registers.

14. The next part of the complexity that needs to be accounted for is the selection of  $\bar{X}$  and  $\bar{Y}$  for the implementation of the  $c^\dagger$  and  $c$  operators. We need only to select between  $\bar{X}$  and  $\bar{Y}$ , but not select the location the  $X$  or  $Y$  is performed since these are applied in the working registers. This selection can therefore be performed entirely with Clifford gates. However, we do need additional control to avoid performing these operators for one of the  $c^\dagger c$  for the one-body component. That is a cost of just one Toffoli for each of  $c^\dagger$  and  $c$ .

A complication is that we need to perform  $Z$  gates on remaining system registers (that have not been swapped into the working registers). We perform unary iteration on the register containing  $\mathbf{k}$  (or  $\mathbf{k} \ominus \mathbf{Q}$ ), and use that to apply the appropriate  $Z$  operators with a Toffoli cost  $N_k - 1$ . The Toffoli complexity is independent of  $N$ , because we only have a Toffoli



cost for the unary iteration, not for the controlled- $Z$  gates.

15. The last part to the complexity when constructing the qubitized operator is the reflection on the control ancillas. The qubits we need to reflect on are as follows.
  - (a) The  $\lceil \log d \rceil$  qubits used for the equal superposition state for the initial preparation, and another qubit rotated for this preparation.
  - (b) The  $\aleph$  used for the equal superposition state for the inequality test in state preparation.
  - (c) The two qubits for the two spins in the sum.
  - (d) There are  $n_k$  qubits that an equal superposition of  $\mathbf{k}$  values is prepared in, and this is done twice for  $2n_k$  qubits. To save on qubit use we can also reuse these qubits and flag that they are equal to zero in between preparing  $\mathbf{k}$  and  $\mathbf{k}'$ , but that does not affect the Toffoli count.
  - (e) There are  $\aleph$  used for the equal superposition state for two rounds of state preparation for  $\mathbf{k}$ . Again these can be reused but that does not affect the qubit count.
  - (f) There are four qubits to select between  $X$  and  $Y$  for each of the  $c^\dagger, c$ .
  - (g) There are two qubits used to generate the symmetries.

This is a total of

$$\lceil \log d \rceil + 3\aleph + 2n_k + 9 \quad (\text{D18})$$

qubits for the control, and the reflection cost is 2 less than this. We do require an additional Toffoli for the control by the phase-estimation registers, and another for unary iteration on the phase-estimation registers. Therefore, this expression can be used for the additional Toffoli cost.

Next we account for the qubit costs.

1. There are  $NN_k$  system qubits.
2. The control register for the phase estimation, and the ancillas for the unary iteration, together need  $2\lceil \log \mathcal{I} \rceil - 1$  qubits.
3. There are all the qubits needed for controls as described in the last item in the Toffoli costing above. Taking the inversion of the equal superposition over  $\mathbf{k}$  to be flagged by a single qubit, the  $2n_k$  can be replaced with  $n_k + 1$ . Similarly flag qubits can be used on the qubits used for the equal superposition state for the inequality test to replace  $3\aleph$  with  $\aleph + 2$  qubits. That gives a number of qubits

$$\lceil \log d \rceil + \aleph + n_k + 12. \quad (\text{D19})$$

4. A phase gradient state of size  $\beth$  is used for rotations.
5. A single  $T$  state is used for the controlled Hadamards.
6. The QROM used for the first state preparation outputs  $m$  qubits.
7. This first state preparation also uses  $m(k_p - 1) + \lceil \log(d/k_p) \rceil - 1$  temporary qubits.
8. A single qubit is used for the result of the inequality test for the coherent alias sampling.
9. A single qubit is used for the result of the inequality test between  $\mu, \mathbf{G}_1$  and  $\nu, \mathbf{G}_2$ .
10. There are  $n_k + 3b_r$  qubits used as the output of the QROM used to give the information needed for the preparation of the equal superposition over  $\mathbf{k}$ , as well as  $b_r$  for  $\mathbf{k}$ .
11. In the preparation of the state for  $\mathbf{k}$  we compute  $\mathbf{k} \ominus \mathbf{Q}$  in an ancilla needing  $n_k$  qubits, and compute a contiguous register that needs  $\lceil \log(N_k^2 M) \rceil$  qubits.
12. The state preparation for  $\mathbf{k}$  uses  $n_k + \beth$  output qubits.
13. The state preparation uses

$$(k_{\text{nm}} - 1)(n_k + \beth) + \left\lceil \log \left( \frac{N_k^2 M}{k_{\text{nm}}} \right) \right\rceil - 1 \quad (\text{D20})$$

temporary qubits.

14. We also use  $\aleph$  in this state preparation for a superposition state, and another qubit for the result of the inequality test.
15. There are  $\lceil \log N_k(M + N/2) \rceil$  qubits used for the contiguous register for the QROM for the qubit rotations.
16. There are  $N\beth k_r$  used for the QROM for the rotations, with another

$$\left\lceil \log \left( \frac{N_k(M + N/2)}{k_r} \right) \right\rceil - 1 \quad (\text{D21})$$

temporary qubits.

Accounting for the maximum total involves determining the maximum number used, which depends on the use of temporary ancillas. We first need to determine whether the temporary qubits described in part 13 or the total qubits in parts 14 to 16 are larger. We take the maximum of these, and add it to the qubits used in parts 8 to 12. Then we compare that to the number of temporary qubits in part 7. The maximum of that is added to the qubits in parts 1 to 6.

