Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction

Joonho Lee,[1,*,§] Dominic W. Berry,[2,†,§] Craig Gidney,[3] William J. Huggins,[3] Jarrod R. McClean,[3] Nathan Wiebe,[4,5] and Ryan Babbush[3,‡]

[1] *Department of Chemistry, Columbia University, New York, New York, USA*
[2] *Department of Physics and Astronomy, Macquarie University, Sydney, NSW, Australia*
[3] *Google Quantum AI, Venice, California, USA*
[4] *Department of Physics, University of Washington, Seattle, Washington, USA*
[5] *Pacific Northwest National Laboratory, Richland, Washington, USA*

(Received 12 December 2020; revised 7 April 2021; accepted 24 May 2021; published 8 July 2021)

We describe quantum circuits with only $\widetilde{\mathcal{O}}(N)$ Toffoli complexity that block encode the spectra of quantum chemistry Hamiltonians in a basis of N arbitrary (e.g., molecular) orbitals. With $\mathcal{O}(\lambda/\epsilon)$ repetitions of these circuits one can use phase estimation to sample in the molecular eigenbasis, where λ is the 1-norm of Hamiltonian coefficients and ϵ is the target precision. This is the lowest complexity shown for quantum computations of chemistry within an arbitrary basis. Furthermore, up to logarithmic factors, this matches the scaling of the most efficient prior block encodings that can work only with orthogonal-basis functions diagonalizing the Coloumb operator (e.g., the plane-wave dual basis). Our key insight is to factorize the Hamiltonian using a method known as tensor hypercontraction (THC) and then to transform the Coulomb operator into an isospectral diagonal form with a nonorthogonal basis defined by the THC factors. We then use qubitization to simulate the nonorthogonal THC Hamiltonian, in a fashion that avoids most complications of the nonorthogonal basis. We also reanalyze and reduce the cost of several of the best prior algorithms for these simulations in order to facilitate a clear comparison to the present work. In addition to having lower asymptotic scaling space-time volume, compilation of our algorithm for challenging finite-sized molecules such as FeMoCo reveals that our method requires the least fault-tolerant resources of any known approach. By laying out and optimizing the surface-code resources required of our approach we show that FeMoCo can be simulated using about four million physical qubits and under 4 days of runtime, assuming 1-μs cycle times and physical gate-error rates no worse than 0.1%.

DOI: 10.1103/PRXQuantum.2.030305

I. INTRODUCTION

A. Background

The quantum computation of quantum chemistry is commonly regarded as one of the most promising applications of quantum computers [1–3]. This is because there are many applications of quantum chemistry that pertain to the development of practical technologies and, for at least some of these applications, quantum algorithms appear to provide an exponential scaling advantage relative to the best known classical approaches [4]. Specifically, most algorithmic work in this area focuses on the construction of quantum circuits that precisely sample in the eigenbasis of the electronic Hamiltonian. This enables the very precise preparation of any electronic eigenstate that can be reasonably approximated with a classically tractable theory, such as the ground states of many molecules. The ability to arbitrarily refine the accuracy of an approximation to molecular eigenstates is valuable because high precision is often required to predict important properties of these systems, such as the rates of chemical reactions, excitation energies, barrier heights, and noncovalent molecular interaction energies [5]. The "holy grail" of quantum chemistry is to have a generally applicable electronic structure method that yields relative energies with errors less than 1.6 millihartrees (so-called "chemical accuracy"[6]), which is still a long-standing challenge in the field.

*linusjoonho@gmail.com
†dominic.berry@mq.edu.au
‡ryanbabbush@gmail.com
§These authors contributed equally.

The most efficient rigorous approaches to this problem in quantum computing use the quantum phase estimation algorithm [7,8] to sample in the eigenbasis of the molecular Hamiltonian by measuring the phase accumulated on an initial state under the application of a unitary operator with eigenvalues that are related to those of the electronic Hamiltonian. The electronic Hamiltonian in an arbitrary second-quantized basis can be expressed as

$$H = T + V, \quad T = \sum_{\sigma\in\{\uparrow,\downarrow\}} \sum_{p,q=1}^{N/2} T_{pq} a^\dagger_{p,\sigma} a_{q,\sigma}$$

$$V = \frac{1}{2} \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}} \sum_{p,q,r,s=1}^{N/2} V_{pqrs} a^\dagger_{p,\alpha} a_{q,\alpha} a^\dagger_{r,\beta} a_{s,\beta}, \qquad (1)$$

where N is the number of spin-orbital basis functions used to discretize the Hamiltonian, T_{pq} represents a matrix element of the effective one-body operator, V_{pqrs} is a matrix element of the two-body Coulomb operator, and $a^\dagger_{p,\sigma}$ and $a_{p,\sigma}$ are fermionic raising and lowering operators for the p^{th} single-particle orbital with spin σ. Note that unlike the convention in Ref. [9], we do not absorb the factor of "1/2" into the V_{pqrs} (this is consistent with conventions of Ref. [10] to avoid confusion). V_{pqrs} is the central tensor in this work, which is defined as

$$V_{pqrs} = \int\int d\mathbf{r}_1\, d\mathbf{r}_2 \frac{\phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \qquad (2)$$

where $\{\phi_i\}_{i=1}^{N/2}$ is a set of real-valued single-particle spatial-orbital basis functions. We note that the matrix element T_{pq} is defined as

$$T_{pq} = h_{pq} - \frac{1}{2} \sum_{r=1}^{N/2} V_{prrs}, \qquad (3)$$

with h_{pq} being a matrix element of the one-body operator that includes the kinetic energy operator and electron-nuclear Coulomb operator.

While most work has focused on encoding the eigenspectra of this Hamiltonian in a unitary for phase estimation by synthesizing the time-evolution operator e^{-iHt} [11], several recent papers (including this one) have demonstrated increased efficiency by instead synthesizing a qubitized quantum walk [12] with eigenvalues proportional to $e^{\pm i \arccos(H/\lambda)}$, where λ is a parameter related to the norm of the Hamiltonian. By repeating this quantum walk a number of times scaling as $\mathcal{O}(\lambda/\epsilon)$ one is able to prepare eigenstates of H and estimate the associated eigenvalue such that the error in the estimated eigenvalue is no greater than ϵ [13,14].

Throughout this paper, all discussion of an error ϵ in the eigenspectra refers to error relative to the finite-sized basis of N spin orbitals that we use to discretize our system. The finite-sized basis we choose also introduces a discretization error with respect to the representation of the system in the continuum limit. While this basis set discretization error can be asymptotically refined as $\epsilon = \mathcal{O}(1/N)$ for common choices of the single-particle basis such as plane waves or molecular orbitals, the precise constant factors in this scaling depend on the particular basis functions used [which also determine the coefficients T_{pq} and V_{pqrs} in Eq. (1)]. In the last few years, several papers have demonstrated quantum algorithms for chemistry with reduced complexity [10,15–19]. Some algorithms achieved a lower scaling by performing simulation in a basis [15,19,20] that diagonalizes the Coulomb operator V such that $V_{pqrs} = 0$ unless $p = q$ and $r = s$. However, those representations typically require a significantly larger N in order to model molecular systems within target accuracy of the continuum limit. While such representations might prove practical for molecules when used in first quantization [21,22], in second quantization the requirement of a significantly larger basis translates into needing significantly more qubits and thus those approaches seem impractical for simulating molecular systems (as opposed to, e.g., crystalline solid-state systems).

Given this, there is a need for efficient quantum algorithms that are compatible with arbitrary basis functions and not limited to those that diagonalize the Coulomb operator. Such quantum algorithms can exploit the molecular-orbital basis directly and thereby significantly reduce the required number of basis functions for a target accuracy towards the continuum limit. There are only three prior papers that have fully determined the cost of performing such a quantum algorithm for chemistry within an error-correcting code [9,10,23].

The first of these papers by Reiher *et al.* [23] deployed a Trotter-based approach to study the quantum simulation of the FeMo cofactor of the nitrogenase enzyme, also known as "FeMoCo" (Fe_7MoS_9C), a molecule that is important for understanding the mechanism of biological nitrogen fixation [24]. The algorithm used for that work had T-gate complexity scaling as approximately $\mathcal{O}(N^2 S/\epsilon^{3/2})$, where S is the Hamiltonian sparsity [$S = \mathcal{O}(N^4)$ in an arbitrary basis but with a sufficiently large, localized basis sometimes $S = \mathcal{O}(N^2)$ [25]]. The work of Reiher *et al.* required more than 10^{14} T gates to simulate an active-space model of FeMoCo. These papers focus on counting (and reducing) the required number of T gates (or Toffoli gates) because within practical error-correcting codes such as the surface code [26], these gates require significantly more time to implement than any other gate and also require a very large number of physical qubits for their implementation. If implemented in the surface code using gates with 10^{-3} error rates, the most efficient protocols for implementing T gates require roughly 15 qubitseconds [27,28] of space-time volume. At those rates, just distilling the

magic states needed for the Reiher *et al.* FeMoCo calculation would require four million qubitdecades (e.g., one million qubits running for 4 decades or one billion qubits running for 2 weeks).

This work and the two papers [9,10] employ a qubitization-based approach [12], and improve considerably over the Trotter-based methods of Ref. [23]. The primary difference between these more recent algorithms is that each adapts qubitization to a different type of tensor factorization of the Coulomb operator. One of these papers, which suggested combining qubitization with tensor factorizations of the Coulomb operator, was by Berry *et al.* [9]. That work presents two approaches (one with lower asymptotic complexity, and one with lower gate counts for molecules such as the FeMoCo active space). The approach with the lower finite-sized gate counts is based on using qubitization to exploit sparsity structures in the electronic Hamiltonian. We discuss that method (referred to throughout as the "sparse" method) and its cost in Appendix A and show that in Ref. [9], an error led us to overestimate the complexity. We also recalculate the number of nonzero entries that must be retained in the Hamiltonian, as well as making a number of minor improvements to the algorithm, leading to the sparse approach improving over the Toffoli complexity of the Reiher *et al.* result by roughly a factor of 1100× rather than the factor of 700× reported in Refs. [9,10]. In terms of asymptotic complexity this approach has Toffoli complexity $\widetilde{\mathcal{O}}[(N+\sqrt{S})\lambda/\epsilon]$ [29] and space complexity $\widetilde{\mathcal{O}}(N+\sqrt{S})$, where λ is a parameter related to the Hamiltonian norm (we discuss λ further towards the end of the Introduction).

The lower scaling method (but with slightly higher constant factors) presented by Berry *et al.* [9] combines qubitization with the first step of a tensor factorization originally discussed for quantum computing in Ref. [30]. Here we refer to that representation as the "single low rank" factorization or simply "SF." The SF uses an eigendecomposition of the two-electron integral tensor, similar to the Cholesky decomposition [31] or density fitting [32–34] as commonly used in electronic structure literature. The single low rank factorization algorithm obtains Toffoli and space complexities of $\widetilde{\mathcal{O}}(N^{3/2}\lambda/\epsilon)$ and $\widetilde{\mathcal{O}}(N^{3/2})$, respectively. We discuss that method and its cost in Appendix B.

The most recent paper by von Burg *et al.* [10] adapts qubitization to a second tensor factorization that occurs on top of the SF used by Berry *et al.* This second tensor factorization was also described for quantum computing in Ref. [30] and corresponds to a diagonalization of the squared one-body operators, which dates back to Ref. [35] in the electronic structure literature. We refer to this as the "double low rank" factorization or simply "DF." We review the method and its cost in Appendix C. Von Burg *et al.* claim that their DF algorithm gives more than an order of magnitude complexity improvement relative to the SF algorithm of Berry *et al.*; however, the numbers they compare to are actually those for the sparse algorithm of Berry *et al.* (which is more efficient than the SF method for the molecules they consider). Furthermore, for the more accurate FeMoCo Hamiltonian introduced by Li *et al.* [36], the sparse algorithm requires half as many logical qubits and 2/3 the number of Toffoli gates as the DF algorithm (so in that case, the sparse algorithm is considerably cheaper). The Toffoli complexity of the DF approach is $\widetilde{\mathcal{O}}(N\lambda\sqrt{\Xi}/\epsilon)$ with space complexity $\widetilde{\mathcal{O}}(N\sqrt{\Xi})$, where Ξ is the average rank of the second tensor factorization discussed in Ref. [30]. In most cases (including the regimes that are usually of interest for small quantum computers) Ξ will scale around $\mathcal{O}(N)$, giving similar scaling as the SF method. There is some evidence that when N is growing towards the thermodynamic limit (the number of atoms is very large and increasing while fixing the ratio of spin orbitals to atoms) Ξ can scale as $\mathcal{O}(1)$ [35,37]; however, this is not the case in our numerics on hydrogen systems just up to 100 hydrogen atoms, which reveal scaling of $\mathcal{O}(N)$.

There are several other commonalities between the algorithms of Refs. [9,10] and the ones developed here, which are worth discussing before introducing the main techniques of this paper. In addition to combining qubitization [12] and tensor factorizations, all three works involve the use of unary iteration, QROM, coherent alias sampling, and qubitized phase estimation bounds from Ref. [16] as well as a more advanced version of QROM with fanout developed in Ref. [38] and then optimized for use in our context [9] (where it is referred to as "QROAM"—a portmanteau of QROM and QRAM). Using these tools one can often construct algorithms to realize qubitized quantum walks with gate complexity that scales as $\mathcal{O}(\sqrt{\Gamma})$ where one requires $\mathcal{O}(\sqrt{\Gamma})$ ancilla, and Γ is the amount of information required to specify the Hamiltonian within a particular tensor factorization. Then, by performing phase estimation on the resultant qubitized quantum walks, one is able to sample in the Hamiltonian eigenbasis with Toffoli complexity $\widetilde{\mathcal{O}}(\sqrt{\Gamma}\lambda/\epsilon)$.

For the sparse algorithm of Ref. [9], no tensor factorization is employed and thus the Hamiltonian [the same as in Eq. (1)] contains a number of integrals scaling as $\Gamma=\widetilde{\mathcal{O}}(S)$. Thus, accounting for a complexity proportional to N required to perform controlled operations, we end up with an algorithm having Toffoli complexity $\widetilde{\mathcal{O}}[(N+\sqrt{S})\lambda/\epsilon]$ and space complexity $\widetilde{\mathcal{O}}(N+\sqrt{S})$. For the SF algorithm of Ref. [9] we rely on the single-low-rank factorized Hamiltonian shown in Appendix B, which is specified with an amount of information scaling as $\Gamma=\widetilde{\mathcal{O}}(N^3)$. This leads to an algorithm with Toffoli complexity $\widetilde{\mathcal{O}}(N^{3/2}\lambda/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N^{3/2})$. Finally, the DF algorithm of Ref. [10] relies on the factorization shown in Appendix C, which compresses the Hamiltonian to $\Gamma=\widetilde{\mathcal{O}}(N^2\Xi)$ pieces of information. Accordingly, this

approach yields gate complexity $\widetilde{\mathcal{O}}(N\lambda\sqrt{\Xi}/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N\sqrt{\Xi})$.

The last element of the complexity of these algorithms that we should discuss is the scaling of λ. The quantity λ is a type of norm of the Hamiltonian that is being simulated, and depends on how it is expressed. For example, for the "sparse" algorithm of Ref. [9] the value of λ is simply the sum of the absolute value of Hamiltonian coefficients (i.e., the 1-norm) appearing in Eq. (1). Numerical studies reveal that λ usually scales somewhere in between $\mathcal{O}(N)$ and $\mathcal{O}(N^3)$ depending on details of the particular system as well as the algorithm and how one is increasing N (e.g., the scaling is lower if N is growing because the number of basis functions per atom is fixed but the number of atoms is increasing and the scaling is higher if N is growing because the number of atoms is fixed but the basis size is growing). The largest λ tends to be that associated with the SF used in Ref. [9], followed by the sparse representation used in Ref. [9], with the smallest λ of three approaches discussed so far being the λ associated with the DF used in Ref. [10]. In most contexts the main advantage of the DF algorithm relative to the SF algorithm is not a substantially smaller Γ (or less complex quantum walk) but rather, a smaller λ. Thus, it is critical to consider how the algorithmic choices that one makes will affect the value of λ.

B. Overview of results

We now discuss the main results of this paper. We present a qubitization-based algorithm that results from exploiting structure in the molecular Hamiltonian that emerges from a tensor factorization of the Coulomb operator known as tensor hypercontraction (THC) [39–41]. THC is a very compact representation for the Hamiltonian, which gives $\Gamma = \widetilde{\mathcal{O}}(N^2)$ regardless of how one increases N. It is relatively straightforward to apply qubitization directly to the THC representation and a method for that is presented in Appendix E. This results in an approach with Toffoli complexity $\widetilde{\mathcal{O}}(N\lambda/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N)$. However, it turns out that directly applying qubitization to the THC representation is not particularly efficient because this causes λ to become even larger than the single low rank λ of Ref. [9] (which was already orders of magnitude larger than the double low rank λ of Ref. [10] for systems like FeMoCo). To remedy this, we discuss a different approach to utilizing the THC representation of the Hamiltonian.

We show that one can use the THC tensors to define a larger orbital basis with exactly the same resolution as the original basis while diagonalizing the Coulomb operator. This form of the Hamiltonian would then be amenable to simulation using the efficient strategies of Refs. [16–18] except for the fact that the orbitals are nonorthogonal. However, by separately rotating into this nonorthogonal basis before operating on each tensor factor of the Hamiltonian we are able to avoid many of the complications that would usually arise from simulating a nonorthogonal operator. To achieve this, we use a strategy for storing and implementing nonorthogonal Givens rotation-based orbital-basis transformations [42] with rotation angles loaded from QROM that is similar to techniques developed for qubitizing orthogonal-basis rotations in Ref. [10]. As these rotations add significant cost to the method, we also introduce a strategy for coarse graining the angles of the basis transformation that allows us to precisely control a trade-off between the complexity of these rotations and the accuracy of the Hamiltonian (a similar technique would also reduce the cost of the algorithm by von Burg *et al.*). Ultimately, our method does not require additional system qubits (beyond those for QROM and other minor ancillary functions) and results in asymptotic Toffoli complexity $\widetilde{\mathcal{O}}(N\lambda/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N)$, while essentially matching the small λ values obtained in the double low rank algorithm.

As can be seen in Table I our algorithm improves over the space-time volume of the approaches in Refs. [9,10] by a factor of about $\widetilde{\mathcal{O}}(N)$ in most contexts. In fact, the scaling is often even better than that due to the lower scaling of λ associated with our representation, as can be seen from numerics on hydrogen systems in Table II. Surprisingly, the λ value associated with our algorithm sometimes scales even less than $\mathcal{O}(N^2)$, which is the scaling of the λ for the lowest scaling qubitization approach requiring orthogonal-basis functions that diagonalize the Coulomb operator [16].

In addition to having the best asymptotic scaling compared to prior approaches, our algorithm also outperforms the finite space-time volume for all molecules studied here including hydrogen chains and FeMoCo. We study active-space models of FeMoCo proposed by Reiher *et al.* [23] as well as by Li *et al.* [36]. The Reiher Hamiltonian was found to be qualitatively incorrect in describing the ground state of FeMoCo as it does not capture the open-shell nature of the system [36]. The Li Hamiltonian was then proposed and shown to capture the open-shell nature of the ground state properly [36]. We focus on the FeMoCo Hamiltonians primarily because they are regarded as a standard benchmark for quantum computing. For the FeMoCo Hamiltonians of Reiher *et al.* and Li *et al.* we find a reduction in space-time volume of about $3\times$ and $6\times$ (respectively) compared to Ref. [10]. The results for FeMoCo are summarized in Table III.