Next we give a little more detail on how the construction is made self-inverse. As explained in Fig. 6 of Ref. [33] (shown above as Fig. 15) the THC construction may be made self-inverse by using the qubit in the  $|+\rangle$  state, which controls the swap of the  $|\mu\rangle$  and  $|\nu\rangle$  registers. As can be seen in Fig. 6, we are using a similar procedure, where the  $|+\rangle$  state controls the swap of  $\mu$  and  $\nu$  at the top left

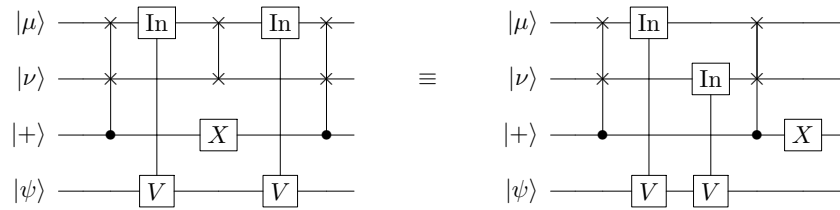


FIG. 15. This shows how to construct a self-inverse procedure for block encoding two-electron terms, as in Fig. 6 of Ref. [33]. The left side is the manifestly self-inverse form, and the right side is a more intuitive form where the  $|+\rangle$  state is used to generate the symmetry between  $\mu$  and  $\nu$ , and the two  $V$  operations are controlled by  $\mu$  and  $\nu$  in succession.

and at the lower right. The  $X$  gate and swaps on the lower left corresponds to the  $X$  gate and swap in the middle of Fig. 15.

We have currently just drawn the quantum circuit as having  $c$  and  $c^\dagger$ , but these would be implemented using ancilla qubits to control the election between  $X$  and  $Y$  in  $X \pm iY$ . For the implementation to be self-inverse, we would want the qubit used to control the first  $c$  to be the same as that for the final  $c^\dagger$ . This can be achieved by taking the four qubits for control of each of the  $c$  and  $c^\dagger$  so the top one can be used as the control each time. In particular, after the first  $c$ , swap the first two qubits, then after the  $c^\dagger$  swap the first pair with the second pair, then after the next  $c$  swap the first two qubits again. As a result, the first qubit can be used as the control each time. Moreover, this arrangement of swaps is obviously self-inverse.

The application of the controlled  $X$  and  $iY$  gates is also automatically self-inverse. The reason is that  $iY$  squared is  $-\mathbb{1}$ . In the block encoding we perform  $Z$  gates for the control qubits for  $c$  but not  $c^\dagger$ . Then performing the unitaries for the block encoding twice, we have first that the final controlled  $c^\dagger$  in the first block encoding is matched with the first  $c$  for the second block encoding. The same control qubit is used, so the two controlled  $X$  operations cancel, and the two controlled  $iY$  operations give  $-\mathbb{1}$ . This cancels with the  $Z$  gate on that control qubit. In this way all the operations can be canceled, and because each time we have controlled  $c$  and  $c^\dagger$  matched, which cancel the  $Z$  gate on the control qubit. There is no additional Toffoli cost for these swaps and phase gates, because they are all Clifford gates.

### APPENDIX E: CORRELATION DIAGNOSTICS FOR LNO

For insulators like the distorted structures of  $\text{LiNiO}_2$ , there are several common diagnostics that are used in molecular calculations to identify cases of “strong correlation.” Here we examine the maximum elements of the CCSD  $T_1$  and  $T_2$  tensors ( $\max(|t_1|)$  and  $\max(|t_2|)$ ), which are commonly used as a measure of correlation [127], as well as the T1 [128] and D1 [129,130] diagnostics computed from the ROHF  $k$ -point CCSD calculations. We also

show the expectation value of the  $S^2$  operator for the UHF solution because spin contamination in the UHF calculation can be a signature of strong correlation. The results are shown in Table VIII. As expected, these results do not suggest particularly strong correlation. Only the  $\max(|t_1|)$  values for the  $C2/m$  and  $P2_1/c$  structures and the UHF  $S^2$  for the  $P2/c$  structure are larger than might be expected. The larger  $\max(|t_1|)$  is likely an indication that the ROHF orbitals are far from optimal, and the symmetry breaking in the UHF calculations does not mean that CCSD cannot provide reliable energies.

### APPENDIX F: CLASSICAL TIMING BENCHMARKS

In order to compare the quantum algorithm runtime to state of the art classical algorithms we measured the cost to compute the CCSD and ph-AFQMC total energy for the benchmark systems listed in Table I in double- and triple- $\zeta$  basis sets. The results of these timings are presented in Fig. 16.

For CCSD we used pycscf [101,102] and timed the data on a node with 30 3.1 GHz Intel Xeon CPUs (30 OpenMP threads).

For ph-AFQMC, which is considerably more expensive than CCSD for small system sizes, we estimated the runtime by performing a short ph-AFQMC calculation for each system using 8 Nvidia V100 GPUs. From this data we can then estimate the runtime to achieve a statistical error bar (per atom) of  $1 \times 10^{-4}$  Ha through the assumption that the statistical error of ph-AFQMC decays like  $N_s^{-1/2}$ ,

TABLE VIII. Some common diagnostics of strong correlation from the ROHF-CCSD calculations for each of the distorted  $\text{LiNiO}_2$  structures. The UHF  $S^2$  values in the final column are given per formula unit and should be compared with the exact doublet ( $\langle S^2 \rangle = 0.75$ ). The basis set (GTH-DZVP) and other details of the calculations are described in Sec. VB.