We also carefully analyze the surface-code resources required to simulate the FeMoCo Hamiltonian of Li *et al.* [36] using our THC approach. Rather than just focus on the cost of distillation, we fully lay out the surface-code computation in space time and optimize resource usage. We determine that the computation could execute using approximately four million physical qubits and run

TABLE I. History of the lowest asymptotic scaling quantum algorithms for simulating quantum chemistry in an arbitrary (e.g., molecular orbital) basis. N is the number of arbitrary orbital basis functions, η is the number of electrons (relevant only in first-quantized simulations), and ϵ is the target precision to which we estimate the Hamiltonian eigenvalues using phase estimation. S is the sparsity of the electronic Hamiltonian; usually $S = \mathcal{O}(N^4)$ when using an arbitrary basis but sometimes the scaling can be lower. The λ parameters (discussed extensively in this paper) are roughly the 1-norm of the Coulomb operator associated with the representation in which we simulate the system. In general, we expect that $\mathcal{O}(\lambda_\zeta) \leq \mathcal{O}(\lambda_{\mathrm{DF}}) \leq \mathcal{O}(\lambda_V) \leq \mathcal{O}(\lambda_{\mathrm{SF}})$ but the precise scaling is difficult to report since it depends on the specific molecule and how N is growing. Roughly, the λ values have scaling that is typically between $\mathcal{O}(N)$ and $\mathcal{O}(N^3)$. Here, Ξ is the average rank of the second-tensor factorization discussed in Motta *et al.* [30], which is also important for the complexity of the work of von Burg *et al.* [10]. In general (for example, when scaling towards the continuum limit) we would expect that $\Xi = \mathcal{O}(N)$. But for large systems that are growing because we are adding more atoms while keeping the basis to atom ratio fixed, Ξ can be smaller; for the hydrogen chains studied in this work we observe $\Xi = \mathcal{O}(N)$ up to 100 hydrogen atoms. See Table II for a better sense of how algorithms that depend on S, λ, and Ξ parameters scale for hydrogen benchmarks. The work of Motta *et al.* [30] does not determine the scaling of the Trotter error for the Trotter steps compiled therein and so this table assumes (rather speculatively) that the Trotter-error scaling for those Trotter steps is the same as the Trotter-error scaling for the Trotter steps of Reiher *et al.* [23]. This table omits methods that require special basis functions or non-Galerkin representations. All such representations are less compact for molecules compared to molecular orbitals and thus require more qubits to reach the same level of accuracy. Notable examples include the grid bases used in Ref. [21,43], the discontinuous Galerkin techniques of Ref. [19], the plane waves required by Refs. [10,17,22] and the basis sets diagonalizing the Coulomb operator required by Refs. [15,18,42].

Year	Reference	Primary algorithmic innovation	Space complexity	Toffoli or T complexity
2005	Aspuru-Guzik *et al.* [4]	First algorithm (no compilation or bounds)	$\mathcal{O}(N)$	$\mathcal{O}[\mathrm{poly}(N/\epsilon)]$
2010	Whitfield *et al.* [11]	First compilation (no Trotter bounds)	$\mathcal{O}(N)$	$\mathcal{O}[\mathrm{poly}(N/\epsilon)]$
2012	Seeley *et al.* [44]	Use of Bravyi-Kitaev transformation	$\mathcal{O}(N)$	$\mathcal{O}[\mathrm{poly}(N/\epsilon)]$
2013	Wecker *et al.* [45]	First chemistry-specific Trotter bounds	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^{10}/\epsilon^{3/2})$
2013	Toloui *et al.* [46]	Use of first quantization	$\mathcal{O}(\eta \log N)$	$\widetilde{\mathcal{O}}(\eta^2 N^8/\epsilon^{3/2})$
2014	Hastings *et al.* [47]	Better compilation and multiresolution Trotter	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^8/\epsilon^{3/2})$
2014	Poulin *et al.* [48]	Tighter Trotter bounds and ordering	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^6/\epsilon^{3/2})$
2014	McClean *et al.* [25]	Exploiting Hamiltonian sparsity with Trotter	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^4 S/\epsilon^{3/2})$
2014	Babbush *et al.* [49]	Tighter system-specific Trotter bounds	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^2 S/\epsilon^{3/2})$
2015	Babbush *et al.* [50]	Use of Taylor series (database method)	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^4 \lambda_V/\epsilon)$
2015	Babbush *et al.* [50]	Use of Taylor series (on-the-fly method)	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^5/\epsilon)$
2015	Babbush *et al.* [51]	Use of Taylor series with first quantization	$\mathcal{O}(\eta \log N)$	$\widetilde{\mathcal{O}}(\eta^2 N^3/\epsilon)$
2016	Reiher *et al.* [23]	First T count and tighter Trotter bounds	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^2 S/\epsilon^{3/2})$
2018	Motta *et al.* [30]	Use of low-rank factorization with Trotter	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^4 \Xi/\epsilon^{3/2})$
2018	Campbell [52]	Use of randomized compiling with Trotter	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(\lambda_V^2/\epsilon^2)$
2019	Berry *et al.* [9]	Use of qubitization (sparse method)	$\widetilde{\mathcal{O}}(N + \sqrt{S})$	$\widetilde{\mathcal{O}}[(N + \sqrt{S})\lambda_V/\epsilon]$
2019	Berry *et al.* [9]	Use of qubitization (single factorization)	$\widetilde{\mathcal{O}}(N^{3/2})$	$\widetilde{\mathcal{O}}(N^{3/2}\lambda_{\mathrm{SF}}/\epsilon)$
2019	Kivlichan *et al.* [53]	Better randomized compiled phase estimation	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(\lambda_V^2/\epsilon^2)$
2020	von Burg *et al.* [10]	Use of qubitization (double factorization)	$\widetilde{\mathcal{O}}(N\sqrt{\Xi})$	$\widetilde{\mathcal{O}}(N\lambda_{\mathrm{DF}}\sqrt{\Xi}/\epsilon)$
2020	Present work	Use of tensor hypercontraction	$\widetilde{\mathcal{O}}(N)$	$\widetilde{\mathcal{O}}(N\lambda_\zeta/\epsilon)$

in under 4 days, assuming surface-code cycle times of 1 μs and gate-error rates of about 0.1%. We find that if error rates are reduced to 0.01% that the computation can complete using about one million physical qubits in under 2 days.

Finally, in Appendix D we also provide a detailed analysis of the error in phase estimation when combined with stochastic approximations such as qDRIFT [52]. We find that while existing analyses naturally lead to reasonable region estimates for the estimated phase, the distribution of errors for the phase estimation procedure can have fat tails. These tails manifest in Table III in the form of impractically large numbers of Toffoli gates needed to perform phase estimation on both benchmark examples for FeMoCo. Specifically, we find that the cost of performing qDRIFT unmodified is nearly 18 orders of magnitude greater than the cost of performing the optimized form of qubitization that we consider for these benchmark molecules. If we require only that an estimate within an α-confidence region is reported, then the costs can be reduced by 12 orders of magnitude. Alternatively, quantifying the performance according to the Hodges-Lehmann estimator provides a further order of magnitude improvement. This illustrates a poorly appreciated fact in the quantum simulation literature: for simulation techniques like qDRIFT that have high failure probability, we need to carefully specify the error metric used for the eigenphase yielded by the algorithm because some error metrics (such as the mean square error) can be much harder to minimize than others.

TABLE II. Empirical complexity of algorithms that have scalings obscured by λ values in Table I for two benchmark chemical series. N is the number of spin-orbital basis functions and ϵ is the target precision to which we aim to realize phase estimation. This table summarizes the findings of numerics discussed in Sec. IV, which reveal the scaling with N that one might observe in practice for these algorithms when the system size is growing towards either the continuum or thermodynamic limits. For the continuum limit we focus on H_4 Hamiltonians with each hydrogen placed on the corners of a square plaquette with side length of 2.0 Bohr radii. We then increase the number of molecular orbitals used to represent the system and determine the scaling of the associated algorithms. For scalings towards the thermodynamic limit we fix the ratio of basis functions to atoms and increase the number of hydrogens in a one-dimensional hydrogen chain, with atom spacings again at 1.4 Bohr radii. For more information on these calculations, see Sec. IV. The justification for the above scalings for qDRIFT in the case where ϵ is the root-mean-square error is given in Appendix D.

	H_4 continuum limit		Hydrogen chain thermodynamic limit	
Algorithm	Space complexity	Toffoli complexity	Space complexity	Toffoli complexity
Babbush *et al.* [50] (Taylor series database)	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^{7.1}/\epsilon)$	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^{5.3}/\epsilon)$
Campbell and Kivlichan *et al.* [52,53] (qDRIFT)	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^{6.2}/\epsilon^2)$	$\mathcal{O}(N)$	$\widetilde{\mathcal{O}}(N^{2.5}/\epsilon^2)$
Berry *et al.* [9] (single factorization)	$\widetilde{\mathcal{O}}(N^{1.5})$	$\widetilde{\mathcal{O}}(N^{3.8}/\epsilon)$	$\widetilde{\mathcal{O}}(N^{1.5})$ & $\widetilde{\mathcal{O}}(N^{4.5}/\epsilon)$	
Berry *et al.* [9] (sparse)	$\widetilde{\mathcal{O}}(N^{1.9})$	$\widetilde{\mathcal{O}}(N^{5.0}/\epsilon)$	$\widetilde{\mathcal{O}}(N)$	$\widetilde{\mathcal{O}}(N^{2.3}/\epsilon)$
von Burg *et al.* [10] (double factorization)	$\widetilde{\mathcal{O}}(N^{1.5})$	$\widetilde{\mathcal{O}}(N^{3.8}/\epsilon)$	$\widetilde{\mathcal{O}}(N^{1.5})$	$\widetilde{\mathcal{O}}(N^{3.4}/\epsilon)$
This work (tensor hypercontraction)	$\widetilde{\mathcal{O}}(N)$	$\widetilde{\mathcal{O}}(N^{3.1}/\epsilon)$	$\widetilde{\mathcal{O}}(N)$	$\widetilde{\mathcal{O}}(N^{2.1}/\epsilon)$

C. Paper organization

Our paper is organized as follows. In Sec. II we describe the THC factorization of the electronic Hamiltonian. We give some background on how the factorization can be obtained, and discuss how it is shown to scale. We outline how one can directly apply qubitization to the THC Hamiltonian, although with large constant factors in the scaling. Then, we introduce an elegant use of the THC Hamiltonian that corresponds to a diagonal Coulomb operator in a larger nonorthogonal basis.

In Sec. III we give a complete description of how qubitization can be combined with our (diagonal and nonorthogonal) THC representation to give the most efficient known quantum algorithm for simulating electronic structure in an arbitrary basis. We compile our

TABLE III. Here we report the finite resources required for various recent algorithms to quantum phase estimate two FeMoCo active spaces to within chemical accuracy. The reason for focusing on two different active spaces is that several papers benchmarked their methods for the FeMoCo Hamiltonian of Reiher *et al.* [23], but the work of Li *et al.* [36] later showed that there is almost no open-shell character in the ground state of the active-space model by Reiher *et al.* and proposed a slightly larger active space with important open-shell character. We report the lowest known Toffoli and logical qubit counts of prior algorithms, consistent with our accounting of these costs in Appendices A, B, C, and D. Note that Appendices A and B show how to reduce the Toffoli counts reported for the algorithms of Berry *et al.* by factors of roughly 13× for the Reiher Hamiltonian and 8× for the Li Hamiltonian (single factorization) and 3× for the Reiher Hamiltonian and 2× for the Li Hamiltonian (sparse), respectively, and we use these more optimized resource estimates here. Likewise, the resource estimates reported for von Burg *et al.* [10] are slightly different from what is reported in their paper because we use a different criterion for determining the truncation thresholds (which, as we discuss in this paper, is a justified, but a tighter criterion than what is used in their work). There were some errors with the algorithm of von Burg *et al.* as presented in their work, which we correct here in Appendix C. Finally, because the Trotter-based methods from Reiher *et al.* are more naturally bottlenecked by T gates than by Toffoli gates, we report half the number of T gates that would be required since a Toffoli gate is roughly twice the cost of a T gate within the surface code [54]. We use 10 bits for state preparation in the methods of Berry *et al.*, von Burg *et al.*, and tensor hypercontraction. A total of 16 bits for the Reiher Hamiltonian and 20 bits for the Li Hamiltonian are used for rotations in tensor hypercontraction and double factorization.

	Reiher *et al.* FeMoCo [23]		Li *et al.* FeMoCo [36]	
Algorithm	Logical qubits	Toffoli count	Logical qubits	Toffoli count
Reiher *et al.* [23] (Trotter)	111	5.0×10^{13}	—	—
Campbell and Kivlichan *et al.* [52,53] (qDRIFT) (D16), (D17)	288	5.2×10^{27}	328	1.8×10^{28}
qDRIFT with 95% confidence interval (D34)	270	1.9×10^{16}	310	1.0×10^{16}
Berry *et al.* [9] (single factorization) (B16), (B17)	3,320	9.5×10^{10}	3,628	1.2×10^{11}
Berry *et al.* [9] (sparse) (A17), (A18)	2,190	8.8×10^{10}	2,489	4.4×10^{10}
von Burg *et al.* [10] (double factorization) (C39), (C40)	3,725	1.0×10^{10}	6,404	6.4×10^{10}
This work (tensor hypercontraction) (44), (46)	2,142	5.3×10^{9}	2,196	3.2×10^{10}

approach all the way to Clifford + Toffoli gates and report the constant factors in the leading-order scaling for both the total Toffoli complexity and total ancilla required.

In Sec. IV we analyze the finite resources required to perform the algorithm of Sec. III for the simulation of several real systems: FeMoCo and hydrogen chains of various sizes. These numerics demonstrate the effectiveness of the THC representation, help to elucidate certain aspects of the scaling of our approach, and allow us to compare the cost of our approach to prior methods. The second part of this section discusses the layout of the Li FeMoCo Hamiltonian simulation in the surface code. Our analysis considers the cost of routing, distillation, and analyzes how many physical qubits are required at all points in the computation. Finally, we reflect on the significance of these results and suggest future directions for research in Sec. V.

Appendices A, B, and C contain extensive and self-contained descriptions of the prior methods from Ref. [9] and Ref. [10], adapted to the notation and conventions of this paper, and in some cases with improved bounds on the complexity of those methods. Furthermore, they also contain numerical details associated with each method that are not presented in the main text. Appendix D covers the randomized compiled methods of Ref. [52], which are not particularly related to the other methods here but also have scaling that depends on λ so we are able to analyze the exact resources required by that approach with numerics that are already available to us. Then, in Appendix E we perform a detailed analysis of how one can directly use the THC representation without projecting into the nonorthgonal basis. This leads to an exceptionally simple-to-understand algorithm (more straightforward than the other approach of this paper or those of Refs. [9,10]) but with worse constant factors in the scaling compared to the primary approach of this paper due to a larger value of λ. The remaining two appendices discuss technical details pertaining to the implementation of important circuit primitives used throughout this work. In Appendix F we discuss a technique and costings for computing contiguous registers. Finally, in Appendix G we describe and analyze the cost of modifying the QROM procedure [9,38] to output two registers at a time.

II. TENSOR HYPERCONTRACTION REPRESENTATIONS FOR QUANTUM SIMULATION

A. The standard tensor hypercontraction representation

The THC representation [39–41] of the electronic Hamiltonian factorizes the Coulomb operator V from Eq. (1) as

$$V \approx G = \frac{1}{2} \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}} \sum_{p,q,r,s=1}^{N/2} G_{pqrs} a^\dagger_{p,\alpha} a_{q,\alpha} a^\dagger_{r,\beta} a_{s,\beta}$$

$$G_{pqrs} = \sum_{\mu,\nu=1}^{M} \chi_p^{(\mu)} \chi_q^{(\mu)} \zeta_{\mu\nu} \chi_r^{(\nu)} \chi_s^{(\nu)}, \tag{4}$$

where $\chi_p^{(\mu)}$ and $\zeta_{\mu\nu} = \zeta_{\nu\mu}$ are real scalars obtained via existing algorithms (*vide infra*) and empirical studies [39–41,55–74] suggest that the approximation G encodes low-energy eigenvalues of V to within error ϵ_{THC} so long as

$$M = \mathcal{O}\left[N \,\mathrm{polylog}\left(1/\epsilon_{\mathrm{THC}}\right)\right]. \tag{5}$$

The ϵ_{THC} inside of Eq. (5) is an error in the energy *per atom*. Thus, if we instead intend ϵ_{THC} to represent a fixed additive error in the total energy then when N is growing towards the thermodynamic limit (i.e., if we are fixing the ratio of basis functions to atoms and adding more atoms) $M = \mathcal{O}[N \,\mathrm{polylog}(N/\epsilon_{\mathrm{THC}})]$, but when N is growing towards the continuum limit (i.e., we are fixing the number of atoms and adding more basis functions) $M = \mathcal{O}[N \,\mathrm{polylog}(1/\epsilon_{\mathrm{THC}})]$.

While the behavior that the THC rank M should scale near linearly in N has been observed in many contexts [39–41,55–74], the most rigorous result establishing the poly-logarithmic dependence on ϵ_{THC} comes from Ref. [74]. Specifically, the work of Ref. [74] employs perturbation theory to empirically evaluate the scaling of the scaling of M in Eq. (4), ultimately concluding that (within second-order perturbation theory) $M = \mathcal{O}[N \,\mathrm{polylog}(1/\epsilon_{\mathrm{THC}})]$. The leading order constant for this scaling is both system and basis dependent. Nevertheless, generally speaking, to achieve 50 μhartree per atom (a widely accepted accuracy threshold for approximating V to within chemical accuracy for molecular simulations), one needs at least as many THC basis functions as density-fitting basis functions when using $\mathcal{O}(N^4)$ algorithms [69] or slightly more with efficient $\mathcal{O}(N^3)$ algorithms [73]. With the $\mathcal{O}(N^4)$ algorithm in Ref. [69], one needs M equal to between $2N$ and $3N$ for a broad class of chemical problems to achieve chemical accuracy.

The structure of the tensor factorization in Eq. (4) is rather different from either the SF or the DF used in Refs. [9,10]. For instance, unlike with the low-rank decompositions, it is unclear how one might combine the THC factorization with reduced scaling product formulas for time evolution [30] or better methods of performing energy measurements for variational algorithms [75]. Similar to how the work of Refs. [9,10] combines qubitization with the tensor factorizations described in Appendices B and C, here we discuss how the THC factorization leads to an advantage when combined with qubitization; however, our approach will require different qubitization oracles (and thus different algorithms) from those in Refs. [9,10].

B. Numerical computation of the tensor hypercontraction factorization

The problem of obtaining the factorization of Eq. (4) is often numerically ill conditioned and has been the subject of research for many years in quantum chemistry [39–41, 55–74]. We write the THC factorization problem as an L_2 norm minimization problem where we seek minimizers, χ and ζ, for

$$\mathcal{L} = \|V - G\|_2 = \sum_{pqrs} \left| V_{pqrs} - G_{pqrs} \right|^2 . \tag{6}$$

This objective function $\mathcal{L}$ is quartic in χ and linear in ζ. This generally exhibits multiple minima as well as a flat optimization landscape, which makes finding global minimizers challenging. We develop a strategy to cope with numerical difficulties, which is effective for the systems considered in this work. We provide details of the strategy below. All numerical results in this paper are obtained from the following protocol. The resulting THC tensors are available in Ref. [76].

1. Initial guess

Due to the nonlinear nature of Eq. (6), good initial guesses are often critical in obtaining accurate THC factorizations. We generate random guesses for χ and ζ when the real-space molecular orbital representation is unavailable. This is the case for the FeMoCo Hamiltonians since only Hamiltonian matrix elements V_{pqrs} are reported in the literature. In this work, we tried 20 random guesses and picked the best performing one in the end.

On the other hand, when we have real-space molecular orbital representation available (which is the case for hydrogen systems and other chemical systems in general) we utilize the interpolative separable density fitting (ISDF) technique to generate an accurate initial guess for χ and ζ. The ISDF approach was originally proposed by Lu and Ying [63]. In terms of the classical precomputation required, this approach is more efficient than the direct gradient-descent approach. It is based on the intuition that the THC factorization is an interpolative decomposition of the electronic pair density in real space:

$$\phi_p(\mathbf{r})\,\phi_q(\mathbf{r}) = \sum_{\mu=1}^{M} \xi_\mu(\mathbf{r})\,\phi_p\left(r_\mu\right)\phi_r\left(r_\mu\right), \tag{7}$$

where $\phi_p(\mathbf{r})$ is the p^{th} single-particle orbital represented on a grid $\{\mathbf{r}\}$, r_μ denotes the interpolation points and $\xi_\mu(\mathbf{r})$ is called an interpolation vector, which can be obtained via a simple least-squares fit. Once r_μ and $\xi_\mu(\mathbf{r})$ are found, the THC factorization is then obtained via

$$\chi_p^{(\mu)} = \phi_p\left(r_\mu\right) \quad \zeta_{\mu\nu} = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{\xi_\mu(\mathbf{r}_1)\xi_\nu(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}. \tag{8}$$

ISDF involves no nonlinear optimization and thereby it is numerically robust and provides straightforward initial guess for subsequent direct minimizations. Finding interpolation points can be also performed with a linear complexity [73] via the centroid Voronoi tessellation (CVT) approach [70]. Determining $\xi_\mu(\mathbf{r})$ scales as $\mathcal{O}(N^3)$ and is ultimately the bottleneck of this algorithm. While this is more economical than also performing further optimization (see the next subsection), the resulting THC factorization is not as compact for given accuracy. Therefore, in this work, we use the ISDF-CVT approach to generate a good initial guess from which we perform further optimization.