Structure	$\max( t_1 )$	$\max( t_2 )$	T1	D1	UHF $S^2$
$C2/m$	0.2538	0.0330	0.0482	0.1912	0.7783
$P2_1/c$	0.2313	0.0322	0.0472	0.2178	0.8027
$P2/c$	0.1688	0.0571	0.0371	0.2089	1.0447

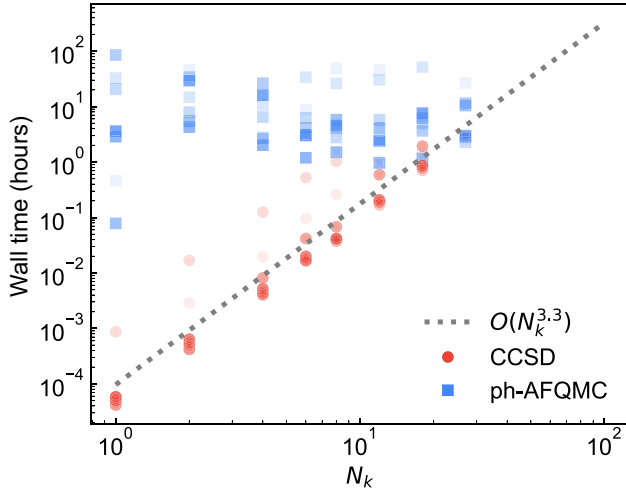


FIG. 16. CCSD and ph-AFQMC timings for the benchmark systems in Table I (hydrogen omitted) in the cc-pVDZ basis set. The CCSD data follow a  $\mathcal{O}(N_k^{3.4})$  scaling, which is consistent with the expected  $\mathcal{O}(N_k^4)$  scaling [18].

where  $N_s$  is the number of Monte Carlo samples. Formally, ph-AFQMC should asymptotically scale like  $\mathcal{O}(N_k^3)$  [23] assuming the number of samples required to reach the desired precision does not scale with the system size. Interestingly, we found that the statistical error bar per atom for a fixed number of samples actually decreased with  $N_k$ , which implies the variance of ph-AFQMC is increasing sublinearly with the system size. For smaller system sizes it is important to note that practically one can saturate the GPU with walkers with nearly no loss speed [117], thus reducing the error bar given a fixed wall time. As a result, the small  $N_k$  AFQMC numbers represent a large overestimation in runtime one would practically need to obtain the desired statistical error. Another confounding factor, which may affect the scaling of ph-AFQMC is the time-step error (we fixed the time step at  $0.005 \text{ Ha}^{-1}$  for all systems). Recent results suggest that ph-AFQMC suffers from a size extensivity error [131], which is practically remedied through time-step extrapolation. This may break the assumption of a constant number of Monte Carlo steps required to reach the desired precision (with a fixed time step) [131]. Due to these factors we decided not to fit a trend line to the ph-AFQMC data.

We should stress that these comparisons are for illustrative purposes only, the absolute timings will differ with different classical implementations, codes and architectures and the runtimes should be considered with these factors in mind.

## APPENDIX G: GENERATING THE THC FACTORS

To generate the  $k$ -point-dependent THC factors we used the interpolative separable density fitting approach (ISDF) [88,90,91] as a starting point, which has recently been

adapted to incorporate translational symmetry [88,89,132]. Recall from Eq. (58) that the goal is to approximate the cell periodic orbital product evaluated on a dense real-space grid using a much reduced set of grid points. In particular, we aim to solve

$$Z = u_{p\mathbf{k}_p}^*(\mathbf{r})u_{q\mathbf{k}_q}(\mathbf{r}) \approx \sum_{\mu} \xi_{\mu}(\mathbf{r})u_{p\mathbf{k}_p}^*(\mathbf{r}_{\mu})u_{q\mathbf{k}_q}(\mathbf{r}_{\mu}) = \Theta C, \quad (\text{G1})$$

for  $\xi_{\mu}(\mathbf{r})$ , where  $N_{\mu} = c_{\text{THC}}N/2$ ,  $\{\mathbf{r}_{\mu}\}$  is a subset of the real-space points  $\{\mathbf{r}\}$ , and  $c_{\text{THC}}$  is the THC rank parameter. We used the  $K$ -means clustering centroidal Voronoi tessellation (CVT) algorithm to find the initial grid points [91]. The interpolating vectors  $\Theta_{\mu\mathbf{r}} = \xi_{\mu}(\mathbf{r})$  can be found via

$$\Theta = ZC^{\dagger}(CC^{\dagger})^{-1}, \quad (\text{G2})$$

where both  $ZC^{\dagger}$  and  $CC^{\dagger}$  can be efficiently formed due to their product separable form [90].

Once the interpolating points ( $\{\mathbf{r}_{\mu}\}$ ) and vectors ( $\xi_{\mu}(\mathbf{r})$ ) have been found we can form any ERI block via

$$V_{p\mathbf{k}_p,q\mathbf{k}_q,r\mathbf{k}_r,s\mathbf{k}_s} = \sum_{\mu\nu} \chi_{p\mathbf{k}_p}^{(\mu)*} \chi_{q\mathbf{k}_q}^{(\nu)} \zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_{pq},\mathbf{G}_{sr}} \chi_{r\mathbf{k}_r}^{(v)*} \chi_{s\mathbf{k}_s}^{(v)}, \quad (\text{G3})$$

where

$$\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_{pq},\mathbf{G}_{sr}} = \int d\mathbf{r} \int d\mathbf{r}' e^{-i(\mathbf{Q}+\mathbf{G}_{pq})\cdot\mathbf{r}} \xi_{\mu}(\mathbf{r}) V(\mathbf{r},\mathbf{r}') \xi_{\nu}(\mathbf{r}') e^{i(\mathbf{Q}+\mathbf{G}_{sr})\cdot\mathbf{r}'}. \quad (\text{G4})$$

To evaluate Eq. (G4) we follow the Gaussian plane-wave density-fitting recipe [50,60] and perform the following steps:

1.  $\xi_{\mu}(\mathbf{G}) = \text{FFT}[\xi_{\mu}(\mathbf{r})]$ ,
2.  $\tilde{V}_{\mu}^{\mathbf{Q},\mathbf{G}_{pq}}(\mathbf{r}') = e^{-i(\mathbf{Q}+\mathbf{G}_{pq})\cdot\mathbf{r}'} \text{IFFT}[\xi_{\mu}(\mathbf{G})V(|\mathbf{G} - (\mathbf{Q} + \mathbf{G}_{pq})|)]$ ,
3.  $\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G}_{pq},\mathbf{G}_{sr}} = \sum_{\mathbf{r}'} \Delta\mathbf{r}' \xi_{\nu}(\mathbf{r}') e^{i(\mathbf{G}_{sr}-\mathbf{G}_{pq})\cdot\mathbf{r}'} \tilde{V}_{\mu}^{\mathbf{Q},\mathbf{G}_{pq}}(\mathbf{r}')$ ,

where  $\Delta\mathbf{r}' = \Omega/N_g$ ,  $\Omega$  is the unit-cell volume,  $N_g$  is the number of real-space grid points, and  $V(\mathbf{G})$  is the Coulomb kernel in reciprocal space.

An important consideration for the quantum implementation of THC is the minimization of the L1 norm of the central tensor  $\zeta_{\mu\nu}^{\mathbf{Q}}$  as this value directly affects the scaling of the algorithm. Following previous work for molecular systems [33,46], we attempt to reduce  $\lambda$  by further compressing the THC factors through a regularized optimization scheme. This is an important step as  $\lambda$  can grow considerably with the THC rank. We use the ISDF solution for  $\chi_{p\mathbf{k}_p}(\mathbf{r}_{\mu})$ , and  $\zeta_{\mu\nu}^{\mathbf{Q},\mathbf{G},\mathbf{G}'}$  as an initial guess and perform a

subsequent optimization of the following cost function:

$$L(\chi, \zeta) = \sum_{\mathbf{Q}} \sum_{\mathbf{k}\mathbf{k}'} \sum_{pqrs} \left( \left| (\mathbf{k}p\mathbf{k} \ominus \mathbf{Q}q|r\mathbf{k}' \ominus \mathbf{Q}s\mathbf{k}') - \sum_{\mu\nu} \chi_{p\mathbf{k}}^{(\mu)*} \chi_{q\mathbf{k} \ominus \mathbf{Q}}^{(\mu)} \zeta_{\mu\nu}^{\mathbf{Q}\mathbf{G},\mathbf{G}'} \chi_{r(\mathbf{k}' \ominus \mathbf{Q})}^{(\nu)*} \chi_{s\mathbf{k}'}^{(\nu)} \right|^2 + p \left| \sum_{\mu\nu} \mathcal{N}_{p\mathbf{k}q(\mathbf{k} \ominus \mathbf{Q})}^{\mu} \zeta_{\mu\nu}^{\mathbf{Q}\mathbf{G}\mathbf{G}'} \mathcal{N}_{r(\mathbf{k}' \ominus \mathbf{Q})s\mathbf{k}'}^{\nu} \right| \right), \quad (\text{G5})$$

where the penalty parameter  $p$  was set such that the  $L_1$  part of the objective function was of similar magnitude to the  $L_2$  part. To optimize the cost function we used the L-BFGS-B implemented in Scipy with gradients evaluated using JAX. Like Ref. [33] we found it necessary to perform an additional optimization of the  $L_2$  part of the loss function after the BFGS optimization in order to obtain good MP2 correlation energies. As these subsequent optimizations are quite costly we limited the number of optimization steps to 3000 for both the L-BFGS-B and AdaGrad stages. With these additional steps we found a rank parameter of  $c_{\text{THC}} = 6 - 8$  was sufficient to obtain an MP2 error of roughly 0.1 mHa error per cell across a range of systems, which is consistent with that found in molecular systems [46].