2. Optimization

While ISDF initial guesses are quite accurate, random guesses are sometimes very far away from any local minima. This is problematic when an ISDF initial guess is unavailable, which is the case for FeMoCo. When starting from random initial guesses, we find that a quasi-Newton method such as the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints (L-BFGS-B) algorithm [77] is quite robust in the warm-up stage. We note that quasi-Newton methods were previously studied for use in obtaining THC factorizations in Ref. [69]. The necessary gradient of Eq. (6) for BFGS is obtained via the automatic differentiation method in JAX [78]. Unfortunately, L-BFGS-B alone could not produce accurate factorization for all of the systems considered in this work. Therefore, in some cases we also have to perform another optimization with a different solver.

After an L-BFGS-B run, we employ a popular machine learning optimizer called AdaGrad [79] as implemented in JAX to finalize the factorization [78]. We find that L-BFGS-B tends to get easily stuck in unwanted local minima and subsequent optimization via AdaGrad helped to escape from a local minimum and improved the accuracy of our optimization by more than an order of magnitude.

C. Diagonal Coulomb operators from projecting into the auxillary tensor hypercontraction basis

One can attempt to apply qubitization directly to the Hamiltonian representation of Eq. (4). In fact, doing this is relatively straightforward and in Appendix E we describe an algorithm based on exactly that approach. We show that this results in an algorithm with Toffoli complexity $\widetilde{\mathcal{O}}(N\lambda_{\text{THC}}/\epsilon)$, which prima facie, is relatively low complexity. The method of Appendix E is also exceptionally simple as far as arbitrary basis quantum chemistry

qubitizations go; thus, that approach might be valuable for pedagogical purposes. However, the problem with directly applying qubitization to this form of the THC Hamiltonian is that we end up with a $\lambda_{\rm THC}$ defined as

$$\lambda_{\rm THC} = \sum_{p,q,r,s=1}^{N/2} \sum_{\mu,\nu=1}^{M} \left| \chi_p^{(\mu)} \chi_q^{(\mu)} \zeta_{\mu\nu} \chi_r^{(\nu)} \chi_s^{(\nu)} \right|. \quad (9)$$

Because the sum over μ and ν appears outside of the absolute value in this expression, $\lambda_{\rm THC}$ is much larger than even λ_V, the 1-norm of the original Hamiltonian. Note that λ_V is already larger than $\lambda_{\rm DF}$ (the λ value associated with the von Burg *et al.* [10] algorithm), and so this larger λ value should be avoided. To avoid this blow up in λ, the main approach of this paper focuses on simulating a different Hamiltonian that we derive from the THC representation.

As discussed in the Introduction, a number of recent papers have demonstrated very high-efficiency quantum algorithms for simulating the electronic Hamiltonian in a basis that diagonalizes the Coulomb operator [15–18,80]. The downside of these approaches is that the required orthogonal-basis sets often require many more qubits in order to reach chemical accuracy for molecules, compared to arbitrary basis functions such as molecular orbitals, which lead to the Coulomb operator in Eq. (1). On the contrary, the THC representation of the Coulomb operator in Eq. (4) may be more compact, but it is not diagonal. However, as we show below, it is possible to use the χ tensor from the THC factorization to define a larger "auxillary" basis in which the Coulomb operator is diagonal.

We can achieve this diagonal representation by first defining a nonunitary (and nonorthogonal) basis rotation corresponding to a projection of the original fermion ladder operators into a larger basis:

$$c_{\mu,\sigma}^\dagger = \sum_{p=1}^{N/2} \chi_p^{(\mu)} a_{p,\sigma}^\dagger, \quad c_{\mu,\sigma} = \sum_{p=1}^{N/2} \chi_p^{(\mu)} a_{p,\sigma}, \quad (10)$$

where the creation and annihilation operators $c_\mu^\dagger$ and c_μ act on a larger space of $2M$ spin orbitals rather than N spin orbitals. Without loss of generality, we are taking $\chi^{(\mu)}$ to be a normalized vector for each μ (because constant factors can be absorbed into $\zeta_{\mu\nu}$). Using this, we rewrite Eq. (4) as

$$\begin{aligned} G &= \frac{1}{2} \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}} \sum_{\mu,\nu=1}^{M} \left(\sum_{p=1}^{N/2} \chi_p^{(\mu)} a_{p,\alpha}^\dagger \right) \left(\sum_{q=1}^{N/2} \chi_q^{(\mu)} a_{q,\alpha} \right) \\ &\quad \times \left(\sum_{r=1}^{N/2} \chi_r^{(\nu)} a_{r,\beta}^\dagger \right) \left(\sum_{s=1}^{N/2} \chi_s^{(\nu)} a_{s,\beta} \right) \zeta_{\mu\nu} \\ &= \frac{1}{2} \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}} \sum_{\mu,\nu=1}^{M} (c_{\mu,\alpha}^\dagger c_{\mu,\alpha})(c_{\nu,\beta}^\dagger c_{\nu,\beta}) \zeta_{\mu\nu}. \end{aligned} \quad (11)$$

This provides a diagonal form of the Coulomb operator in the expanded basis:

$$G = \frac{1}{2} \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}} \sum_{\mu,\nu=1}^{M} \zeta_{\mu\nu} n_{\mu,\alpha} n_{\nu,\beta}, \quad (12)$$

where $n_{\mu,\sigma} = c_{\mu,\sigma}^\dagger c_{\mu,\sigma}$ is the number operator in the larger basis. We derive this representation of the Hamiltonian (it does not seem to appear in any prior literature even for classical electronic structure). Even though the basis size (and, thus, the number of qubits) is increased by a factor that is roughly between 4 and 10 in most contexts, the appeal of highly efficient algorithms presented in Refs. [15–18,80] for diagonal Coulomb operators makes this representation interesting. Unfortunately, this larger basis is not orthogonal, which complicates our approach to directly simulating this Hamiltonian. For example, one cannot use methods such as those in Refs. [16,18] that were developed to simulate diagonal electronic Hamiltonians in orthogonal-basis sets. Instead, we pursue a qubitization-based approach that involves rotating into the underlying nonorthogonal basis, one tensor factor at a time, avoiding complications that arise if the rotations were done globally. The Hamiltonian in Eq. (12) is referred to as the nonorthogonal THC Hamiltonian.

D. Deriving the λ value associated with the nonorthogonal THC Hamiltonian representation

We now give a very high-level overview of the main approach of this paper, which involves qubitizing Eq. (12), and we derive the λ value associated with that representation. In Ref. [10] it was shown that when qubitizing an operator in a different basis, the basis rotation does not need to rotate all the bases at once. Instead, because it is controlled by a register, it is sufficient to perform basis rotations independently for each number operator controlled by the register. As shown in Eq. (51) of Ref. [10], only $N/2$ Givens rotations are needed, instead of $\mathcal{O}(N^2)$ if all N basis vectors are being rotated at once. So in our case, we can control on μ to rotate the basis to that described by $n_{\nu,\alpha}$ using N Givens rotations in the same way as shown in Eq. (51) of Ref. [10]. Since $\chi^{(\mu)}$ is taken to be a normalized vector for each μ, this is a valid rotation for each individual μ. It does not matter that the basis is not orthogonal, because the rotations are done individually for each μ separately, rather than jointly for all μ together.

It is also possible to apply other methods from Ref. [10] associated with implementing these rotations.

1. Use a QROM to output the $N/2$ rotation angles controlled on the register μ, which takes M values.
2. Since the number operators can be represented by $(\mathbb{1} - Z)/2$, we can take the identity parts out and combine them with the one-body terms, and the

remaining two-body terms have a λ value that is divided by 4.

Note also that the procedure to rotate to and from the new basis must be done twice, once for the operators dependent on ν and again for those dependent on μ. The net result is that the complexity for a single step is $\widetilde{\mathcal{O}}(N)$, and the value of λ has a contribution from two-body terms

$$\lambda_\zeta = \frac{1}{2}\sum_{\mu,\nu}|\zeta_{\mu\nu}|. \tag{13}$$

To determine the λ value more carefully, G can be given as

$$G = \frac{1}{2}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger n_{1,\alpha} U_\mu U_\nu^\dagger n_{1,\beta} U_\nu \zeta_{\mu\nu}. \tag{14}$$

In actually implementing this, we would use α or β to control swaps between the qubits representing spin-up and spin-down orbitals, so U_μ would need only to act on $N/2$ qubits, but for simplicity this is not shown explicitly here. Then by taking $n_{1,\alpha} = (\mathbb{1} - Z_{1,\alpha})/2$, we have

$$\begin{aligned}
G &= \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger(\mathbb{1} - Z_{1,\alpha})U_\mu U_\nu^\dagger(\mathbb{1} - Z_{1,\beta})U_\nu\zeta_{\mu\nu} \\
&= -\frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger U_\mu U_\nu^\dagger U_\nu\zeta_{\mu\nu} \\
&\quad + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger(\mathbb{1} - Z_{1,\alpha})U_\mu U_\nu^\dagger U_\nu\zeta_{\mu\nu} \\
&\quad + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger U_\mu U_\nu^\dagger(\mathbb{1} - Z_{1,\beta})U_\nu\zeta_{\mu\nu} \\
&\quad + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger Z_{1,\alpha} U_\mu U_\nu^\dagger Z_{1,\beta} U_\nu\zeta_{\mu\nu} \\
&= -\frac{1}{2}\mathbb{1}\sum_{\mu,\nu=1}^{M}\zeta_{\mu\nu} + \frac{1}{2}\sum_{\alpha\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger(\mathbb{1} - Z_{1,\alpha})U_\mu\zeta_{\mu\nu} \\
&\quad + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger Z_{1,\alpha} U_\mu U_\nu^\dagger Z_{1,\beta} U_\nu\zeta_{\mu\nu}.
\end{aligned} \tag{15}$$

The first term corresponds to an overall energy shift that can be ignored. The third term is the two-body term that gives the contribution to λ given in Eq. (13). The middle term can be given as

$$\begin{aligned}
T^{(2\to1)} &= \sum_{\alpha\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M} U_\mu^\dagger n_{1,\alpha} U_\mu\zeta_{\mu\nu} \\
&= \sum_{\alpha\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M}\left(\sum_{p=1}^{N/2}\chi_p^{(\mu)} a_{p,\alpha}^\dagger\right)\left(\sum_{q=1}^{N/2}\chi_q^{(\mu)} a_{q,\alpha}\right)\zeta_{\mu\nu} \\
&= \sum_{\alpha\in\{\uparrow,\downarrow\}}\left(\sum_{\mu,\nu=1}^{M}\chi_p^{(\mu)}\chi_q^{(\mu)}\zeta_{\mu\nu}\right) a_{p,\alpha}^\dagger a_{q,\alpha}.
\end{aligned} \tag{16}$$

Now note that

$$\begin{aligned}
\sum_{r=1}^{N/2} G_{pqrr} &= \sum_{\mu,\nu=1}^{M}\chi_p^{(\mu)}\chi_q^{(\mu)}\zeta_{\mu\nu}\sum_{r=1}^{N/2}\chi_r^{(\nu)}\chi_r^{(\nu)} \\
&= \sum_{\mu,\nu=1}^{M}\chi_p^{(\mu)}\chi_q^{(\mu)}\zeta_{\mu\nu}.
\end{aligned} \tag{17}$$

For this term we can replace the approximation G_{pqrr} with the exact V_{pqrr}, so the approximation is used only in the two-body term. Therefore, combining $T^{(2\to1)}$ with T gives a new one-body operator

$$\begin{aligned}
T' &= T + T^{(2\to1)} = \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T'_{pq} a_{p,\sigma}^\dagger a_{q,\sigma} \quad \text{with} \\
T'_{pq} &= T_{pq} + \sum_{r=1}^{N/2} V_{pqrr}.
\end{aligned} \tag{18}$$

Then T' can be diagonalized as

$$\begin{aligned}
T' &= \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{\ell=1}^{N/2} t_\ell n_{\ell,\sigma} = \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{\ell=1}^{N/2} t_\ell U_{T,\ell}^\dagger n_{1,\sigma} U_{T,\ell} \\
&= \sum_{\ell=1}^{N/2} t_\ell\mathbb{1} - \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{\ell=1}^{N/2} t_\ell U_{T,\ell}^\dagger Z_{1,\sigma} U_{T,\ell},
\end{aligned} \tag{19}$$

where t_ℓ are eigenvalues of T'_{pq}, and $U_{T,p}$ are individual rotations for T similar to the U_p for G. The first term is an overall energy shift, and the second term is the one-body term that contributes towards λ. The net result of this is

$$\lambda = \sum_{\ell=1}^{N/2}|t_\ell| + \frac{1}{2}\sum_{\mu,\nu}|\zeta_{\mu\nu}| = \mathcal{O}\left(\lambda_\zeta\right). \tag{20}$$

As we discuss later on, λ_ζ actually scales even better than the λ values associated with any prior algorithm in the literature, including the λ_{DF} associated with the doubled factorized algorithm of von Burg *et al.* [10].

III. QUBITIZING THE NONORTHOGONAL TENSOR HYPERCONTRACTION HAMILTONIAN

A. Approach to qubitization

Our approach to encoding the eigenspectra of the THC representation of the electronic Hamiltonian in a unitary for phase estimation uses the linear combination of unitaries (LCU) query model [81]. Specifically, we use qubitization [12] to block encode [82] the Hamiltonian eigenspectra as a Szegedy quantum walk [83]. What all LCU methods have in common is that they involve simulating or block encoding the Hamiltonian from a representation where it can be accessed as a linear combination of unitaries:

$$H = \sum_{\ell=1}^{L} \omega_\ell \, U_\ell, \tag{21}$$

where U_ℓ are unitary operators and the ω_ℓ are scalars. LCU methods are defined in terms of queries to two oracle circuits that are commonly defined as

$$\begin{aligned} \text{SELECT} \, |\ell\rangle \, |\psi\rangle &\mapsto |\ell\rangle \, U_\ell \, |\psi\rangle \\ \text{PREPARE} \, |0\rangle^{\otimes \log L} &\mapsto \sum_{\ell=1}^{L} \sqrt{\frac{\omega_\ell}{\lambda}} \, |\ell\rangle \equiv |\mathcal{L}\rangle \\ \lambda &= \sum_{\ell=1}^{L} |\omega_\ell| \, , \end{aligned} \tag{22}$$

where $|\psi\rangle$ is the system register, $|\ell\rangle$ is an ancilla register, which usually indexes the terms in Eq. (21) in binary, and this is the general definition of λ.

Currently, the most practical fault-tolerant approaches for simulating quantum chemistry are based on qubitization [9,10,12,16]. Following the analysis and techniques of Ref. [16], one can use phase estimation based on qubitized quantum walks to sample in the eigenbasis of a Hamiltonian with error in the sampled eigenvalue bounded from above by ϵ with Toffoli complexity scaling exactly as

$$\left\lceil \frac{\pi \lambda}{2\epsilon_{\text{PEA}}} \right\rceil \left[C_S + C_P + C_{P^\dagger} + \log L + \mathcal{O}(1) \right], \tag{23}$$

where C_S is the gate complexity of SELECT, C_P is the gate complexity of PREPARE, $C_{P^\dagger} \leq C_P$ is the gate complexity of uncomputing PREPARE, and ϵ_{PEA} is the allowable error in the phase estimation.

The multiplying factor of $\lceil \pi \lambda / (2\epsilon_{\text{PEA}}) \rceil$ corresponds to the number of repetitions of the LCU step used in the phase estimation. In Ref. [16] the number of repetitions was taken to be a power of 2, because that makes the phase estimation particularly simple, with each qubit of the control registers controlling a number of repetitions that is a power of 2. It is also possible to use a number of repetitions that is an arbitrary integer. This was assumed by von Burg *et al.* [10], although doing this requires a more sophisticated control which those authors do not show how to perform. We explain how this can be accomplished. The general principle is to control each step using the unary iteration procedure introduced in Ref. [16].

To explain the procedure in more detail, one should combine the SELECT operation together with a reflection $R = 2 \, |\mathcal{L}\rangle\langle\otimes| \, \mathbb{1} - \mathbb{1}$ based on the PREPARE operation, to create a step of a quantum walk $\mathcal{W}$, which has eigenvalues proportional to $e^{\pm i \arccos(E_n/\lambda)}$ where $H \, |n\rangle = E_n \, |n\rangle$ [13,14]. In order for this procedure to work, the SELECT needs to be self-inverse. It is possible to obtain the same complexity if there is controlled application of SELECT or SELECT†, but we avoid that because it doubles the complexity, and instead construct SELECT so it is self-inverse.

Simplistically, one could make $\mathcal{W}$ controlled as shown in Fig. 1, but for the purpose of obtaining the complexity shown in Eq. (23), one needs to control application of $\mathcal{W}$ and its inverse. Given that SELECT is self-inverse, one can obtain $\mathcal{W}^\dagger$ simply by performing the reflection before SELECT instead of after. That means that one could control between $\mathcal{W}$ and $\mathcal{W}^\dagger$ by performing a controlled reflection before SELECT as well as after (and not making SELECT controlled). But, it is not necessary to perform two controlled reflections, only one. To see why, consider the case where the control is selecting four applications of $\mathcal{W}$ with the rest of the operations being $\mathcal{W}^\dagger$. Then we need to perform the sequence of operations

$$\begin{aligned} &(RU)(RU)(RU)(RU)(UR)(UR)(UR)\ldots \\ &\quad = (RU)(RU)(RU)(RU)\mathbb{1}(UR)(UR)(UR)\ldots, \end{aligned} \tag{24}$$

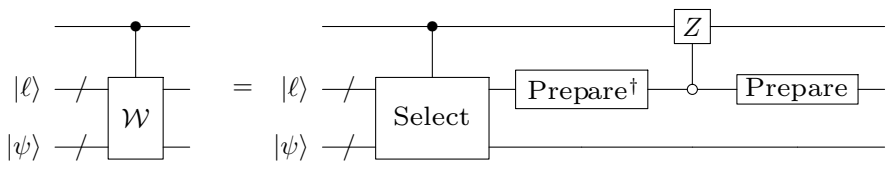

FIG. 1. A circuit realizing the qubitized quantum-walk operator $\mathcal{W}$ controlled on an ancilla qubit. Note that the Z gate with the 0 control is actually controlled on the zero state of the entire $|\ell\rangle$ register and not just a single qubit. Accordingly, implementation of that controlled Z has gate complexity $\log L + \mathcal{O}(1)$ where $\lceil \log L \rceil$ is the size of the $|\ell\rangle$ register.

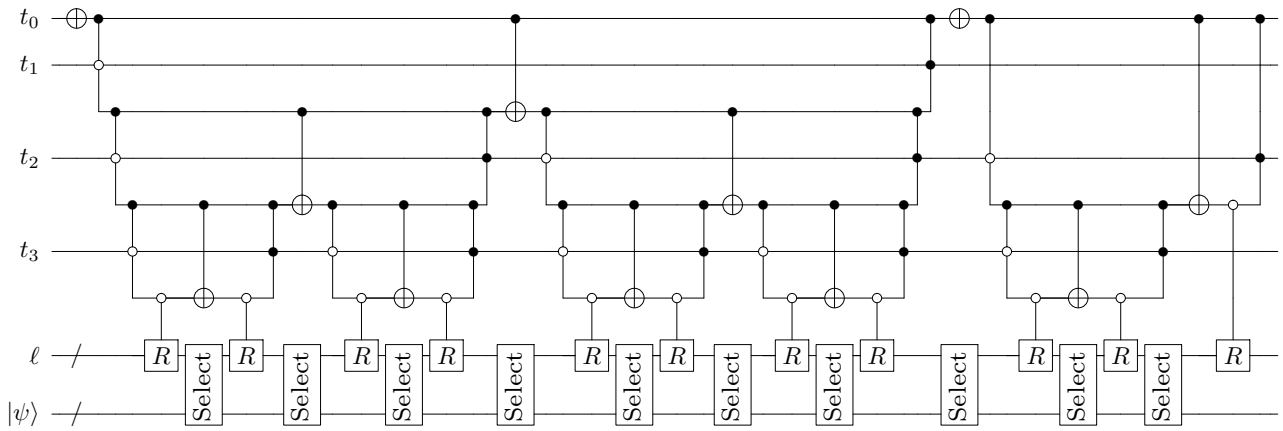

FIG. 2. A unary iteration circuit for selecting $\mathcal{W}^{2t-10}$ for t from 0 to 10. Here R indicates the inverse preparation, reflection, and preparation. As in Fig. 1 only the reflection needs to be made controlled.

where U is being used for SELECT and R is being used for the reflection (i.e., the combined inverse preparation, reflection about the zero state, and preparation). We use only U or R with this meaning in this equation and in Fig. 2. Here, after the fourth U, we simply perform the identity rather than a reflection. This means that we *always* perform the reflection in between each successive SELECT, except when the number of times we perform SELECT matches the number in the control register. That means we need only one controlled reflection in between each SELECT, which corresponds to removing the control on SELECT in Fig. 1.