- [1] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, *Proc. Natl. Acad. Sci.* **105**, 18681 (2008).
- [2] R. Babbush, C. Gidney, D. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, *Phys. Rev. X* **8**, 041015 (2018).
- [3] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, Fault-tolerant quantum simulations of chemistry in first quantization, *PRX Quantum* **2**, 040332 (2021).
- [4] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-depth quantum simulation of materials, *Phys. Rev. X* **8**, 011044 (2018).
- [5] A. Grüneis, J. J. Shepherd, A. Alavi, D. P. Tew, and G. H. Booth, Explicitly correlated plane waves: Accelerating convergence in periodic wavefunction expansions, *J. Chem. Phys.* **139**, 84112 (2013).
- [6] T. Kato, On the eigenfunctions of many-particle systems in quantum mechanics, *Commun. Pure Appl. Math.* **10**, 151 (1957).
- [7] J. Lee, L. Lin, and M. Head-Gordon, Systematically improvable tensor hypercontraction: Interpolative separable density-fitting for molecules applied to exact exchange, second- and third-order Møller–Plesset perturbation theory, *J. Chem. Theory Comput.* **16**, 243 (2019).
- [8] M. Imada, A. Fujimori, and Y. Tokura, Metal-insulator transitions, *Rev. Mod. Phys.* **70**, 1039 (1998).
- [9] E. Dagotto, Correlated electrons in high-temperature superconductors, *Rev. Mod. Phys.* **66**, 763 (1994).
- [10] S. Sachdev, *Colloquium: Order and quantum phase transitions in the cuprate superconductors*, *Rev. Mod. Phys.* **75**, 913 (2003).
- [11] C. Pisani, Local techniques for the ab initio quantum-mechanical description of the chemical properties of crystalline materials, *J. Mol. Struct. (Theochem)* **621**, 141 (2003).
- [12] C. Pisani, M. Busso, G. Capecchi, S. Casassa, R. Dovesi, L. Maschio, C. Zicovich-Wilson, and M. Schütz, Local-MP2 electron correlation method for nonconducting crystals, *J. Chem. Phys.* **122**, 094113 (2005).
- [13] A. Grüneis, G. H. Booth, M. Marsman, J. Spencer, A. Alavi, and G. Kresse, Natural orbitals for wave function based correlated calculations using a plane wave basis set, *J. Chem. Theory Comput.* **7**, 2780 (2011).
- [14] M. D. Ben, J. Hutter, and J. Vandevondele, Second-order Møller–Plesset perturbation theory in the condensed phase: An efficient and massively parallel Gaussian and plane waves approach, *J. Chem. Theory Comput.* **8**, 4177 (2012).
- [15] M. D. Ben, J. Hutter, and J. Vandevondele, Electron correlation in the condensed phase from a resolution of identity approach based on the Gaussian and plane waves scheme, *J. Chem. Theory Comput.* **9**, 2654 (2013).
- [16] G. H. Booth, A. Grüneis, G. Kresse, and A. Alavi, Towards an exact description of electronic wavefunctions in real solids, *Nature* **493**, 365 (2013).
- [17] G. H. Booth, T. Tsatsoulis, G. K. L. Chan, and A. Grüneis, From plane waves to local Gaussians for the simulation of correlated periodic systems, *J. Chem. Phys.* **145**, 084111 (2016).
- [18] J. McClain, Q. Sun, G. K. L. Chan, and T. C. Berkelbach, Gaussian-based coupled-cluster theory for the ground-state and band structure of solids, *J. Chem. Theory Comput.* **13**, 1209 (2017).
- [19] V. A. Neufeld, H.-Z. Ye, and T. C. Berkelbach, Ground-state properties of metallic solids from ab initio coupled-cluster theory, *J. Phys. Chem. Lett.* **13**, 7497 (2022).
- [20] Z. H. Cui, T. Zhu, and G. K. L. Chan, Efficient formulation of ab initio quantum embedding in periodic systems: Dynamical mean-field theory, *J. Chem. Theory Comput.* **16**, 119 (2020).
- [21] T. Zhu, Z. H. Cui, and G. K. L. Chan, Efficient formulation of ab initio quantum embedding in periodic systems: Dynamical mean-field theory, *J. Chem. Theory Comput.* **16**, 141 (2020).
- [22] Z.-H. Cui, H. Zhai, X. Zhang, and G. K.-L. Chan, Systematic electronic structure in the cuprate parent state from quantum many-body simulations, *Science* **377**, 1192 (2022).
- [23] M. Motta, S. Zhang, and G. K.-L. Chan, Hamiltonian symmetries in auxiliary-field quantum Monte Carlo calculations for electronic structure, *Phys. Rev. B* **100**, 045127 (2019).
- [24] F. A. Cotton, *Chemical Applications of Group Theory* (John Wiley & Sons, Hoboken, NJ, 1991).
- [25] T. D. Crawford and R. Di Remigio, in *Annual Reports in Computational Chemistry*, Vol. 15 (Elsevier, 2019), p. 79.



- [26] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions, *Rev. Mod. Phys.* **68**, 13 (1996).
- [27] P. Cortona, Self-consistently determined properties of solids without band-structure calculations, *Phys. Rev. B* **44**, 8454 (1991).
- [28] J. Inglesfield, A method of embedding, *J. Phys. C: Solid State Phys.* **14**, 3795 (1981).
- [29] G. Knizia and G. K.-L. Chan, Density matrix embedding: A simple alternative to dynamical mean-field theory, *Phys. Rev. Lett.* **109**, 186404 (2012).
- [30] Z.-H. Cui, T. Zhu, and G. K.-L. Chan, Efficient implementation of ab initio quantum-embedding in periodic systems: Density matrix-embedding theory, *J. Chem. Theory Comput.* **16**, 119 (2019).
- [31] H. Q. Pham, M. R. Hermes, and L. Gagliardi, Periodic electronic structure calculations with the density matrix embedding theory, *J. Chem. Theory Comput.* **16**, 130 (2019).
- [32] H. Zheng, H. J. Changlani, K. T. Williams, B. Busemeyer, and L. K. Wagner, From real materials to model Hamiltonians with density matrix downfolding, *Front. Phys.* **6**, 43 (2018).
- [33] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, *PRX Quantum* **2**, 030305 (2021).
- [34] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [35] A. V. Ivanov, C. Sünderhauf, N. Holzmann, T. Ellaby, R. N. Kerber, G. Jones, and J. Camps, *ArXiv:2210.02403* (2022).
- [36] Y. A. Bychkov and E. I. Rashba, Oscillatory effects and the magnetic susceptibility of carriers in inversion layers, *J. Phys. C: Solid State Phys.* **17**, 6039 (1984).
- [37] C. L. Kane and E. J. Mele, Quantum spin Hall effect in Graphene, *Phys. Rev. Lett.* **95**, 226801 (2005).
- [38] F. Giustino, Electron-phonon interactions from first principles, *Rev. Mod. Phys.* **89**, 015003 (2017).
- [39] M. Desjarlais, J. Kress, and L. Collins, Electrical conductivity for warm, dense aluminum plasmas and liquids, *Phys. Rev. E* **66**, 025401 (2002).
- [40] M. Pozzo, C. Davies, D. Gubbins, and D. Alfe, Thermal and electrical conductivity of iron at Earth's core conditions, *Nature* **485**, 355 (2012).
- [41] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Theory of Superconductivity, *Phys. Rev.* **108**, 1175 (1957).
- [42] G. H. Low, V. Kliuchnikov, and L. Schaeffer, *ArXiv:1812.00954* (2018).
- [43] R. Babbush, C. Gidney, D. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear t complexity, (2018).
- [44] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, *Proc. Natl. Acad. Sci.* **114**, 7555 (2017).
- [45] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, *Phys. Rev. Res.* **3**, 033055 (2021).
- [46] J. J. Goings, A. White, J. Lee, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers, *Proc. Natl. Acad. Sci.* **119**, e2203533119 (2022).
- [47] M. Bianchini, M. Roca-Ayats, P. Hartmann, T. Brezesinski, and J. Janek, There and back again—the journey of LiNiO<sub>2</sub> as a cathode active material, *Angew. Chem. Int. Ed.* **58**, 2 (2019).
- [48] S. Siculo, M. Mock, M. Bianchini, and K. Albe, And yet it moves: LiNiO<sub>2</sub>, a dynamic Jahn-Teller system, *Chem. Mater.* **32**, 10096 (2020).
- [49] H. Chen, C. L. Freeman, and J. H. Harding, Charge disproportionation and Jahn-Teller distortion in LiNiO<sub>2</sub> and NaNiO<sub>2</sub>: A density functional theory study, *Phys. Rev. B* **84**, 085108 (2011).
- [50] J. Vandevondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter, Quickstep: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach, *Comput. Phys. Commun.* **167**, 103 (2005).
- [51] T. D. Kühne, *et al.*, CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations, *J. Chem. Phys.* **152**, 194103 (2020).
- [52] R. Dovesi, F. Pascale, B. Civalleri, K. Doll, N. M. Harrison, I. Bush, P. D'Arco, Y. Noel, M. Rera, P. Carbonniere, M. Causa, S. Salustro, V. Lacivita, B. Kirtman, A. M. Ferrari, F. S. Gentile, J. Baima, M. Ferrero, R. Demichelis, and M. D. L. Pierre, The CRYSTAL code, 1976–2020 and beyond, a long story, *J. Chem. Phys.* **152**, 204111 (2020).
- [53] M. Guidon, J. Hutter, and J. VandeVondele, Robust periodic Hartree-Fock exchange for large-scale simulations using Gaussian basis sets, *J. Chem. Theory Comput.* **5**, 3010 (2009).
- [54] M. Guidon, J. Hutter, and J. VandeVondele, Auxiliary density matrix methods for Hartree-Fock exchange calculations, *J. Chem. Theory Comput.* **6**, 2348 (2010).
- [55] C. Pisani, R. Dovesi, and C. Roetti, Hartree-Fock ab initio treatment of crystalline systems, (1988).
- [56] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Ab initio molecular simulations with numeric atom-centered orbitals, *Comput. Phys. Commun.* **180**, 2175 (2009).
- [57] S. Goedecker, Linear scaling electronic structure methods, *Rev. Mod. Phys.* **71**, 1085 (1999).
- [58] D. R. Bowler and T. Miyazaki,  $\mathcal{O}(N)$  methods in electronic structure calculations, *Rep. Prog. Phys.* **75**, 036503 (2012).
- [59] X. Wu, A. Selloni, and R. Car, Order-N implementation of exact exchange in extended insulating systems, *Phys. Rev. B - Condens. Matter Mater. Phys.* **79**, 085102 (2009).
- [60] G. Lippert, J. Hutter, and M. Parrinello, A hybrid Gaussian and plane wave density functional scheme, *Mol. Phys.* **92**, 477 (1997).
- [61] J. L. Whitten, Coulombic potential energy integrals and approximations, *J. Chem. Phys.* **58**, 4496 (1973).
- [62] J. W. Mintmire and B. I. Dunlap, Fitting the Coulomb potential variationally in linear-combination-of-