Next we explain how to address cases where $\lceil \pi\lambda/(2\epsilon_{\mathrm{PEA}})\rceil$ is not a power of 2. To achieve that, one can simply prepare the optimal control state as described in Ref. [16] except using a superposition over a number of basis states that is smaller than a power of 2. Then, instead of controlling on each successive qubit of this state, use the unary iteration procedure of Ref. [16] to control the reflection. The unary iteration procedure introduces a trivial additional cost of one Toffoli per step, and doubles the ancilla cost of the control register for the temporary ancillas for the unary iteration. To show explicitly how this control is done, see the example in Fig. 2.

In our algorithm there are four sources of error:

1. Error due to measurement in phase estimation ϵ_{PEA}, which first appears in Eq. (23).
2. The approximation of the Hamiltonian coefficients as part of the coherent alias sampling procedure from Ref. [16] that is used as part of our PREPARE strategy.
3. The approximation of the individual Givens rotations needed for the basis rotations.
4. The approximation of tensor hypercontraction ϵ_{THC}, which first appears in Eq. (5).

To bound the overall error, one could take these individual errors and simply add them together. However, the sources of error 2 to 4 all contribute to error in the approximation of the Hamiltonian. That is, they together give an approximate Hamiltonian, and one can simply estimate the error due to using this approximate Hamiltonian.

One approach would be to determine the root-mean-square sum of the errors of each of the coefficients in the Hamiltonian as is done in Ref. [10]. Based on classical simulations we find that this overestimates the error, so we instead perform classical calculations of the error in the energy to estimate the allowable truncations. We include all error from approximation of the Hamiltonian into ϵ_{THC}, and require that

$$\epsilon \geq \epsilon_{\mathrm{PEA}} + \epsilon_{\mathrm{THC}}, \tag{25}$$

where ϵ is our target accuracy, which we take to be 0.0016 hartree (chemical accuracy) for the resource estimates of this paper. We take $\epsilon_{\mathrm{PEA}} \leq 0.001$ hartree and $\epsilon_{\mathrm{THC}} \leq 0.0006$ hartree. The allowable error for phase estimation in Ref. [10] was 0.0009 hartree, but we recalculate the cost with $\epsilon_{\mathrm{PEA}} \leq 0.001$ hartree for the DF method of Ref. [10] for a fair comparison. We also recalculate the costs using the same parameters for the sparse and SF approaches.

Note that in quantum chemistry it is typical to require the differences in energy between two configurations to be determined to chemical accuracy, which naïvely would require accuracy of 0.0008 hartree on each estimation of each energy. That precision is expected to not be necessary, because the approximations made in the Hamiltonian for the two configurations have correlated errors. For the phase estimation, the errors would add if the computation is performed independently for the two configurations, but it is possible to use the control register to control a forward evolution for one Hamiltonian on one target system, *and* reverse evolution on a second target system with

the Hamiltonian for the second configuration. This can be combined with the improved phase-estimation techniques of Ref. [14], which help to ensure one is projecting into the correct states at lower cost than the entire phase estimation. The phase estimation will then provide an estimate of the energy difference. We therefore continue to assume that the accuracy required is overall 0.0016 hartree, which also provides consistency with prior work.

We now discuss how to implement and compile PREPARE and SELECT for our algorithm.

B. State preparation for the nonorthogonal tensor hypercontraction Hamiltonian

The method to perform the PREPARE step is based on expressing the Hamiltonian in a nonorthogonal basis as in Eq. (12). We can rewrite the Hamiltonian in the form

$$H = -\frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{\ell=1}^{N/2} t_\ell U_{T,\ell}^\dagger Z_{1,\sigma} U_{T,\ell} + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{\mu,\nu=1}^{M}\zeta_{\mu\nu} U_\mu^\dagger Z_{1,\alpha} U_\mu U_\nu^\dagger Z_{1,\beta} U_\nu, \quad (26)$$

where the first term is from T' in Eq. (19) and the second term is from G in Eq. (15). The state we need is

$$\frac{1}{\sqrt{\lambda}}|+\rangle|+\rangle\left[\sum_{\ell=1}^{N/2}\sqrt{|t_\ell|}\,|\ell\rangle|M+1\rangle + \frac{1}{\sqrt{2}}\sum_{\mu,\nu=1}^{M}\sqrt{|\zeta_{\mu\nu}|}\,|\mu\rangle|\nu\rangle\right]. \quad (27)$$

Here the first and second registers give the superpositions over spins, and the third and fourth registers store μ and ν. The last register takes the value $M+1$ to flag that this is the first term in the Hamiltonian. For simplicity we adopt the convention that these registers are labeled starting from 1, though 1 would be stored as binary 0 in the register.

For the state preparation, we have a three-step procedure where we first prepare an equal superposition over the μ and ν registers, then perform coherent alias sampling [16], then swap the μ and ν registers controlled by an ancilla in a $|+\rangle$ state. This means we need only to initially prepare the range $\mu \le \nu$, and the number of coefficients needed in the state preparation is

$$d = N/2 + M(M+1)/2. \quad (28)$$

In the first step we need to create an equal superposition over $\mu \le \nu \le M+1$ for registers 3 and 4, with $\mu \le N/2$ for $\nu = M+1$. The procedure needed is depicted in Fig. 3.

The method is to perform Hadamards on all qubits for these registers, then perform the inequality tests $\nu \le M+1$, $\mu \le \nu$, and $\mu \le N/2$ controlled on $\nu = M+1$. These inequality tests have cost $4(n_M - 1)$, with $n_M = \lceil \log(M+1) \rceil$. Rotate an ancilla qubit by adding a constant into a phase gradient register (as described in Appendix A of Ref. [84]), to obtain an overall amplitude for success on this qubit and the three inequality tests approximately 1/2. This needs $b_r - 3$ Toffolis, using b_r bits of precision, which can typically be taken to be about 7. We can flag failure on an ancilla and perform the identity (instead of SELECT) in the case of failure. Then we need to reflect on five registers, which we do using three Toffolis. Next, we can invert the inequality tests (which may be done without further Toffoli costs using the out-of-place adders of Ref. [85]), and invert the rotation with cost $b_r - 3$. Then inverting the Hadamards and reflecting about zero has cost $2n_M - 1$. Then we perform the Hadamards and inequality tests again with cost $4n_M - 3$. Checking that

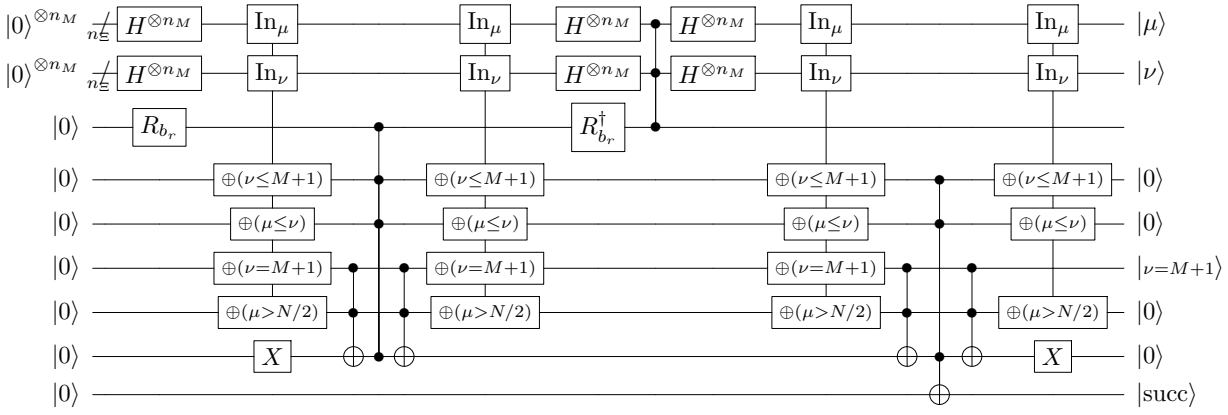

FIG. 3. The circuit for creating an equal superposition over the μ and ν registers, with $n_M = n_M = \lceil \log(M+1) \rceil$, and R_{b_r} being a Y rotation to b_r bits of precision. The state is prepared on the first two registers, the last register flags success of the entire procedure, and the fourth-last register records whether ν is equal to $M+1$. We use $|\mu\rangle$ and $|\nu\rangle$ to label the outputs on the top two registers, though these are in a superposition state.

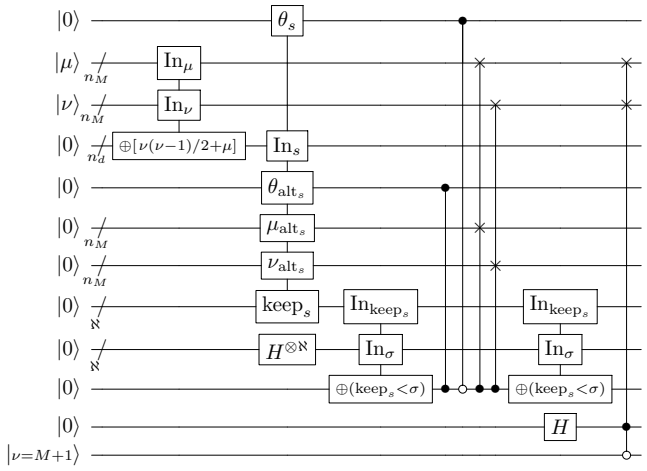

FIG. 4. The state preparation after preparation of the equal superposition state in Fig. 3. Here $n_d = \lceil \log d \rceil$, and we use $|\mu\rangle$ and $|\nu\rangle$ to label the input registers that are in a superposition. The modifications to this state preparation over that in Ref. [16] are that a contiguous register is calculated in the beginning, and at the end the μ and ν registers are swapped controlled on a $|+\rangle$ state and ν not being equal to $M+1$, which is contained in a register output from the procedure for creating the equal superposition. Since we simply need to perform a Z gate on the sign qubit θ, instead of explicitly performing a controlled swap with the alt value, we can perform two controlled phase operations thereby eliminating one non-Clifford gate. These controlled phase gates need not be performed when inverting the state preparation.

the inequality tests are satisfied has cost 3, giving a total cost of $10\lceil \log(M+1) \rceil + 2b_r - 9$ Toffolis.

Once we prepare the equal superposition over the μ and ν registers, we can then prepare the state as shown in Fig. 4. In more detail, the steps needed are as follows.

1. Use Hadamards to create the $|+\rangle$ states on the first two registers.
2. Create a new contiguous register from registers 3 and 4. This contiguous register can be given as

$$s = \nu(\nu-1)/2 + \mu, \tag{29}$$

where we are using the convention in this equation that numbering starts from 1. Because we are using the convention that $\nu = M+1$ flags the first term where $\mu \leq N/2$, the allowed range of values for μ and ν gives a contiguous range of values for s. The complexity of computing s is $n_M^2 + n_M - 1$, which is shown in Appendix F.
3. Use the new contiguous register to output alternate values for registers 4 and 5, a sign qubit and an alternate sign qubit (to account for the signs in the linear combination of unitaries), and keep values. The size of the output is then

$$m = 2n_M + 2 + \aleph, \tag{30}$$

where $\aleph$ is the number of bits for the keep register. The cost of the QROM is then

$$\left\lceil \frac{d}{k_{s1}} \right\rceil + m(k_{s1} - 1). \tag{31}$$

4. Perform an inequality test between the keep register and a register in an equal superposition, with cost $\aleph$.
5. Perform a swap controlled on the result of the inequality test, with cost $2n_M$. We can simply perform controlled phase gates on the sign qubits instead of explicitly swapping them, which is why we simply have the cost of swapping the μ and ν registers with the alternate registers.
6. Controlled on a $|+\rangle$ state, *and* the result of the test for $\nu = M+1$ from step 2, swap the μ and ν registers, with cost $n_M + 1$. Here the $+1$ is the Toffoli needed to perform the AND operation on two qubits for the control.

In inverting the state preparation, the cost of the inverse QROM is reduced to

$$\left\lceil \frac{d}{k_{s2}} \right\rceil + k_{s2}, \tag{32}$$

but the other costs are unchanged. Adding all these costs together gives

$$C_P + C_{P^\dagger} = 28n_M + 4b_r - 18 + 2n_M^2 + 2\aleph + \left\lceil \frac{d}{k_{s1}} \right\rceil + m(k_{s1} - 1) + \left\lceil \frac{d}{k_{s2}} \right\rceil + k_{s2}, \tag{33}$$

with $m = 2n_M + 2 + \aleph$, $n_M = \lceil \log(M+1) \rceil$.

The number of bits used for the keep register was $\aleph = \lceil 2.5 + \log(10\lambda/\epsilon) \rceil$ in Ref. [10]. The error in the keep probability is translated to that in the state in the following way. The squared amplitudes of the state needed are $|t_\ell|/\lambda$ and $|\zeta_{\mu\nu}|/\lambda$ for $\mu \neq \nu$, but $|\zeta_{\mu\mu}|/2\lambda$ for $\mu = \nu$. These amplitudes are compared to initial squared amplitudes in the equal superposition of $1/d$. Taking $\aleph$ as the number of bits for the keep probability is equivalent to discretizing the squared amplitudes to the nearest $1/(2^\aleph d)$ [see Eq. (35) of Ref. [16]]. It would at first be expected that this would lead to an error no larger than $\lambda/(2^{\aleph+1}d)$ in $|t_\ell|$ or $|\zeta_{\mu\nu}|$ for $\mu \neq \nu$, or $\lambda/(2^\aleph d)$ for $|\zeta_{\mu\mu}|$, with a factor of 2 reduction in the error because rounding gives error no larger than half the discretization size. The subtlety is that if we round all the squared amplitudes, the result will no longer be normalized. To maintain normalization it will be necessary to

discretize some of the amplitudes in a different way than rounding, resulting in an error larger than $\lambda/(2^{\aleph+1}d)$, but still no larger than $\lambda/(2^{\aleph}d)$. That is why the factor of 2 is not included in Ref. [16]. However, it is possible to make the typical errors smaller than $\lambda/(2^{\aleph+1}d)$ by adjusting the threshold for rounding from $1/2$ to maintain normalization. In practice only a very small adjustment is needed.

C. Hamiltonian selection oracle for the nonorthogonal tensor hypercontraction Hamiltonian

For the SELECT operation, we have two steps. In the first we need to perform the operation U_μ or $U_{T,\mu}$ for the two-electron or one-electron terms, respectively. In the second we need only to perform U_ν. To perform these operations, we can use a QROM to generate a sequence of $N/2$ rotations, then perform the rotations using the method given in Ref. [10]. The costs are as follows.

1. There is a cost of $N/2$ to perform the swaps controlled on the spin qubit.
2. The dominant cost is the QROM. The output size is large, so it is most efficient to perform the QROM as in Ref. [16]. In the first step (for μ) we have $M + N/2$ sets of data to output which has complexity $M + N/2 - 2$. Note that we need to output one set of rotations if the ν register is in the state $|M+1\rangle$, or a different set if the ν register is not in that state. The value $\nu = M + 1$ is still flagged in an ancilla. In the second step (for ν) we need only to generate the rotations for the two-electron terms, so the cost is $M - 2$.
3. We can perform the rotations with cost $N(\beth - 2)$ by adding the rotations into the phase gradient state. Here $\beth$ is the number of bits of precision used for the rotation angles.
4. The Z operation has no Toffoli cost. It needs to be controlled on the qubit, which is the result of testing whether $\nu = M + 1$ for the case where we apply the operations for ν.
5. We invert the rotations with cost $N(\beth - 2)$.
6. To erase the QROM, in the case of μ we need to subdivide the bits into the more- and less-significant bits, where the more-significant bit includes distinguishing the case $\nu = M + 1$. We have a cost of $\lceil M/k_{r1} \rceil$ for the case $\nu < M + 1$, and a cost $\lceil N/2k_{r1} \rceil$ for the case $\nu = M + 1$, then a cost k_{r1} for the remaining k_{r1} less-significant bits, for a total of
$$\left\lceil \frac{M}{k_{r1}} \right\rceil + \left\lceil \frac{N}{2k_{r1}} \right\rceil + k_{r1}. \tag{34}$$
For the case of μ, we have to perform only the QROM on M possible values, giving a cost of
$$\left\lceil \frac{M}{k_{r2}} \right\rceil + k_{r2}. \tag{35}$$
7. Perform the swaps controlled by the ancillas (for spin) again with cost $N/2$.

This procedure needs to be performed twice, once for μ and once for ν. The complete procedure is shown in Fig. 5.

A subtlety is that the operation needs to be controlled by success of the state preparation, which can be achieved

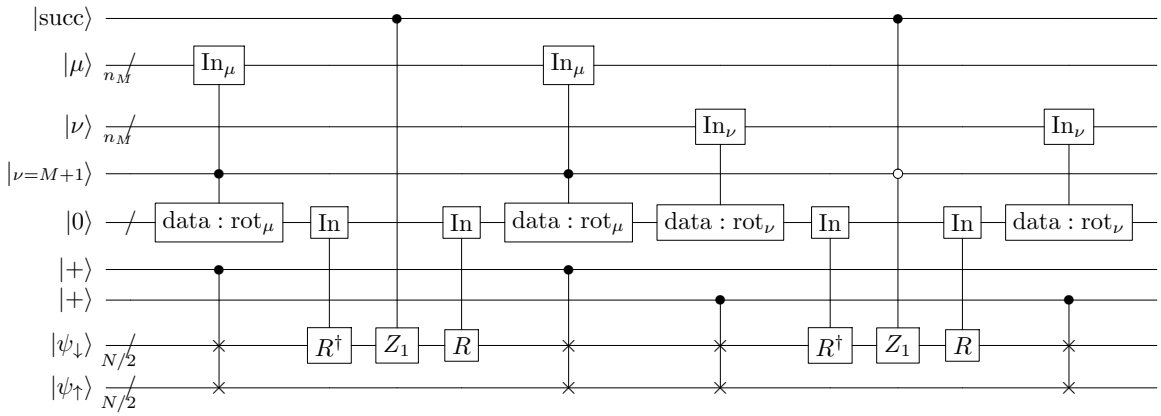

FIG. 5. The circuit for performing the controlled operations (select) for the linear combination of unitaries. The registers from top to bottom are the success flag register, the μ and ν control registers, the qubit flagging that $\nu = M + 1$, a blank ancilla to put the database of rotations in, two $|+\rangle$ states to provide the superposition over the operations on the spin-up and -down components of the state, and the qubits representing the spin-down and -up orbitals. The controlled R is indicating the basis change obtained from the sequence of rotations with angles provided in the database, equivalent to $U_{\vec{u},0}U_{\vec{u},1}$ in Ref. [10], which is depicted in Eqs. (58) and (59) of the Supplemental Material of that work. The registers labeled $|\psi_\downarrow\rangle$ and $|\psi_\uparrow\rangle$ correspond to the qubits for the spin-down and spin-up orbitals. The operation Z_1 acts on only one of the qubits of the $|\psi_\downarrow\rangle$ register, similar to the circuit diagram labeled (57) in the Supplemental Material of Ref. [10].

simply by making the Z operation controlled, which is a Clifford gate, so does not add to the Toffoli cost. Another subtlety is that for ν, we perform no operation if $\nu = M + 1$. That can be achieved simply by making the Z operation controlled by the qubit output by the result of the equality test as well, requiring just one more Toffoli gate.

A major subtlety is that the SELECT operation needs to be made self-inverse, and the operation as depicted is not self-inverse. This problem can be averted by using a NOT operation on the register controlling the swap of the μ and ν registers. To see why this is so, consider the prepared state (ignoring the one-electron term for simplicity)

$$\sum_{\mu\leq\nu} \zeta'_{\mu\nu} |\mu\rangle |\nu\rangle , \tag{36}$$

where $\zeta'_{\mu\nu}$ are the amplitudes needed for the asymmetric state. Introduce the ancilla in the $|+\rangle$ state and perform a controlled swap, giving

$$\frac{1}{\sqrt{2}} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} \left(|0\rangle |\mu\rangle |\nu\rangle + |1\rangle |\nu\rangle |\mu\rangle\right) . \tag{37}$$

Then, denoting $V_\mu = U_\mu^\dagger Z_{1,\alpha} U_\mu$ and similarly for ν, and performing the controlled operation on the target system in state $|\psi\rangle$, we have

$$\frac{1}{\sqrt{2}} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} \left(|0\rangle |\mu\rangle |\nu\rangle V_\mu |\psi\rangle + |1\rangle |\nu\rangle |\mu\rangle V_\nu |\psi\rangle\right) . \tag{38}$$

Now performing an X on the ancilla qubit and swapping the μ and ν registers, we have

$$\frac{1}{\sqrt{2}} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} \left(|0\rangle |\mu\rangle |\nu\rangle V_\nu |\psi\rangle + |1\rangle |\nu\rangle |\mu\rangle V_\mu |\psi\rangle\right) . \tag{39}$$

Then, again performing the controlled operation on the target system, we have

$$\frac{1}{\sqrt{2}} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} \left(|0\rangle |\mu\rangle |\nu\rangle V_\mu V_\nu |\psi\rangle + |1\rangle |\nu\rangle |\mu\rangle V_\nu V_\mu |\psi\rangle\right) . \tag{40}$$

Performing the controlled swap again gives

$$\frac{1}{\sqrt{2}} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} \left(|0\rangle |\mu\rangle |\nu\rangle V_\mu V_\nu |\psi\rangle + |1\rangle |\mu\rangle |\nu\rangle V_\nu V_\mu |\psi\rangle\right) . \tag{41}$$

Finally, projecting onto $|+\rangle$ for the ancilla qubit gives

$$\frac{1}{2} \sum_{\mu\leq\nu} \zeta'_{\mu\nu} |\mu\rangle |\nu\rangle \left(V_\mu V_\nu + V_\nu V_\mu\right) |\psi\rangle . \tag{42}$$

Thus, this procedure gives the desired sum of operations $V_\mu V_\nu + V_\nu V_\mu$ and is self-inverse. As shown in Fig. 6, the form of the circuit can be simplified so that there are just controlled swaps performed on the μ and ν registers at the beginning and the end. Those controlled swaps correspond to the controlled swap at the end of Fig. 4, with the controlled swap at the end corresponding to that done when inverting the state preparation.

To be more specific, for the complete SELECT operation we can include the controlled swaps, and perform the operation as shown in Fig. 7. For that form of the SELECT operation the controlled swap at the end of state preparation in Fig. 4 would not be performed, because it is being bundled into the SELECT operation. As can be seen from Fig. 7, that form of the operation is more clearly self-inverse, though there is the subtlety that we have the qubit with $\nu = M + 1$ flagging the one-electron component of the Hamiltonian. In the case $\nu \neq M + 1$, the circuit simplifies to a completely symmetric form. In the case $\nu = M + 1$, only the left half of the circuit (shown in the dotted box) is performed, which is itself clearly self-inverse due to symmetry. The swap of the μ and ν registers is shown as controlled by the register flagging $\nu = M + 1$ here, which would require more

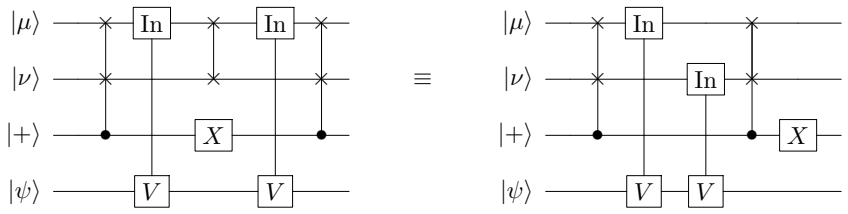

FIG. 6. This shows that the procedure for generating the two-electron terms is self-inverse. The self-inverse controlled operations V are equivalent to $V_\mu = U_\mu^\dagger Z_{1,\alpha} U_\mu$, or V_ν, but we omit the dependence on μ and ν because it is in a superposition of being dependent on both. On the left is the form of the circuit as we describe it in Eq. (36) to Eq. (42), which is obviously self-inverse because of the symmetry of the circuit. On the right is a simplified form of the circuit.

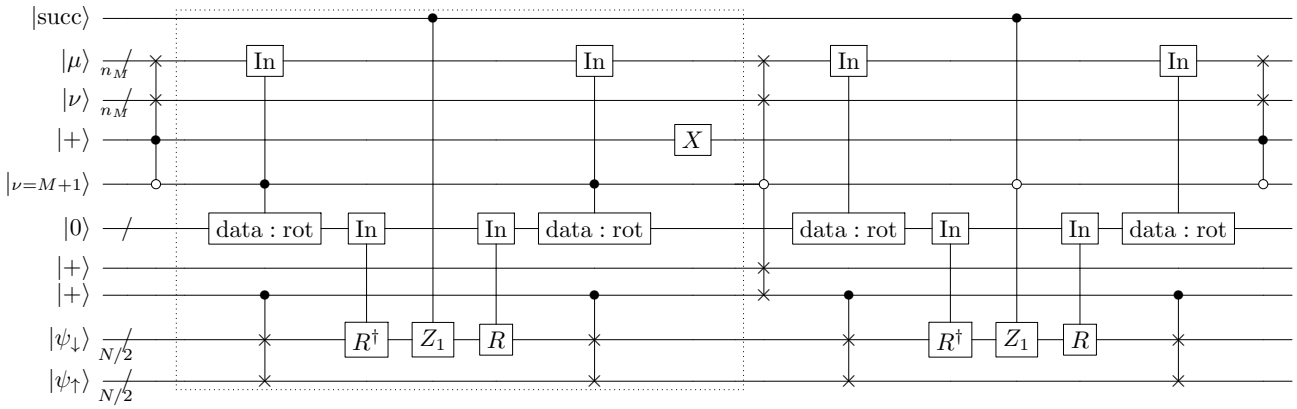

FIG. 7. The circuit for performing the controlled operations (SELECT) for the linear combination of unitaries in a form that is more clearly self-inverse. The registers from top to bottom are the success flag register, the μ and ν control registers, the qubit in the $|+\rangle$ state to control swapping the μ and ν registers, the qubit flagging that $\nu = M + 1$, a blank ancilla to put the database of rotations in, two $|+\rangle$ states to provide the superposition over the operations on the spin-up and -down components of the state, and the qubits representing the spin-down and -up orbitals. The dotted box shows the only part that is applied when $\nu = M + 1$. The controlled R is the rotation of the Majorana basis, equivalent to $U_{\vec{u},0}U_{\vec{u},1}$ in Ref. [10]. The registers labeled $|\psi_\downarrow\rangle$ and $|\psi_\uparrow\rangle$ correspond to the qubits for the spin-down and spin-up orbitals. The operation Z_1 acts on only one of the qubits of the $|\psi_\downarrow\rangle$ register, similar to the circuit diagram labeled (57) in the Supplemental Material of Ref. [10].

Toffoli gates. However, this form is just used to illustrate that the circuit is self-inverse, and that controlled swap can be moved to the end and combined with the other controlled swap, resulting in no more Toffolis being needed. There is one more Toffoli needed in order to apply the controlled swap between the two ancilla qubits in the $|+\rangle$ state which give the spin. As a result, the total complexity for the SELECT operation is only two Toffolis more (the second Z operation must be made controlled) than described in the list above, and can be given as

$$C_S = 2M + 4N\beth - \frac{11N}{2} + \left\lceil \frac{M}{k_{r1}} \right\rceil + \left\lceil \frac{N}{2k_{r1}} \right\rceil + k_{r1} + \left\lceil \frac{M}{k_{r2}} \right\rceil + k_{r2} - 2. \tag{43}$$

D. Overall costs of the nonorthogonal tensor hypercontraction Hamiltonian simulation

In order to construct the entire step of the quantum walk, we also need a reflection on the ancillas used for the control state. We need $2n_M + \aleph + 4$ qubits, which are as follows.

1. We need $2n_M$ for the μ and ν registers.
2. The equal superposition state for the coherent alias sampling needs $\aleph$ qubits.
3. One qubit that controls the swap of the μ and ν registers.
4. One ancilla that is rotated to ensure the amplitude amplification for the equal superposition state preparation works.
5. Two qubits encoding the superposition over spins.

The complexity needed to perform a reflection on this many qubits is $2n_M + \aleph + 2$. Another cost needed for each step is to perform a step of the unary iteration on the control register. This has a cost of another single Toffoli for each step. The output of that unary iteration needs to be used to control the reflection about the ancilla as well, which adds another single Toffoli. Another single Toffoli cost is the controlled swap of the spin registers in Fig. 7. Altogether that gives an additional cost of $2n_M + 5$ for each step, in addition to the cost of the PREPARE and SELECT operations. The total complexity for a single step can then be given as

$$\begin{aligned} C_S + C_P + C_{P^\dagger} + 2n_M + \aleph + 4 &= 30n_M + 4b_r - 16 + 2n_M^2 + 3\aleph + \left\lceil \frac{d}{k_{s1}} \right\rceil \\ &+ m(k_{s1} - 1) + \left\lceil \frac{d}{k_{s2}} \right\rceil + k_{s2} \\ &+ 2M + 4N\beth - \frac{11N}{2} + \left\lceil \frac{M}{k_{r1}} \right\rceil + \left\lceil \frac{N}{2k_{r1}} \right\rceil \\ &+ k_{r1} + \left\lceil \frac{M}{k_{r2}} \right\rceil + k_{r2}\,, \end{aligned} \tag{44}$$

with $m = 2n_M + 2 + \aleph$, $n_M = \lceil \log(M + 1) \rceil$. For the total cost, this needs to be multiplied by the number of iterations needed for the phase estimation

$$\mathcal{I} = \lceil \pi\lambda/(2\epsilon_{\mathrm{PEA}}) \rceil. \tag{45}$$

Next we consider the costs for the number of logical qubits.

1. The control register for the phase estimation needs $\lceil \log(\mathcal{I}+1) \rceil$ qubits. The unary iteration to control on this register needs another $\lceil \log(\mathcal{I}+1) \rceil - 1$ qubits.
2. N qubits used for the system register.
3. There are $2n_M$ qubits used for the μ and ν registers.
4. There is an ancilla with $\aleph$ bits that is placed in an equal superposition.
5. We need another qubit for the $|+\rangle$ state to control the swap of the μ and ν registers.
6. In preparing the equal superposition state we also need to perform a rotation on a single ancilla.
7. There are two qubits used for the spin registers in the prepared state.
8. One qubit is used for flagging success of the inequality tests in the state preparation.
9. One qubit is used to flag if $\nu = M+1$.
10. There are $\beth$ qubits for the phase gradient state.
11. The contiguous register has size $\lceil \log d \rceil$.
12. The QROM has qubit cost (including the output) of $mk_{s1} + \lceil \log d/k_{s1} \rceil$. Of these, $m(k_{s1}-1) + \lceil \log d/k_{s1} \rceil$ are temporary ancillas, which can be reused later. That is, $\lceil \log d/k_{s1} \rceil$ temporary ancillas for QROM, and $m(k_{s1}-1)$ are extra outputs that may be erased by measuring in the X basis. If the costs in the following steps are smaller, then we can ignore them and use the cost of the QROM here instead.
13. The size of the data output for the rotations is $\beth N/2$, and there will be $\lceil \log M \rceil$ temporary ancillas as well.
14. There are $\beth - 2$ temporary qubits when adding into the phase gradient state for rotations, which will be larger than $\lceil \log M \rceil$ in the previous step for the systems of interest.
15. The ancilla costs for erasing the QROM can be ignored because they can reuse ancillas that were previously erased.

This costing gives us a total number of logical qubits

$$2\lceil \log(\mathcal{I}+1) \rceil + N + 2n_M + \beth + \lceil \log d \rceil + \aleph + 5 + \max\left(mk_{s1} + \lceil \log d/k_{s1} \rceil,\ m + \beth N/2 + \beth - 2\right). \tag{46}$$

E. Error metrics for approximate tensors

The evaluation of Hamiltonian approximation errors, i.e., ϵ_{THC}, is critical for precisely estimating the resource requirements of any quantum simulation method based on tensor factorizations, including tensor hypercontraction and the low-rank methods. Moreover, to fairly compare against other methods such as the single factorization (SF), double factorization (DF), and the sparse method, one should use a consistent error metric for all. We take the perspective that one should estimate the error in the *exact* ground-state energy made by approximating the Hamiltonian matrix elements since preparing ground states is very often the goal. Obviously, estimating the error in the exact ground-state energy is not feasible for problems whose ground-state computations are not classically tractable. We list below desired properties, which a good error metric should satisfy:

1. An error metric should be classically tractable to evaluate so that computing it for a handful of medium-sized systems is relatively straightforward.
2. An error metric should be size extensive and size consistent. Size extensivity and size consistency are important properties of the exact ground state wave function and energy [86,87]. Violating these properties often results in poor approximations of the exact ground state wave functions and therefore these are important properties to preserve. Size extensivity asserts the asymptotic scaling of the energy to be linear in system size and size consistency asserts the product separability of wave functions and the additivity of energies when two subsystems are infinitely far part. Therefore, any error metrics that attempt to bound the ground-state energy should satisfy these two properties.
3. An error metric should provide a reliable bound on the exact ground state energy error and also be well correlated with the actual error in the energy. By well correlated, we mean that improving the error metric should lead to the improvement in the ground-state energy error as well. This is generally very difficult to meet and also verify because, in general, we do not know the exact ground-state energy.

We summarize existing error metrics here and discuss which one is most suitable for our purposes. We refer to the approximate integral tensor as $\widetilde{V}$ and keep the discussion of error metrics as general as possible so that it can be applied to any form of approximation methods.

In von Burg *et al.* [10], two error metrics were introduced and advocated: (1) coherent error (ϵ_{co}) and (2) incoherent error (ϵ_{in}). In fact, these are the 1-norm and 2-norm measures of the difference between the exact and approximate integral tensors; i.e.,

$$\epsilon_{\mathrm{co}} \equiv \sum_{pqrs} \left| V_{pqrs} - \widetilde{V}_{pqrs} \right|, \quad \epsilon_{\mathrm{in}} \equiv \sqrt{\sum_{pqrs} \left| V_{pqrs} - \widetilde{V}_{pqrs} \right|^2}. \tag{47}$$

The incoherent scheme provides a rigorous bound to the shift in the ground-state energy but we find these bounds to be too loose. In examples considered here, they are not well correlated with the error in the ground-state energy despite bounding that error. For instance, a slight improvement

in ϵ_{in} can lead to a drastic improvement in the ground-state energy estimate. More importantly, both ϵ_{co} and ϵ_{in} scale up to quartically with system size (at the very least superlinearly in general), which grows far too quickly with system size and violates the size extensivity criterion. Furthermore, ϵ_{in} is no longer size consistent. We believe that the violation of size extensivity and size consistency is what makes these error metrics often not well correlated with the error in the exact ground-state energy.

Berry *et al.* [9] advocated error metrics based on classical quantum chemistry methods called the second-order Møller-Plesset perturbation theory (MP2) and configuration interaction with singles and doubles (CISD). The MP2 correlation energy error metric can be thought of as a weighted signed difference between two tensors:

$$\epsilon_{\text{MP2}} = \sum_{pqrs} w_{pqrs} \left(V_{pqrs} - \widetilde{V}_{pqrs} \right), \tag{48}$$

where w_{pqrs} is a positive weight that is defined as the orbital energy differences defined for each quartet (p, q, r, s). It can be rigorously shown that Eq. (48) satisfies size consistency and size extensivity since the MP2 correlation energy satisfies these [87]. We note that the correlation energy in a finite basis is defined as the energy difference between an approximate method and a mean-field approach (Hartree-Fock) in the same basis [86]. Unfortunately, the MP2 correlation energy behaves quite erratically in many cases [88] so the error bound based on the MP2 correlation energy may not be well correlated with the actual error in the exact ground-state energy. Since there are other classical approaches that go beyond MP2, one may consider other options as well. The CISD correlation energy error metric is the option that Berry *et al.* [9] chose. Unfortunately, the CISD correlation energy is neither size extensive nor size consistent. Consequently, in the limit of infinite system size, the CISD correlation energy approaches zero.

In this work, we advocate the use of an error metric based on the correlation energy error in the coupled cluster with singles, doubles, and perturbative triples [CCSD(T)] approach [89]. CCSD(T) is so-called the "gold standard" method in classical simulations of quantum chemistry. While its quantitative accuracy on strongly correlated systems is often doubtful, it is still the best classical method that satisfies criteria (1) and (2) above. Whether it meets criterion (3) is again difficult to assess, but for some Hamiltonians for which CCSD(T) is qualitatively accurate, it is reasonable to expect that (3) is satisfied. Furthermore, CCSD(T) contains the MP2 wave-function contribution in it in the sense that some diagrams contained in CCSD(T) exactly correspond to those of MP2. Therefore, we think that measuring the errors made by approximating the Hamiltonian elements based on CCSD(T) would be representative of classical simulation methods. Ultimately, it is possible that the errors in the CCSD(T) correlation energy are not well correlated with the errors in the exact ground-state energy. We leave more precise understanding of these error metrics for future study and focus on comparing different approaches based on the CCSD(T) correlation energy error. For strongly correlated Hamiltonians, one can consider high-spin states where single determinant wave functions provide a good description. In this case, the CCSD(T) energy must be well correlated with the exact high-spin eigenstates. As long as the underlying integral approximation does not assume the underlying spin state (which is the case for all approximations considered here), this should facilitate a fair comparison between the methods discussed in this work.

In addition to the inherent factorization error, it is useful to further account for the error made by the approximation of the Hamiltonian coefficients (ζ) as part of coherent alias sampling and the individual qubit rotations via χ. This is achieved by first making approximate representations for χ and ζ for fixed numbers of bits to represent these tensors, $\beth$ and $\aleph$, respectively. That is, χ is represented by a sequence of rotations with angles $\{\theta_p^{(\mu)}\}$ and these angles are approximated by the limited precision given by $\beth$. More specifically, a rotation vector, $\vec{\chi}^{(\mu)}$ is parametrized by a set of angles $\{\theta_p^{(\mu)}\}$, which are defined recursively [10] as

$$\chi_p^{(\mu)} \left[\vec{\theta}^{(\mu)} \right] = \cos(2\theta_p^{(\mu)}) \Pi_{q<p} \sin(2\theta_q^{(\mu)}). \tag{49}$$

Next, we define units for θ and ζ,

$$u_\theta = \frac{2\pi}{2^{\beth}}, \qquad u_\zeta^o = \frac{\lambda_z}{d 2^{\aleph}}, \qquad u_\zeta^d = \frac{\lambda_z}{d 2^{\aleph - 1}}, \tag{50}$$

where u_θ is the unit for θ, u_ζ^o is the unit for off-diagonal elements of ζ, and u_ζ^d is the unit for diagonal elements of ζ. Finally, write approximate χ and ζ as $\tilde{\chi}$ and $\tilde{\zeta}$, respectively, by

$$\begin{aligned} \tilde{\chi}_p^{(\mu)} &= \chi_p{}^{(\mu)} \left[u_\theta \times \text{round}(\vec{\theta}^{(\mu)} / u_\theta) \right] \\ \tilde{\zeta}_{\mu\nu} &= \begin{cases} u_\zeta^o \times \text{round}(\zeta_{\mu\nu} / u_\zeta^o + x) & \text{for } \mu \neq \nu \\ u_\zeta^d \times \text{round}(\zeta_{\mu\mu} / u_\zeta^d + x) & \text{for } \mu = \nu \end{cases}, \end{aligned} \tag{51}$$

where x is a small constant to ensure the normalization of $|\tilde{\zeta}|$. We employ a sextic polynomial fit for $x \in [-1, 1]$ to find the optimal x that normalizes $|\tilde{\zeta}|$. The resulting λ of $|\tilde{\zeta}|$ is normalized to 1 with a small error on the order of 10^{-6}. These can be used to build an approximate representation $\widetilde{G}$ for G. In addition to evaluating the CCSD(T) correlation energy error using G, we also evaluate the error using $\widetilde{G}$ to provide more precise resource estimates.