- atomic-orbitals density-functional calculations, *Phys. Rev. A* **25**, 88 (1982).
- [63] B. I. Dunlap, Robust and variational fitting: Removing the four-center integrals from center stage in quantum chemistry, *J. Mol. Struct.: THEOCHEM* **529**, 37 (2000).
- [64] F. Weigend, A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).
- [65] S. Varga, Density fitting of Coulomb integrals in electronic structure calculations of solids, *Phys. Rev. B* **71**, 073103 (2005).
- [66] L. Maschio and D. Usvyat, Fitting of local densities in periodic systems, *Phys. Rev. B* **78**, 073102 (2008).
- [67] A. M. Burow, M. Sierka, and F. Mohamed, Resolution of identity approximation for the Coulomb term in molecular and periodic systems, *J. Chem. Phys.* **131**, 214101 (2009).
- [68] X. Wang, C. A. Lewis, and E. F. Valeev, Efficient evaluation of exact exchange for periodic systems via concentric atomic density fitting, *J. Chem. Phys.* **153**, 124116 (2020).
- [69] H. Z. Ye and T. C. Berkelbach, Fast periodic Gaussian density fitting by range separation, *J. Chem. Phys.* **154**, 131104 (2021).
- [70] E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, Tensor hypercontraction density fitting. I. Quartic scaling second- and third-order Møller-Plesset perturbation theory, *J. Chem. Phys.* **137**, 1085 (2012).
- [71] R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, Tensor hypercontraction. II. Least-squares renormalization, *J. Chem. Phys.* **137**, 224106 (2012).
- [72] E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martínez, Communication: Tensor hypercontraction. III. Least-squares tensor hypercontraction for the determination of correlated wavefunctions, *J. Chem. Phys.* **137**, 221101 (2012).
- [73] H.-Z. Ye and T. C. Berkelbach, Correlation-consistent Gaussian basis sets for solids made simple, *J. Chem. Theory Comput.* **18**, 1595 (2022).
- [74] C. Hartwigsen, S. Goedecker, and J. Hutter, Relativistic separable dual-space Gaussian pseudopotentials from H to Rn, *Phys. Rev. B* **58**, 3641 (1998).
- [75] J. Heyd, J. E. Peralta, G. E. Scuseria, and R. L. Martin, Energy band gaps and lattice parameters evaluated with the Heyd-Scuseria-Ernzerhof screened hybrid functional, *J. Chem. Phys.* **123**, 174101 (2005).
- [76] A. Grüneis, M. Marsman, and G. Kresse, Second-order Møller-Plesset perturbation theory applied to extended systems. II. Structural and energetic properties, *J. Chem. Phys.* **133**, 074107 (2010).
- [77] M. R. Nadler and C. Kempier, Crystallographic Data 186. Lithium, *Anal. Chem.* **31**, 2109 (1959).
- [78] F. L. Tang, X. X. Che, W. J. Lu, G. B. Chen, Y. Xie, and W. Y. Yu, Surface structure and solidification morphology of aluminum nanoclusters, *Phys. B: Condens. Matter* **404**, 2489 (2009).
- [79] D. W. Berry, C. Gidney, M. Motta, J. McClean, and R. Babbush, Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization, *Quantum* **3**, 208 (2019).
- [80] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019), p. 193.
- [81] M. Szegedy, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2004), p. 32.
- [82] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, *Quantum Inf. Comput.* **12**, 901 (2012).
- [83] A. Baldereschi, Mean-value point in the Brillouin zone, *Phys. Rev. B* **7**, 5212 (1973).
- [84] G. H. Low, V. Kliuchnikov, and L. Schaeffer, [ArXiv:1812.00954](https://arxiv.org/abs/1812.00954) (2018).
- [85] H.-J. Werner, F. R. Manby, and P. J. Knowles, Fast linear scaling second-order Møller-Plesset perturbation theory (MP2) using local and density fitting approximations, *J. Chem. Phys.* **118**, 8149 (2003).
- [86] L. Maschio and D. Usvyat, Fitting of local densities in periodic systems, *Phys. Rev. B* **78**, 073102 (2008).
- [87] O. Oumarou, M. Scheurer, R. M. Parrish, E. G. Hohenstein, and C. Gogolin, [ArXiv:2212.07957](https://arxiv.org/abs/2212.07957) (2022).
- [88] J. Lu and L. Ying, Fast algorithm for periodic density fitting for Bloch waves, *Ann. Math. Sci. Appl.* **1**, 321 (2016).
- [89] K. Wu, X. Qin, W. Hu, and J. Yang, Low-rank approximations accelerated plane-wave hybrid functional calculations with k-point sampling, *J. Chem. Theory Comput.* **18**, 206 (2021).
- [90] W. Hu, L. Lin, and C. Yang, Interpolative separable density fitting decomposition for accelerating hybrid density functional calculations with applications to defects in silicon, *J. Chem. Theory Comput.* **13**, 5420 (2017).
- [91] K. Dong, W. Hu, and L. Lin, Interpolative separable density fitting through centroidal Voronoi tessellation with applications to hybrid functional electronic structure calculations, *J. Chem. Theory Comput.* **14**, 1311 (2018).
- [92] “Project by BMW, BASF, Samsung SDI and Samsung Electronics to enhance sustainable cobalt mining,” <https://www.basf.com/global/en/who-we-are/sustainability/responsible-partnering/cobalt-initiative.html> (accessed October 26, 2020).
- [93] E. A. Olivetti, G. Ceder, G. G. Gaustad, and X. Fu, Lithium-ion battery supply chain considerations: Analysis of potential bottlenecks in critical metals, *Joule* **1**, 229 (2017).
- [94] J. R. Dahn, U. Vonsacken, and C. A. Michal, Structure and electrochemistry of  $\text{Li}_{1-\delta}\text{NiO}_2$  and a new  $\text{Li}_2\text{NiO}_2$  phase with the  $\text{Ni}(\text{OH})_2$  structure, *Solid State Ionics* **44**, 87 (1990).
- [95] T. Ohzuku, A. Ueda, and M. Nagayama, Electrochemistry and structural chemistry of  $\text{LiNiO}_2$  (R3m) for 4 Volt secondary lithium cells, *J. Electrochem. Soc.* **140**, 1862 (1993).
- [96] M. D. Radin, J. C. Thomas, and A. Van der Ven, Order-disorder versus displacive transitions in Jahn-Teller active layered materials, *Phys. Rev. Mater.* **4**, 043601 (2020).
- [97] C. Møller and M. S. Plesset, Note on an approximation treatment for many-electron systems, *Phys. Rev.* **46**, 618 (1934).