IV. RESOURCE ESTIMATES FOR REAL SYSTEMS

A. Resource estimates for simulating active-space models of FeMoCo molecule

Keeping with the tradition of the work by Reiher *et al.* [23], Berry *et al.* [9], and von Burg *et al.* [10], here we benchmark our methods on the problem of simulating the ground state of active-space models of FeMoCo. While this is not the only quantum chemistry problem worth studying on a quantum computer, it is easier to compare to prior methods if we study the same molecule. We look forward to also deploying our method to some of the catalysis Hamiltonians studied by von Burg *et al.* [10] as soon as they are made available by those authors. Unfortunately, the first of these papers by Reiher *et al.* used an active space for FeMoCo that was later found to be problematic for the ground-state simulation by Li *et al.* [36]. The ground state of the active space proposed by Reiher *et al.* does not capture the open-shell nature (i.e., strong correlation) of the FeMoCo model cluster and therefore the gold standard CCSD(T) calculation was found to be off by only 5 millihartree from the near-exact density matrix renormalization group (DMRG) energy [36]. Therefore, in their work, Li *et al.* propose an alternative active space. Whereas the original Reiher Hamiltonian involved 108 qubits, the integrals by Li *et al.* involve 152 qubits and is thus a more challenging problem. The work by Berry *et al.* estimates the cost for both Hamiltonians but here we focus on assessing the resources that would be required to simulate the Li *et al.* Hamiltonian, as the more accurate active space.

In Fig. 8 we analyze the dependence of the shift in the ground-state energy on the THC rank M. For the Reiher *et al.* Hamiltonian, as mentioned above, the gold standard CCSD(T) is quantitatively accurate so it is quite sensible to assess the error of approximate integral tensors based on the error in the CCSD(T) correlation energy. The ground-state computation involves $N = 108$ qubits and 54 electrons and the target spin state is $S = 0$ (singlet). We employ spin-restricted Hartree-Fock orbitals for this Hamiltonian. As we can see, we achieve chemical accuracy for $M \geq 250$.

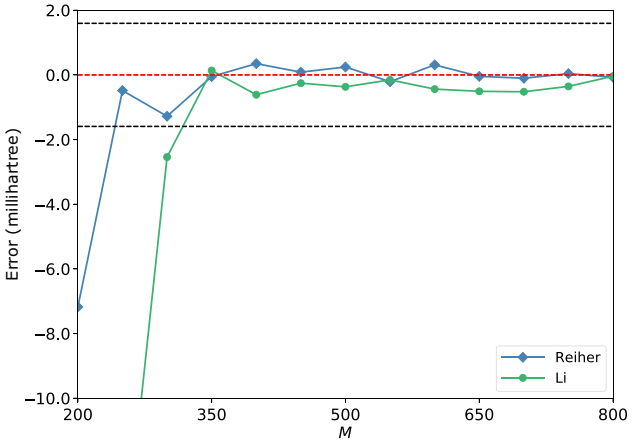

FIG. 8. Error in the CCSD(T) correlation energy (millihartree) for the Reiher *et al.* [23] and the Li *et al.* [36] Hamiltonian as a function of the THC rank M. Black dotted lines indicate chemical accuracy of 1.6 millihartree. Chemical accuracy for both Hamiltonians is achieved for $M \geq 350$. Note that the energy improvement is not monotonic in M in part due to difficulties associated with the nonlinear optimization of THC factors.

For the Li Hamiltonian, there are 152 spin orbitals and 113 electrons in this active-space model. For this Hamiltonian, CCSD(T) is no longer quantitatively accurate for low-spin states (such as the ground state $S = 3/2$) due to its inability to capture the antiferromagnetically coupled open-shell nature. However, if we consider a high-spin state such as $S = 35/2$, CCSD(T) with spin-unrestricted Hartree-Fock orbitals should be quantitatively accurate compared to the exact energy of the $S = 35/2$ state. Furthermore, there is no aspect in the THC factorization (or other methods compared in this work) specialized for a specific spin state. Therefore, we believe that the CCSD(T) error metric for the $S = 35/2$ is well correlated with the actual error made by integral approximations in the exact $S = 3/2$ ground-state energy. As shown in Fig. 8, THC achieves chemical accuracy for $M \geq 350$.

For both Hamiltonians, we assess the CCSD(T) correlation energy error with approximate rotation angles and ζ. We find that 10 bits for state preparation and 16 bits for rotations are enough for the Reiher Hamiltonian, whereas 10 bits for state preparation and 20 bits for rotations are needed for the Li Hamiltonian. The resulting CCSD(T) correlation energy error, λ, Toffoli count, and logical qubit count are available in Table IV for the Reiher Hamiltonian and Table V for the Li Hamiltonian. Given these data,

TABLE IV. THC costing for the Reiher Hamiltonian with 10 bits for state preparation and 16 bits for rotations. For our analysis of this Hamiltonian, we use $M = 350$, which is the highlighted entry.

M	CCSD(T) Error (mE_h)	λ	Toffoli count	Logical qubits
250	−1.31	294.1	4.4×10^9	1115
300	−1.12	302.8	4.9×10^9	1183
350	−0.29	306.3	5.3×10^9	2142
400	−0.18	315.1	5.6×10^9	2144
450	0.13	327.9	6.1×10^9	2144
500	0.03	339.2	6.6×10^9	2146
550	−0.07	343.0	7.1×10^9	2278
600	−0.10	347.8	7.6×10^9	2278
650	−0.29	361.4	8.2×10^9	2278
700	−0.10	365.1	8.7×10^9	2278
750	−0.09	373.6	9.3×10^9	4327
800	−0.04	380.2	9.7×10^9	4327

TABLE V. THC costing for the Li Hamiltonian with 10 bits for state preparation and 20 bits for rotations. For our analysis of this Hamiltonian, we use $M = 450$, which is the highlighted entry.

M	CCSD(T) Error (mE_h)	λ	Toffoli count	Logical qubits
350	0.39	1279.0	3.2×10^{10}	2194
400	−1.14	1258.4	3.2×10^{10}	2196
450	−0.18	1201.5	3.2×10^{10}	2196
500	−0.49	1214.9	3.3×10^{10}	2196
550	−0.08	1161.2	3.3×10^{10}	2328
600	−0.16	1140.8	3.4×10^{10}	2328
650	−0.09	1132.2	3.5×10^{10}	2328
700	−0.39	1119.8	3.6×10^{10}	2328
750	−0.23	1114.4	3.6×10^{10}	4377
800	−0.32	1123.7	3.8×10^{10}	4377

we recommend $M = 350$ for the Reiher Hamiltonian and $M = 450$ for the Li Hamiltonian.

B. Resource estimates and scaling analysis for hydrogen chain and H_4 benchmarks

In order to study the scaling of these methods in this section we analyze a chemical series consisting of hydrogen chains. We use the same chemical series as the one studied in Ref. [9] in order to facilitate a direct comparison with that work: one-dimensional hydrogen chains with a spacing of 1.4 Bohr. For the finite-size scaling, we use the STO-6G minimal basis set and grow the system by adding more hydrogens. Denoting the number of hydrogens in a chain by N_H, we study the chemical series from $N_H = 10$ to $N_H = 100$. When estimating thermodynamic asymptotes, it is often convenient to fix the error per particle as opposed to the total error for every system size [90]. Indeed, the most widely used integral factorization, the resolution-of-the-identity approximation, typically yields an error per atom to be 50–60 μhartrees [90]. We follow a similar standard in this section.

For the hydrogen-chain calculations, we use a threshold of 5×10^{-5} for the sparse method, 0.0015 for the single factorization method using the modified Cholesky factorization [91] in the atomic orbital basis (which yields roughly two Cholesky vectors per H atom), and 0.01 for the double factorization method [see Eq. (C41)]. For THC, we fix the THC rank to be $7 \times N_H$. These thresholds are enough to obtain the CCSD(T) correlation energy error per atom to be less than 50 μhartrees as shown in Table VI.

For H_4, we use the cc-pVTZ basis set [92] to obtain V and considered truncated Hamiltonians in the molecular orbital basis with a varying number of orbitals from $N/2 = 10$ to 56. The number of Cholesky vectors used in the SF method is fixed to be 300 and the THC rank is held at 576. This is enough to maintain the accuracy of 1–10 μhartrees for $N/2$ greater than 35 in all methods. With respect to the basis set size, the data size for state preparation scales cubically with basis set size in the case of the SF method and quadratically with basis set size for THC [73]. Therefore, we focus on the scaling of λ for SF and THC methods. We use a threshold of 5×10^{-5} for the sparse method and 1×10^{-4} for the DF method. On the other hand, the data size of sparse and DF methods varies with system so we obtain representation for each instance of the truncated Hamiltonian. This gives us the scaling of

TABLE VI. CCSD(T) correlation energy error per hydrogen in hartree of each hydrogen-chain system for various methods. We aim to reach 10–50 μhartrees for all methods but SF. For the SF method, the largest error below 50 μhartrees/H atom that we find is about 9 μhartrees.

N_H	Sparse	SF	DF	THC
10	3.9×10^{-8}	7.4×10^{-6}	3.5×10^{-5}	4.4×10^{-6}
20	1.2×10^{-6}	8.3×10^{-6}	3.4×10^{-5}	1.7×10^{-5}
30	5.3×10^{-6}	8.6×10^{-6}	4.6×10^{-5}	-2.9×10^{-7}
40	1.2×10^{-5}	8.8×10^{-6}	4.7×10^{-5}	1.2×10^{-5}
50	1.7×10^{-5}	8.9×10^{-6}	3.6×10^{-5}	5.4×10^{-6}
60	2.1×10^{-5}	9.0×10^{-6}	3.8×10^{-5}	5.1×10^{-5}
70	2.5×10^{-5}	9.0×10^{-6}	4.1×10^{-5}	1.9×10^{-5}
80	2.7×10^{-5}	9.0×10^{-6}	3.5×10^{-5}	4.9×10^{-6}
90	2.2×10^{-5}	9.1×10^{-6}	3.5×10^{-5}	2.2×10^{-5}
100	3.4×10^{-6}	9.1×10^{-6}	4.1×10^{-5}	-1.6×10^{-5}

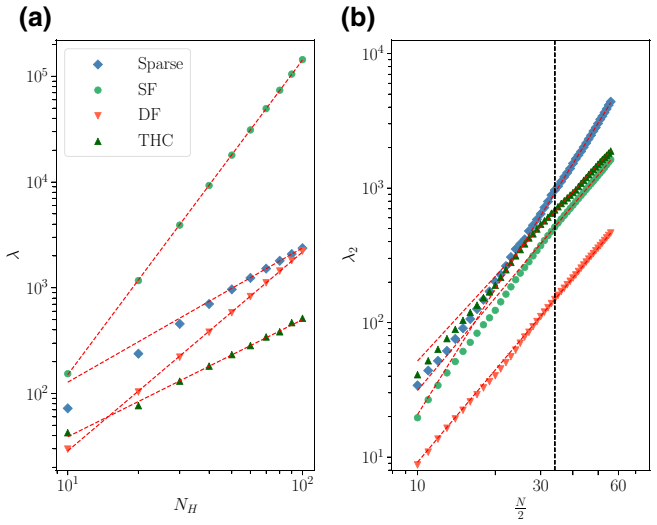

FIG. 9. (a) The number of H atoms (N_H) versus total λ values and (b) the number of orbitals ($N/2$) versus two-body λ values for different approaches. Note that these λ values in (a) include both one-body and two-body λ values relevant to each approach and the two-body λ_2 values in (b) are also approach-specific values. The vertical black dotted line in (b) is drawn at $N/2 = 36$ beyond which all lines behave linearly and associated CCSD(T) correlation energy errors are roughly constant. Red dotted lines represent the linear fits on a log scale for each data whose slopes are listed in Tables VII and VIII. Those slopes are used to compute scalings listed in Table II.

TABLE VII. Slopes of linear fits on a log scale to the hydrogen-chain data in Fig. 9(a).

Approach	Slope	Data range	R^2
Sparse	1.27	the last five points	0.9998
SF	2.98	all	1.0000
DF	1.88	all	0.9998
THC	1.11	exclude the first two and the last two points	0.9991

data size with respect to the basis set size. In all systems, the sparse method λ values are evaluated with localized orbitals using the Edmiston-Ruedenberg method [93] to exploit spatial locality. All other methods used canonical Hartree-Fock orbitals.

Values of λ for each method are presented in Fig. 9. In (a), we have the hydrogen-chain data from which we can infer the thermodynamic size scaling of different approaches. In (b), we have the H_4 two-body λ data from which we can find the continuum limit scaling of different approaches. Since H_4 is a very small system, the one-body contribution is comparable to the two-body contribution to λ. However, as we are interested only in asymptotes, we focus just on the two-body contribution because asymptotically the two-body term dominates λ. These scalings of λ are used to generate the Toffoli complexity scaling data in Table II. For the space complexity scaling, we need additional information for the sparse method (i.e., the number of nonzero elements) and the DF method (i.e., the average number of eigenvectors in the second factorization). These are provided in Appendices A and C, respectively. We perform linear fits to obtain the asymptotic scaling of each method appropriately. The results of these linear fits are summarized in Tables VII and VIII along with data range and R^2 values.

C. Surface-code compilation for the active-space model of FeMoCo by Li *et al.*

We now come to the problem of determining how much physical space and time is used by our computations if we plan to realize them assisted by quantum error correction within the surface code. Producing these estimates requires making assumptions about the performance characteristics of physical qubits and the classical control system making up the quantum computer. We assume a physical gate-error rate of 0.1%, a control system reaction time of 10 μs, and a surface-code cycle time of 1 μs. We also consider the more optimistic gate-error rate of 0.01% to give a point of comparison.

TABLE VIII. Slopes of linear fits on a log scale to the H_4 data in Fig. 9(b). All data points above $N/2 = 35$ are used in the linear fit.

Approach	Slope	R^2
Sparse	3.10	0.9993
SF	2.29	0.9998
DF	2.28	0.9999
THC	2.09	0.9991

1. Space and time constraints

In surface-code quantum computations, space-time trade-offs are ubiquitous, meaning one can often shorten the length of a computation at the expense of a higher physical qubit overhead and vice versa. A good first approximation is that a quantum computation is like a liquid that can be poured into any desired space-time volume. For example, often one can (roughly) halve the amount of time taken by a computation by doubling the number of magic state factories. Of course, the "uncompressible liquid computation volume" approximation does break down when pushed enough. For example, as the space available is reduced, the contortions needed to fit the computation become more and more challenging. The time has to increase by proportionally more, to accommodate the contortions. There is also a minimum number of qubits needed to store the system being simulated, which no amount of contortion will get below. Additionally, there are constraints that prevent the *time* of a quantum computation from being reduced arbitrarily far, such as the code distance d, the reaction time t, and details of the algorithm being run. To give the reader a better understanding of the layout decisions we made, we discuss these two constraints in more detail.

The code distance d determines the level of protection against errors one has, and is chosen based on a combination of the physical error rate and number of operations in the computation. For most practical code implementations, protection against timelike errors up to that distance d dictates a local physical stabilizer has to be measured $O(d)$ times before it is possible to error correct those measurements to a desired level of reliability. We call the amount of time it takes to run d rounds of the surface-code cycle a "beat." Any surface-code construction that involves changing which stabilizers are being measured will (usually) take at least one beat to complete, since that is how long it takes to become confident about the values of the new stabilizers.

The reaction time t measures how long it takes for the classical control system to receive the raw physical measurements corresponding to a logical measurement, error correct them to recover the logical measurement, and trigger a following logical measurement that is either in the X or Z basis depending on the value of the previous logical measurement. This quantity becomes relevant when performing non-Clifford operations via magic state distillation and gate teleportation. The gate teleportation will apply the desired operation, but will also randomly apply other Clifford operations that need to be undone

by corrective Clifford operations. If the corrective Clifford operations needed to finish a gate teleportation are applied "in place," by waiting for the correction to be known and then applying operations to the target qubits, then the correction process would have a time measured in beats. This is a strategy we have used in the past when laying out surface-code computations [16]. However, when using this strategy, it is difficult for the computation consuming magic states to keep pace with even a single magic state factory. A more time-efficient strategy is to precompute multiple world lines, one for each possible Clifford correction, and selectively teleport the affected qubits through the appropriate world line [27,94,95]. The selective teleportation can be controlled by measuring certain qubits in either the X or Z basis. Because logical X and Z measurements are implemented transversally, the time they take does not depend on the code distance, and so they are not beat limited. Instead, the limiting time is how quickly the control system can process the deluge of data coming at it and iteratively figure out what the next basis to measure in is.

It is not always feasible to execute one non-Clifford operation per reaction time. For example, there might not be enough magic state factories to produce one magic state per reaction time. To account for this, we introduce another time unit: the "tick." A tick is either the reaction time of the control system or the average period between the production of magic states; whichever is slower.

Optimally packing a quantum computation often involves balancing tick-based constraints and beat-based constraints. For example, the driving force of a QROM read is unary iteration sequentially preparing qubits flagging whether or not the address register is equal to each possible address value [16]. Unary iteration is made up of a series of interdependent Toffolis; it is primarily tick constrained. On the other hand, the flag qubits prepared by unary iteration are used as controls for huge multitarget CNOTs reaching into the target data registers. These CNOTs are done via lattice surgery, which involves changing which stabilizers are being measured, and so the CNOTs are primarily beat constrained. Optimally packing a QROM computation into space time requires balancing the beat-based constraints from the multitarget CNOTs and the tick-based constraints from the unary iteration.

2. QROM trade-offs

The performance in the surface code depends not only on the raw operations, but also on the layout design of the qubits. Suppose we lay out data qubits into columns with every third column left empty as an access hallway. In this layout, each data qubit has a single side exposed to a hallway. We call this layout "single-side storage." When using single-side storage, a multitarget CNOT will monopolize the single side of its target qubits for one beat. A basic QROM read performs one multitarget CNOT per possible address value, and therefore when using single-side storage a basic QROM read with n addresses will take at least n beats to complete.

If n beats is too slow (e.g., if it is significantly slower than the tick-limited unary iteration), we could instead use a storage layout where every second column is left empty as an access hallway. In this "double-side storage" layout, every qubit has two exposed sides. This allows a second CNOT to start before the first has finished, by targeting the second side, lowering the minimum time required for an n-address QROM read from n beats to $n/2$ beats. If $n/2$ beats is still too slow, alternative QROM circuits can reduce the number of multitarget CNOTs by an "output expansion factor" $k < \sqrt{n}$ by using k times more workspace qubits [9,38]. These alternative circuits also lower the number of Toffoli gates required, reducing the tick-based lower bound from the unary iteration.

There are other possibilities for optimizing the packing of a QROM read, but for the physical parameter regime we are interested in there are three relevant design choices. (1) Should we use single- or double-sided storage? (2) How many magic state factories should there be? (3) What should the expansion factor k be, for each QROM read? After some experimentation and iteration we settled on using single-side storage, four magic state factories, and an expansion factor of 16 for the QROM read during the PREPARE subroutine and 1 for the QROM reads during the SELECT subroutine. These decisions are based on looking at the algorithm's resource utilization at a global level, for various possible choices. We now describe how that is analyzed.

3. Diagram-driven decisions

When laying out a quantum algorithm in the surface code, there are two diagrams we recommend making. The first diagram is a floorplan, or footprint, of where the various parts of the computation will live. Our final floorplan diagram is in Fig. 10. The goal of the floorplan diagram is not to find a perfectly optimal layout, but rather to get a rough idea of how things will fit together and how much space will be needed to ensure it is possible to route data and magic states into operating areas quickly enough to keep the computation going.

The second useful diagram to make is an inventory of allocated qubits over time, which shows how many qubits are allocated as the algorithm progresses and also what they are being used for. Our final diagram of this type is in Fig. 11. This diagram can reveal spikes where too many qubits are being used, and holes where a lot of qubits are available for use (e.g., as additional magic state factories).

The reason these diagrams are useful to make is that understanding resource utilization allows one to make optimizations. For example, elsewhere in this paper we

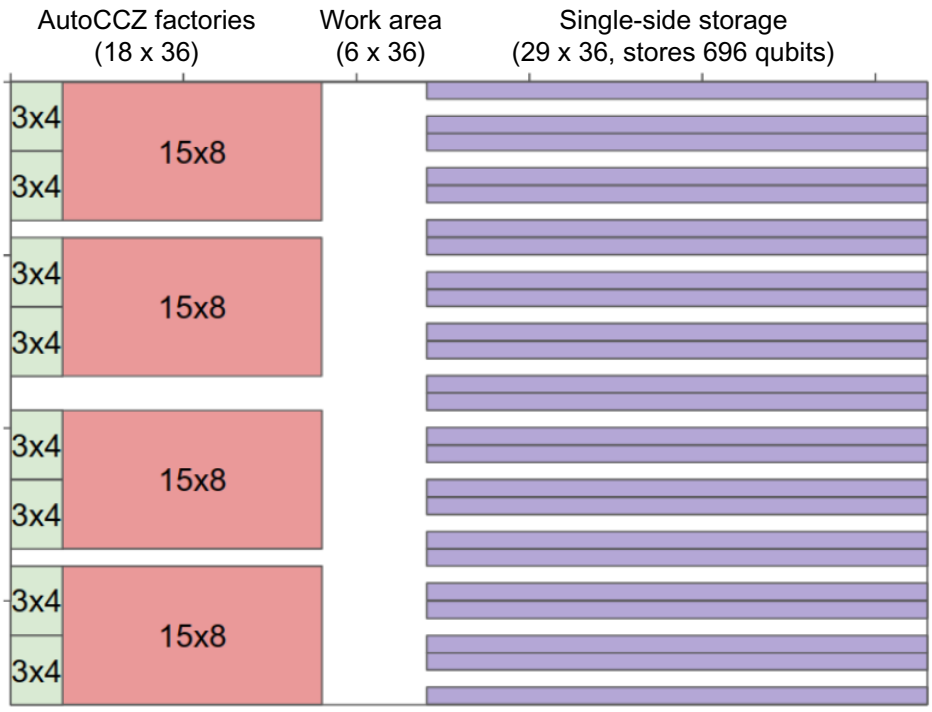

FIG. 10. Floorplan for the THC-based FeMoCo qubitized phase-estimation computation using the Li *et al.* Hamiltonian [36]. Covers $53 \times 36 = 1908$ logical qubits. The code distance is 31. The number of physical qubits is $1908 \times 2 \times 32^2 \approx 4 \times 10^6$. Red areas are the CCZ factory from [95], with a level-1 code distance of 19 and a level-2 code distance of 31. Green areas are for the AutoCCZ fixup box from Ref. [95]. Purple areas are for data qubit storage. White areas are for routing, performing Clifford gates, and teleporting magic states into data qubits.

note that the phasing operations during the SELECT subroutine will use angles loaded from QROM. Originally, we intended for all the angles to be loaded simultaneously from one QROM read. This avoids redundant QROM reads from the same address register and correspondingly minimizes the number of Toffolis. However, when plotting allocated qubits over time as in Fig. 12, it became obvious that loading the angles is much faster than using them, and so loading half of the angles at a time would not cost much more overall. The benefit of loading half of the angles at a time is that half as many output qubits are needed. This is a substantial space savings for an acceptable time loss.

Because of the lower number of qubits available, we also have to adjust the output expansion factor of the QROM read during the prepare subroutine. We reduce it from 32 to 16, which again increases the Toffoli count slightly but substantially reduces the workspace needed. We also change the reflections to be directly controlled by the control qubits used for phase estimation, instead of using the unary iteration circuit shown in Fig. 2, which needs extra ancillas. Contrast the original allocation plan in Fig. 12 with the more space-efficient allocation plan in Fig. 11. In short, because we made a diagram showing allocated qubits over time, giving us a global view of our resource usage, we spotted a reasonably simple optimization that reduced the number of data qubits by 40%. This is why we recommend making these diagrams.

4. Routing and distilling

We now consider the routing of data during the phasing operations within the SELECT subroutine. These phasing operations are implemented by adding qubits output from a QROM read into a phase-gradient register, controlled by a target qubit. This process creates an amount of phase kickback proportional to the amount read from QROM, and this phase kickback acts on the target qubit due to it being used as a control. Because the phase-gradient register is used for every single phasing operation, the phase-gradient register should be moved out of the storage area and kept in the operating area. The data being added into the phase-gradient register then needs to be gradually streamed out of the storage area as the phasing operations are applied. The majority of this data is the data that was produced by the QROM read. In order to avoid traffic jams, it is important that data that will be needed at the same time be placed down separate hallways. The QROM circuit construction gives us full control over where the target data lives, so this is not a difficult constraint to meet.

In the past, we have had trouble laying out quantum computations in a way that could consume incoming magic states quickly enough [16]. This is because the layout strategy we were using involved performing the probabilistic Clifford operations resulting from this process directly on the qubits being acted on by the magic states. We are now using a different strategy, based on using AutoCCZ states, which attach the corrections to the magic state instead of to the target qubits [27,94,95]. When using AutoCCZs, the corrective operations can be packed into space time far away from the data qubits being operated on, allowing the computation to progress at a reaction-limited rate instead of a beat-limited rate. This removes a significant constraint we were previously operating under. Instead of struggling to keep pace with one magic state factory, we could now,

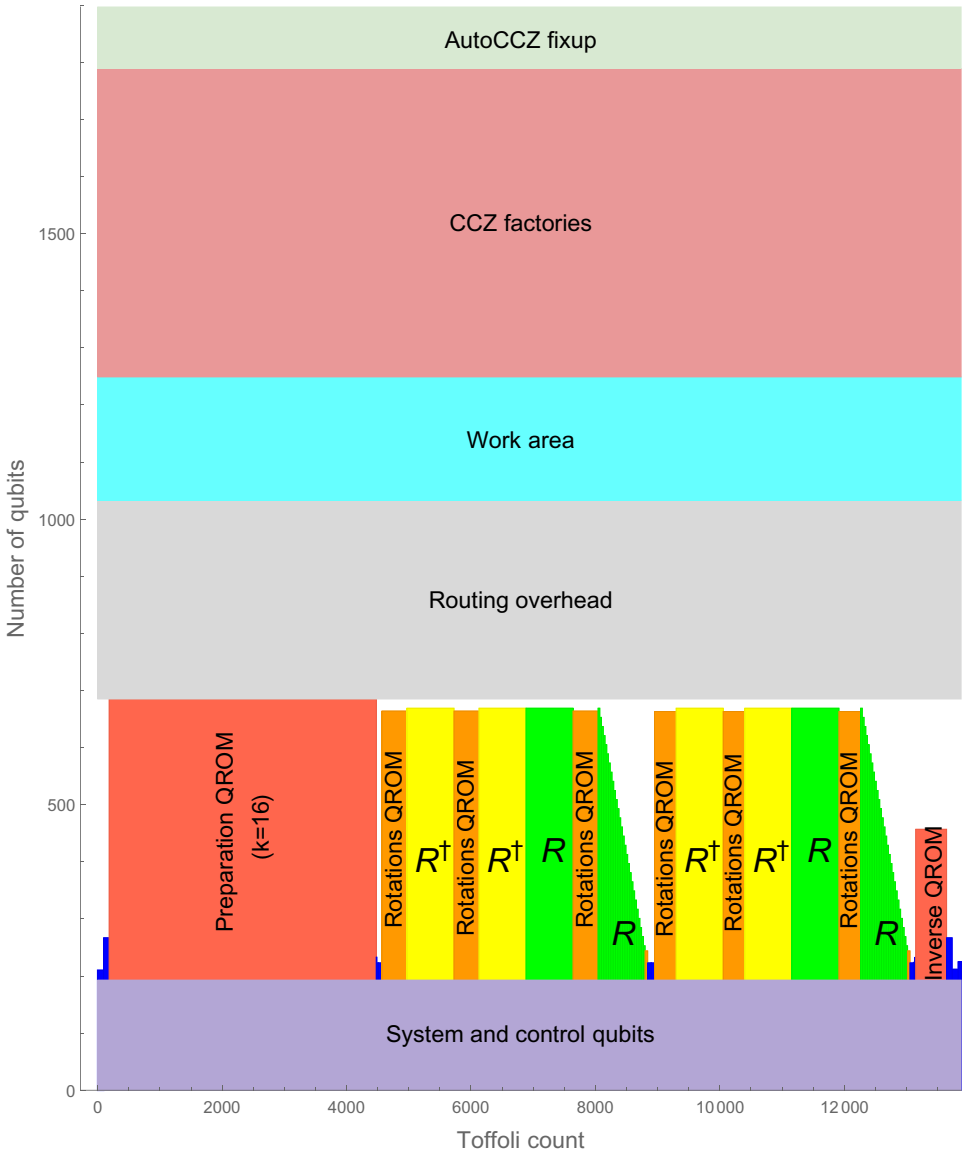

FIG. 11. An improved strategy for the inner loop of the computation, where half of the phasing angles are loaded at a time. The output expansion factor during the first QROM read has correspondingly been decreased from 32 to 16.

in principle, keep pace with ten or more (if we are willing to pay the qubit cost of operating so many factories).

The magic state factory we use is the CCZ factory from Ref. [95]. Note that results from Ref. [96] suggest the factory can be reduced in size, due to the distillation process heralding topological errors. We do not incorporate these results into our factory designs, but intend to do so in the future.

After exploring several cases using the estimation spreadsheet from the ancillary files of Ref. [54], tweaked slightly to account for the improvements to the factory made in Ref. [95] and for the presence of multiple factories, we decided to use four factories with a level-1 code distance of 19 and a level-2 code distance of 31. According to the "logical_factory_dimensions" method from the "estimate_costs.py" script in the ancillary files of Ref. [97], the large level-1 code distance results in the factory having a footprint of $15d \times 8d$ and a depth of $5d$ where $d = 31$ is the level-2 code distance. This choice gives a roughly 0.1% chance of any distillation failure occurring when executing ten billion Toffolis, allocates nearly a million physical qubits for use as factories, and results in a Toffoli production rate of 25 kHz. We give each factory a row of routing space to move the produced CCZ states to where they are needed, and two 3×4 AutoCCZ fixup areas for correcting the teleportation of the magic states being produced.

As shown in Fig. 11, the number of data qubits we store when running FeMoCo is less than 700. The number of Toffoli operations is less than ten billion, and we perform Toffolis at a rate of 25 kHz. According to the spreadsheet from Ref. [54], a data-code distance of 31 is sufficient for this regime while maintaining a 1% total error budget. We summarize this information in Fig. 10, which shows one potential way to lay out the factories and the data qubits. With this layout the FeMoCo computation would span four million physical qubits.

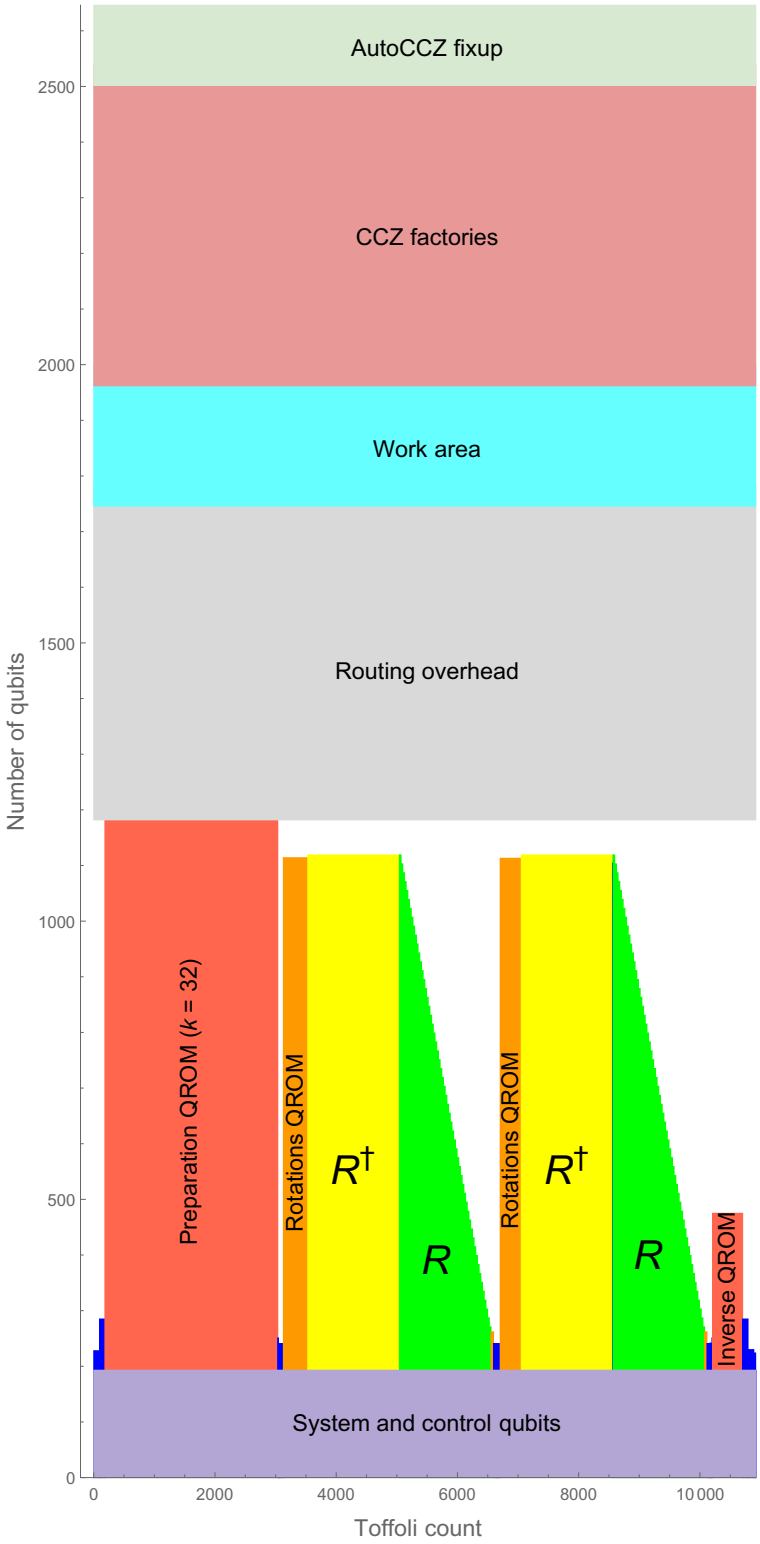

FIG. 12. The original strategy we intended to use for the inner loop of the computation. The angles loaded from QROM in order to perform phasing operations are all loaded at the same time, to minimize the number of angle-loading QROM reads (in orange). It is clear from the diagram that the QROM reads are not the dominant cost; the diagram suggests performing redundant reads while storing fewer angles.

The total Toffoli count of the algorithm is approximately 483 000 inner loops times 13 880 Toffolis per inner loop, which equals 6.7 billion Toffolis. It would take 3 days to perform these Toffolis with four CCZ factories distilling at a total rate of 25 kHz.

5. Estimate at optimistic error rates

So far in this section, our estimates are based on conservatively assuming a physical error rate of 0.1% and a corresponding error suppression factor Λ of 10 (meaning we assume the logical error rate per round goes down by a

factor of 10 each time we increase the code distance by 2). If we optimistically assume physical error rates of 0.01%, and a corresponding Λ of 100, all of the code distances can be cut in half while achieving the same reliability.

We again refer to the spreadsheet from Ref. [54]. We find that, under a total error budget of 1% and a physical error rate of 0.01%, ten billion Toffolis can be executed using a level-1 distillation code distance of 9, a level-2 distillation code distance of 15, and a data-code distance of 15. The resulting CCZ factory has a footprint of $14d \times 8d$, proportionally slightly better than the $15d \times 8d$ we had with a physical error rate of 0.1%. Therefore we can simply use the floorplan from Fig. 10 unchanged, but using $d = 15$ instead of $d = 31$. Since the physical qubit count per logical qubit is $2(d+1)^2$, reducing the code distance from 31 to 15 reduces the physical qubit count by a factor of $32^2/16^2 = 4$. The physical qubit count shrinks from four million to one million.

Because the computation is still beat limited at distance 15 when using four factories (as opposed to being reaction limited) the execution time at distance 15 is the execution time at distance 31 multiplied by 15/31. The execution time shrinks from 3.5 days to 1.5 days.

V. CONCLUSIONS

The primary contribution of this paper is to introduce a method for simulating electronic Hamiltonians in arbitrary basis on a quantum computer. In terms of both the asymptotic scaling and the finite resources required for complex benchmark molecules, our method appears to have lower cost for realization on a fault-tolerant quantum computer than any prior algorithm in the literature. We then compile these circuits and perform an analysis of error-correction overheads to arrive at the most accurate estimates yet of what would be required to realize an important and classically intractable quantum computation of chemistry within the surface code.

The method we introduce uses a number of now standard techniques for quantum simulation, including phase estimation of quantum walks, qubitization, QROM, coherent alias sampling, and unary iteration. Our key innovation is to adapt this set of tools to the tensor hypercontraction representation of quantum chemistry, which had previously gone unexplored in the context of quantum computing. While the standard THC representation compresses the Hamiltonian considerably, enabling highly efficient quantum walks, using that representation directly leads to very large λ values, requiring many repetitions of the quantum walk. To avoid this, we use tensors obtained from THC to transform the standard electronic Hamiltonian into a representation that has a diagonal Coulomb operator but in a larger and nonorthogonal auxiliary basis. We then develop algorithms for realizing the associated qubitization oracles, which work by rotating into the correct basis one tensor factor at a time, thus avoiding the need for a global nonorthogonal-basis rotation, and allowing us to apply our algorithm without using any more system qubits than what is required by the original Hamiltonian.

The most efficient prior algorithms for simulating arbitrary basis quantum chemistry were the "sparse" method of Berry *et al.* [9] and the "double low rank" method of von Burg *et al.* [10]. In both cases we correct errors in the original work (leading to reduced Toffoli complexity of the former approach). The sparse algorithm has Toffoli complexity $\widetilde{\mathcal{O}}[(N+\sqrt{S})\lambda_V/\epsilon]$ and space complexity $\widetilde{\mathcal{O}}(N+\sqrt{S})$ where S is the Hamiltonian sparsity. The double low rank algorithm has Toffoli complexity $\widetilde{\mathcal{O}}(N\lambda_{\mathrm{DF}}\sqrt{\Xi}/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N\lambda_{\mathrm{DF}}\sqrt{\Xi})$ where Ξ is the rank of the second tensor factorization discussed for quantum computing in Ref. [30]. By contrast, the tensor hypercontraction approach introduced here has Toffoli complexity $\widetilde{\mathcal{O}}(N\lambda_\zeta/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N)$. To contextualize the scaling of these λ values as well as S and Ξ we analyze the asymptotic scaling of these methods applied to hydrogen systems growing towards both continuum and thermodynamic limits. Of the two prior algorithms, the double low rank method scales better towards the continuum limit with Toffoli complexity $\widetilde{\mathcal{O}}(N^{3.9}/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N^{1.5})$. The THC algorithm has a favorable scaling with Toffoli complexity of $\widetilde{\mathcal{O}}(N^{3.1}/\epsilon)$ and space complexity of $\widetilde{\mathcal{O}}(N)$. When scaling towards the thermodynamic limit the sparse algorithm scales considerably better than the double low rank algorithm, having Toffoli complexity $\widetilde{\mathcal{O}}(N^{2.3}/\epsilon)$ and space complexity $\widetilde{\mathcal{O}}(N)$. Again, our THC algorithm still has even lower scaling with the same asymptotic space complexity but Toffoli complexity of $\widetilde{\mathcal{O}}(N^{2.1}/\epsilon)$. See Table II for details.

Next, we compare the finite resources required to implement these methods for popular FeMoCo benchmarks. As can be seen in Table III, the THC algorithm has both fewer logical qubits and less Toffoli complexity than any of the other competitive methods. We note that there might be more compact and accurate THC factors than those we find, which could improve our results even further. Due to the difficulties associated with nonlinear optimization, we expect that our solutions are suboptimal. In this sense, the estimates we report for the cost of the THC approach should be regarded as upper bounds on the cost of the most efficient possible implementations. We discuss a very detailed scheme for efficiently laying out the Li Hamiltonian FeMoCo computation within the surface code. Ultimately, we show that the computation could be realized with about four million physical qubits and under 4 days of runtime, assuming physical gate-error rates of about 0.1%. With the more optimistic assumption of 0.01% per gate-error rates we could realize the same computation with about one million physical qubits and under 2 days of runtime.

It is interesting to note that, in our physical cost estimates, most qubits are used for routing and distillation. Quantum circuits describing a quantum algorithm typically omit these qubits, which make up the majority of the cost. For example, consider modifying a random-access algorithm to use sequential access to save space by reducing routing overheads. If avoiding random access forced a trade-off versus the number of data qubits, the abstract circuit model would classify this improvement as a downgrade. Because of this, we caution the reader that comparisons derived purely from quantum circuit metrics (such as counting Toffolis and data qubits) can be misleading. They are good approximations, but these approximations can fail in known ways.