- [98] D. Cremer, Møller-Plesset perturbation theory: From small molecule methods to methods for thousands of atoms, *Ltd. WIREs Comput. Mol. Sci.* **1**, 509 (2011).
- [99] G. D. Purvis and R. J. Bartlett, A full coupled-cluster singles and doubles model: The inclusion of disconnected triples, *J. Chem. Phys.* **76**, 1910 (1982).
- [100] R. J. Bartlett and M. Musiał, Coupled-cluster theory in quantum chemistry, *Rev. Mod. Phys.* **79**, 291 (2007).
- [101] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K. L. Chan, PySCF: the Python-based simulations of chemistry framework, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **8**, e1340 (2018).
- [102] Q. Sun, *et al.*, Recent developments in the PySCF program package, *J. Chem. Phys.* **153**, 024109 (2020).
- [103] J. Kim, A. D. Baczewski, T. D. Beaudet, A. Benali, M. C. Bennett, M. A. Berrill, N. S. Blunt, E. J. L. Borda, M. Casula, and D. M. Ceperley, *et al.*, QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids, *J. Phys.: Condens. Matter* **30**, 195901 (2018).
- [104] P. R. C. Kent, *et al.*, QMCPACK: Advances in the development, efficiency, and application of auxiliary field and real-space variational and diffusion quantum Monte Carlo, *J. Chem. Phys.* **152**, 174105 (2020).
- [105] S. Goedecker, M. Teter, and J. Hutter, Separable dual-space Gaussian pseudopotentials, *Phys. Rev. B* **54**, 1703 (1996).
- [106] J. Hutter, New optimization of gth pseudopotentials for PBE, scan, PBE0 functionals. gth pseudopotentials for Hartree-Fock. NLCC pseudopotentials for PBE.
- [107] J. VandeVondele and J. Hutter, Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases, *J. Chem. Phys.* **127**, 114105 (2007).
- [108] X. Xing, X. Li, and L. Lin, Staggered mesh method for correlation energy calculations of solids: Second-order Møller–Plesset perturbation theory, *J. Chem. Theory Comput.* **17**, 4733 (2021).
- [109] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions, *Rev. Mod. Phys.* **68**, 13 (1996).
- [110] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti, Electronic structure calculations with dynamical mean-field theory, *Rev. Mod. Phys.* **78**, 865 (2006).
- [111] K. Held, Electronic structure calculations using dynamical mean field theory, *Adv. Phys.* **56**, 829 (2007).
- [112] D. Vollhardt (2020) p. 11001
- [113] G. Knizia and G. K.-L. Chan, Density matrix embedding: A simple alternative to dynamical mean-field theory, *Phys. Rev. Lett.* **109**, 186404 (2012).
- [114] G. Knizia and G. K. L. Chan, Density matrix embedding: A strong-coupling quantum embedding theory, *J. Chem. Theory Comput.* **9**, 1428 (2013).
- [115] Y. Liu, O. R. Meitei, Z. E. Chin, A. Dutt, M. Tao, T. Van Voorhis, and I. L. Chuang, *ArXiv:2301.01457* (2023).
- [116] N. C. Rubin, J. Lee, and R. Babbush, Compressing many-body fermion operators under unitary constraints, *J. Chem. Theory Comput.* **18**, 1480 (2022).
- [117] F. D. Malone, S. Zhang, and M. A. Morales, Accelerating auxiliary-field quantum Monte Carlo simulations of solids with graphical processing units, *J. Chem. Theory Comput.* **16**, 4286 (2020).
- [118] M. A. Morales and F. D. Malone, Accelerating the convergence of auxiliary-field quantum Monte Carlo in solids with optimized Gaussian basis sets, *J. Chem. Phys.* **153**, 194111 (2020).
- [119] S. Chiesa, D. M. Ceperley, R. M. Martin, and M. Holzmann, Finite-size error in many-body simulations with long-range interactions, *Phys. Rev. Lett.* **97**, 076404 (2006).
- [120] N. Drummond, R. Needs, A. Sorouri, and W. Foulkes, Finite-size errors in continuum quantum Monte Carlo calculations, *Phys. Rev. B* **78**, 125106 (2008).
- [121] S. Azadi and W. Foulkes, Systematic study of finite-size effects in quantum Monte Carlo calculations of real metallic systems, *J. Chem. Phys.* **143**, 102807 (2015).
- [122] M. Holzmann, R. C. Clay III, M. A. Morales, N. M. Tubman, D. M. Ceperley, and C. Pierleoni, Theory of finite size effects for electronic quantum Monte Carlo calculations of liquids and solids, *Phys. Rev. B* **94**, 035126 (2016).
- [123] M. Dagrada, S. Karakuzu, V. L. Vildosola, M. Casula, and S. Sorella, Exact special twist method for quantum Monte Carlo simulations, *Phys. Rev. B* **94**, 245108 (2016).
- [124] T. N. Mihm, T. Schäfer, S. K. Ramadugu, L. Weiler, A. Grüneis, and J. J. Shepherd, A shortcut to the thermodynamic limit for quantum many-body calculations of metals, *Nat. Comput. Sci.* **1**, 801 (2021).
- [125] Y. Yang, V. Gorelov, C. Pierleoni, D. M. Ceperley, and M. Holzmann, *et al.*, Electronic band gaps from quantum Monte Carlo methods, *Phys. Rev. B* **101**, 085115 (2020).
- [126] Y. R. Sanders, D. W. Berry, P. C. S. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, Compilation of fault-tolerant quantum heuristics for combinatorial optimization, *PRX Quantum* **1**, 020312 (2020).
- [127] N. J. DeYonker, K. A. Peterson, G. Steyl, A. K. Wilson, and T. R. Cundari, Quantitative computational thermochemistry of transition metal species, *J. Phys. Chem. A* **111**, 11269 (2007).
- [128] T. J. Lee and P. R. Taylor, A diagnostic for determining the quality of single-reference electron correlation methods, *Int. J. Quantum Chem.* **36**, 199 (1989).
- [129] I. M. Nielsen and C. L. Janssen, Double-substitution-based diagnostics for coupled-cluster and Møller–Plesset perturbation theory, *Chem. Phys. Lett.* **310**, 568 (1999).
- [130] T. J. Lee, Comparison of the T1 and D1 diagnostics for electronic structure theory: A new definition for the open-shell D1 diagnostic, *Chem. Phys. Lett.* **372**, 362 (2003).
- [131] J. Lee, H. Q. Pham, and D. R. Reichman, Twenty years of auxiliary-field quantum Monte Carlo in quantum chemistry: An overview and assessment on main group chemistry and bond-breaking, *J. Chem. Theory Comput.* **18**, 7024 (2022).
- [132] C.-N. Yeh and M. A. Morales, *arXiv preprint ArXiv:2306.04880* (2023).