Despite the rapid progress detailed in Table I, we expect that we are nearing the end of a series of asymptotic speedups for arbitrary basis quantum chemistry algorithms. At least for second quantized approaches based on linear combinations of unitaries, it would be difficult to imagine an algorithm that improves more than logarithmically on the $\widetilde{\mathcal{O}}(N\lambda/\epsilon)$ Toffoli complexity achieved by our approach (although it is possible that one could reduce λ). This is because it would seem that $\Omega(N)$ complexity should be required for any nontrivial quantum walk on N qubits and phase estimation of LCU methods generically requires $\Omega(\lambda/\epsilon)$ repetitions. Indeed, we suspect that the near $\widetilde{\mathcal{O}}(N^2/\epsilon)$ complexity obtained when scaling towards the thermodynamic limit for hydrogen chains would be the lowest possible scaling for simulation of the Coulomb operator as a consequence of its pairwise nature; also, this roughly matches the lowest scaling that has been achieved for simulating the Coulomb operator in special basis sets [98]. However, perhaps there is room to improve over the $\widetilde{\mathcal{O}}(N^3/\epsilon)$ scaling we observe when scaling hydrogen systems towards their continuum limit. Still, there are several ways that we might hope to extend these methods and reduce constant factors. We emphasize that these asymptotic scalings should be further investigated in a more general setup than hydrogenic systems in the future, though we expect that the same asymptotes will be observed in a much larger system.

While the THC results presented in this work show remarkable improvements over previous qubitization approaches, the nonlinear optimization associated in obtaining the THC factorization is challenging for broad applications. Future research and some ongoing effort in quantum chemistry may help to resolve this issue [73,74]. One natural question to ask is whether the THC representation [and in particular, our nonorthogonal THC Hamiltonian in Eq. (12)] has utility for quantum computing beyond qubitization-based methods; e.g., can one combine the THC representation with Trotter methods? Another question is whether one can leverage hierarchical matrix representations of the Coulomb operator [99] to compress the Hamiltonian in a way that is useful for qubitization either within the THC framework or within the sparse method. Another area to consider would be developing strategies of choosing active-space orbitals with an aim towards reducing the value of λ associated with qubitizing the resultant Hamiltonian.

Similar to other prior papers on quantum computing for chemistry, the current work focuses on eigenstate preparation with the assumption of a sufficiently good initial guess state. The usual justification for this is that: (1) for most small molecules near their equilibrium geometry, simple initial states such as the Hartree-Fock state have reasonably good overlap with the ground state, (2) more complex state-preparation procedures such as adiabatic state preparation may have a negligible additive cost to the cost of phase estimation, which could produce a better overlapping initial state, and (3) as long as the overlap is not too small and especially when one has further knowledge of the gap there are methods for improving the $\mathcal{O}(1/a)$ scaling with the initial state overlap a [14,100]. In Ref. [101], the authors argued that this state-preparation cost should often increase exponentially as one grows systems towards a thermodynamic limit, and this might also be expected (at least in the worst case) from the QMA hardness of the electronic structure problem [102]. However, in practice this asymptotic scaling may not matter because correlation lengths are finite and we tend to need only to perform very accurate correlated calculations on finite-sized systems. Nevertheless, for very challenging systems such as FeMoCo, better methods of state preparation might be required. Future work should place more emphasis on analyzing and account for those costs.

Finally, we should continue to identify more concrete molecular benchmarks beyond the capabilities of classical electronic structure methods that could be solved on a quantum computer to provide insights about chemistry. Noting that by combining qubitization with quantum signal processing [103] one can adapt our approach to perform highly efficient time evolutions rather than phase estimation, it is also worth exploring applications of quantum chemistry that would benefit from time evolution of the electronic Hamiltonian.

A. Data and code availability

To maximize reproducibility, we share data and code used in this work on a public Zenodo repository [76]. The repository includes all of the integral tensors (i.e., both exact and approximate) for systems studied in this work, python codes for generating the DF factors, and computing λ values as well as Mathematica notebooks for evaluating the cost of the SF, DF, sparse, and THC methods. Since THC factorization done in this work involves challenging nonlinear optimization, it may be difficult for others to reproduce our numerical THC data (although it is easily verified). Thus, we recommend that interested readers use the available THC factors on Ref. [76] for future study.

ACKNOWLEDGMENTS

The authors thank Nicholas Rubin for helpful discussions, the authors of Ref. [10] and Ref. [36] for sharing FeMoCo molecular integrals used in their work, and Michael Streif for pointing out a mistake in the one-body λ value of hydrogen chains. 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. N.W. is funded by a grant from Google Quantum AI, and his theoretical work on randomized simulation is supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-Design Center for Quantum Advantage under Contract No. DE-SC0012704, support for N.W.'s other theoretical contributions came from the Pacific Northwest National Laboratory LDRD program and the "Embedding Quantum Computing into Many-Body Frameworks for Strongly Correlated Molecular and Materials Systems" project, funded by the US Department of Energy (DOE). J.L. thanks David Reichman for encouragement and support.

AUTHOR CONTRIBUTIONS

J.L. and R.B. proposed the idea to combine tensor hypercontraction with qubitization. J.L. performed tensor hypercontraction analysis and all quantum chemistry numerics and proposed the nonorthogonal representation. D.W.B. developed and analyzed quantum algorithm implementations with contribution from R.B., including the qubitization technique for simulating the nonorthogonal representation. C.G. contributed the surface-code layout analysis. N.W. wrote most of Appendix D. W.J.H., N.W., and J.R.M. proposed and investigated a scheme for simulating the nonorthogonal representation by using quantum linear algebra tools, which was ultimately found to be less efficient than the one described. W.J.H. and J.R.M. also participated in many discussions and made useful suggestions. The paper was written by J.L., D.W.B., and R.B. with contributions from the others. Collaboration was organized and managed by R.B.

APPENDIX A: THE "SPARSE" ALGORITHM OF BERRY *ET AL.*

1. Representing the sparse Hamiltonian as a linear combination of unitaries

Here we review the method of Ref. [9], with some further optimizations of the method. The first step in simulating a Hamiltonian using qubitization is to represent it as a linear combination of unitaries. How one chooses this linear combination of unitaries has significant ramifications for the λ factor, which will scale the cost of the qubitized simulation. We start by expressing the electronic Hamiltonian in an arbitrary second-quantized basis as in Eq. (1). We can map these operators to qubits as

$$\begin{aligned} T &= \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T_{pq}\left(a^\dagger_{p,\sigma}a_{q,\sigma}+a^\dagger_{q,\sigma}a_{p,\sigma}\right)\\ &= \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T_{pq}Q'_{pq\sigma} \end{aligned} \tag{A1}$$

and

$$\begin{aligned} V &= \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{p,q,r,s=1}^{N/2} V_{pqrs}\left(a^\dagger_{p,\alpha}a_{q,\alpha}+a^\dagger_{q,\alpha}a_{p,\alpha}\right)\\ &\quad\times\left(a^\dagger_{r,\beta}a_{s,\beta}+a^\dagger_{s,\beta}a_{r,\beta}\right)\\ &= \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{p,q,r,s=1}^{N/2} V_{pqrs}Q'_{pq\alpha}Q'_{rs\beta}, \end{aligned} \tag{A2}$$

where the Jordan-Wigner transform that is used is expressed as

$$\begin{aligned} Q'_{pq\sigma} &= \begin{cases} X_{p,\sigma}\vec{Z}X_{q,\sigma}, & p<q,\\ Y_{p,\sigma}\vec{Z}Y_{q,\sigma}, & p>q,\\ \mathbb{1}-Z_{p,\sigma}, & p=q,\end{cases}\\ a^\dagger_{p,\sigma}a_{q,\sigma}+a^\dagger_{q,\sigma}a_{p,\sigma} &\mapsto \frac{X_{p,\sigma}\vec{Z}X_{q,\sigma}+Y_{p,\sigma}\vec{Z}Y_{q,\sigma}}{2},\\ a^\dagger_{p,\sigma}a_{p,\sigma} &\mapsto \frac{\mathbb{1}-Z_{p,\sigma}}{2}, \end{aligned} \tag{A3}$$

where X, Y, and Z are the Pauli operators, the subscripts indicate the qubits these operators act on, and $A_p\vec{Z}A_q$ is shorthand for $A_pZ_{p+1}\cdots Z_{q-1}A_q$. Note that we get a factor of $1/8$ in front of Eq. (A2) from multiplying the original factor of $1/2$ by a factor of $1/4$ that comes from expanding the pq and rs terms as the sum of their Hermitian conjugate for each index. Note that we have made a correction of a factor of 2 from $Q_{pq\sigma}$ as defined in Ref. [9].

An improvement we can make is to remove the identity from the case where $p=q$, so that $Q_{pq\sigma}$ has the same weightings with on-diagonal and off-diagonal terms, as

$$Q_{pq\sigma} = \begin{cases} X_{p,\sigma}\vec{Z}X_{q,\sigma}, & p<q,\\ Y_{p,\sigma}\vec{Z}Y_{q,\sigma}, & p>q,\\ -Z_{p,\sigma}, & p=q.\end{cases} \tag{A4}$$

Then the expansion of T can be written as

$$T = \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T_{pq}Q_{pq\sigma} + \sum_{p=1}^{N/2} T_{pp}\mathbb{1}, \tag{A5}$$

and the expansion of V can then be written as

$$V = \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\left(\sum_{p,q,r,s=1}^{N/2} V_{pqrs}Q_{pq\alpha}Q_{rs\beta} + \sum_{p,r,s=1}^{N/2} V_{pprs}Q_{rs\beta} + \sum_{p,q,r=1}^{N/2} V_{pqrr}Q_{pq\alpha} + \sum_{p,r=1}^{N/2} V_{pprr}\mathbb{1}\right). \tag{A6}$$

Here the middle two terms correspond to one-body terms that can be combined with T, and the fourth is a constant offset that can be omitted. We can therefore write the Hamiltonian in terms of a new T' and V', given by

$$T' = \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T'_{pq}Q_{pq\sigma}, \tag{A7}$$

$$V' = \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{p,q,r,s=1}^{N/2} V_{pqrs}Q_{pq\alpha}Q_{rs\beta}, \tag{A8}$$

with

$$T'_{pq} = T_{pq} + \sum_{r=1}^{N/2} V_{pqrr}. \tag{A9}$$

Then $H = T' + V'$ plus a term proportional to the identity, which can be omitted because it gives a constant shift to the eigenvalues.

Because the operator $Q_{pq\sigma}$ as well as products $Q_{pq\alpha}Q_{rs\beta}$ are all unitary operators we now see that $H = T' + V'$ is a linear combination of unitaries. In this representation the associated λ values are

$$\lambda = \lambda_T + \lambda_V \qquad \lambda_T = \sum_{p,q=1}^{N/2}\left|T_{pq} + \sum_{r=1}^{N/2} V_{pqrr}\right|,$$
$$\lambda_V = \frac{1}{2}\sum_{p,q,r,s=1}^{N/2}\left|V_{pqrs}\right|. \tag{A10}$$

For T' the multiplying the factor of $1/2$ cancels with a factor of 2 for summing the spin degree of freedom. The factor of $1/2$ in front of the V term comes from multiplying the factor of $1/8$ by 4 for summing the spin degrees of freedom.

Note that these λ definitions differ from those given in Ref. [9]. At first glance λ_V appears to be a factor of 8 smaller. The first reason for this is because we deviate from the convention of absorbing the factor of $1/2$ into V (since this is just a difference in convention, it is consistent with prior work). However, there is roughly another factor of 4 difference that comes in because the work of Ref. [9] accidentally left out a factor of $1/2$ in Eq. (A3). This leads to a λ_V that is reduced by a factor of 4. For the Reiher Hamiltonian [23], we previously reported $\lambda_T = 1{,}490$ a.u., $\lambda_V = 8{,}373$ a.u. so $\lambda = 9{,}863$ a.u. with a truncation threshold of 2×10^{-4}. This should be updated to $\lambda_T = 90$ a.u., $\lambda_V = 2{,}045$ a.u. so $\lambda = 2{,}135$ a.u. with a truncation threshold of 7.5×10^{-5}. For the Li Hamiltonian [36] we previously gave $\lambda_T = 3{,}446$ a.u., $\lambda_V = 4{,}168$ a.u. so $\lambda = 7{,}614$ a.u. with a truncation threshold of 1×10^{-4}. This should be updated to $\lambda_T = 561$ a.u., $\lambda_V = 986$ a.u. so $\lambda = 1{,}547$ a.u. with a truncation threshold of 3.5×10^{-5}. The truncation threshold became tighter in this work because the metric we choose [i.e., CCSD(T) correlation energy error] is more conservative than what was used in our previous work [9].

2. The cost of qubitization of the sparse chemistry Hamiltonian

The state to be prepared is similar to that in Eq. (48) of Ref. [9], except T_{pq} is replaced with T'_{pq} and V_{pqrs} is replaced with $V_{pqrs}/2$ (due to the factor of 2 in the definition of V), so the state to be prepared is

$$\begin{aligned}&|0\rangle\,|+\rangle\,|0\rangle\,|0\rangle \\ &\otimes\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2}\sqrt{\frac{|T'_{pq}|}{\lambda}}\,|\theta^T_{pq}\rangle\,|p,q,\sigma\rangle\,|0,0,0\rangle \\ &+|1\rangle\,|+\rangle\,|+\rangle\,|+\rangle \\ &\otimes\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{p,q,r,s=1}^{N/2}\sqrt{\frac{|\widetilde{V}_{pqrs}|}{2\lambda}}\,|\theta^V_{pqrs}\rangle\,|p,q,\alpha\rangle\,|r,s,\beta\rangle\,.\end{aligned} \tag{A11}$$

This state can be prepared as described in Ref. [9], using controlled swaps to generate the symmetries of the state.

Because we are here not including the identity in $Q_{pq\sigma}$, the controlled unitaries can be performed in a simpler way than shown in Fig. 1 of Ref. [9]. In that work there are inequality tests between the p and q registers, which are needed to produce the sum of the identity and Z operations. When the identity is not included, then the circuit needed simplifies to just two applications of the circuit for Majorana operators shown in Fig. 9 of Ref. [16] and Fig. 1 of [104]. The simplified circuit is shown in Fig. 13. When the SELECT is controlled as shown, then the complexity is $2(N-1)$. If it does not need to be controlled, then the complexity is only $2(N-2)$. We require only one of these two SELECT operations to be controlled, because for the case of the one-body term in the Hamiltonian only one SELECT should be applied, so one of the SELECT needs to be controlled on the qubit flagging one- and two-body terms in the Hamiltonian. The total complexity of these SELECT operations is therefore $4N-6$.

In the sparse simulation method, the relevant parameters are the number of orbitals N, the λ value, and the number of unique nonzero entries d. If we are allowing error in the

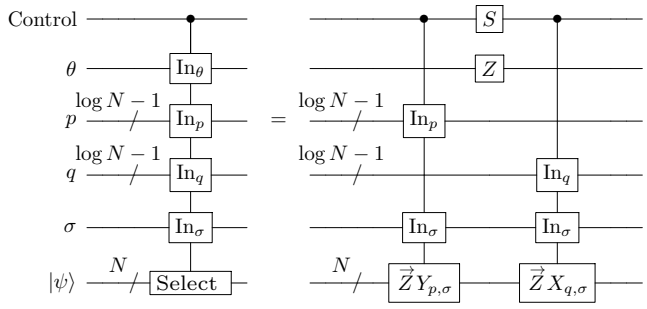

FIG. 13. The circuit needed to perform a controlled SELECT operation. This is similar to that in Ref. [9], except it is not necessary to perform the equality test $p = q$. The unitaries labeled as $\overrightarrow{Z}A_j$ apply the operation $Z_0 \cdots Z_{j-1}A_j$ to the target register, depending on the value from the input register, using the technique shown in Fig. 9 of Ref. [16]. The sign is shown as being obtained via a Z gate on the sign qubit, but in practice this would be obtained as part of the state preparation.

energy due to the state preparation of ϵ_{PREP}, the output size for the keep probabilities for the QROM is

$$\aleph = \left\lceil \log\left(\frac{\lambda}{2\epsilon_{\text{PREP}}}\right)\right\rceil. \tag{A12}$$

There are eight registers of size $n_N = \lceil \log(N/2) \rceil$, because the sparse preparation scheme needs to output ind values and alt values of p, q, r, s. There are also 2 qubits needed for the two output values of θ (one for the ind and one for the alt values of p, q, r, s), as well as two qubits used for ind and alt values of the first register, which distinguishes between T and V. As a result the QROM output size is

$$m = \aleph + 8n_N + 4. \tag{A13}$$

The cost of the preparation is then

$$\lceil d/k_1 \rceil + m(k_1 - 1) \tag{A14}$$

and of the inverse preparation is

$$\lceil d/k_2 \rceil + k_2. \tag{A15}$$

To begin the state preparation, we need to prepare an equal superposition state over d basis states. Given that 2^η is a factor of d, the procedure and its costs are as follows.

1. Perform an inequality test on $\lceil \log d \rceil - \eta$ bits with $\lceil \log d \rceil - \eta - 1$ Toffolis.
2. Rotate an ancilla qubit to obtain overall amplitude for success of $1/2$ using b_r bits of precision. This has cost $b_r - 3$ Toffolis (since the rotation angle is given classically).
3. Reflect on success for both, which may be performed via a controlled phase gate (no Toffolis).
4. Invert the rotation with cost $b_r - 3$.
5. Invert the inequality test, which may be performed with Cliffords provided the first inequality test was performed using the out-of-place adder.
6. Perform a reflection on $\lceil \log d \rceil - \eta + 1$ qubits with cost $\lceil \log d \rceil - \eta - 1$.
7. Perform the inequality test again, with cost $\lceil \log d \rceil - \eta - 1$.

The total cost is then $3\lceil \log d \rceil - 3\eta + 2b_r - 9$. This is a cost paid both for the preparation and for the inverse preparation.

Other minor Toffoli costs are as follows. In the following we can use extra ancillas to save cost, because a large number of ancillas are used for the QROM, and can be reused here without increasing the maximum number of ancillas used.

1. Perform SELECT as shown in Fig. 13 twice, with only one being controlled. As discussed above, this has complexity $4N - 6$.
2. The state preparation needs an inequality test on $\aleph$ qubits, as well as controlled swaps. The cost of the inequality test on $\aleph$ qubits is $\aleph$, and by performing an inequality test with the out-of-place adder we can invert it in the inverse preparation with no additional Toffoli cost. The controlled swaps are on $4n_N + 1$ qubits. Here the $4\lceil \log(N/2) \rceil$ are for the values of p, q, r, and s, and the $+1$ is for the qubit, which distinguished between the one- and two-electron terms. It is not necessary to perform a controlled swap on the sign qubits output, because the correct phase can be applied with Clifford gates. It is possible to invert the controlled swaps in the inverse preparation with Cliffords. The method is to copy all values being swapped before they are swapped. Then to invert the controlled swap, perform measurements on the swapped values in the X basis. We can perform phase fixups using controlled-phase operations, where the control is the control qubit for the controlled swaps, and the targets are the copies of the registers. That means we can eliminate the non-Clifford cost of the inverse preparation, giving a Toffoli cost of $\aleph + 4n_N + 1$.
3. The controlled swaps used to generate the symmetries have a cost of $4n_N$. Again these controlled swaps can be inverted for the inverse preparation with measurements and Clifford gates.
4. A reflection on the ancilla is needed as well. The qubits that need to be reflected on are the $\lceil \log d \rceil$ qubits needed for preparing the state, the $\aleph$ qubits used for the equal superposition state in the coherent alias sampling, the three qubits that are used for controlled swaps to generate the symmetries of the state, and the ancilla qubit that is rotated to produce the

equal superposition state. That gives a Toffoli cost of $\lceil \log d \rceil + \aleph + 2$.

5. For each step one more Toffoli is needed for the unary iteration used for the phase estimation, and one more Toffoli is needed to make the reflection controlled.

Adding all these minor costs together gives

$$2(3\lceil \log d \rceil - 3\eta + 2b_r - 9) + (4N - 6) + (\aleph + 4n_N + 1) + 4n_N + \lceil \log d \rceil + \aleph + 4 = 4N + 8n_N + 2\aleph + 7\lceil \log d \rceil - 6\eta + 4b_r - 19. \tag{A16}$$

The total cost for a single step is then

$$\left\lceil \frac{d}{k_1} \right\rceil + m(k_1 - 1) + \left\lceil \frac{d}{k_2} \right\rceil + k_2 + 4N + 8n_N + 2\aleph + 7\lceil \log d \rceil - 6\eta + 4b_r - 19, \tag{A17}$$

with $m = \aleph + 8n_N + 4$, $n_N = \lceil \log(N/2) \rceil$, and η an integer such that 2^η is a factor of d.

The logical qubits used are 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 N qubits.
3. The QROM uses a state with $\lceil \log d \rceil$ qubits.
4. A qubit is needed to flag success of the equal superposition state preparation.
5. An ancilla qubit is rotated in the preparation of the equal superposition state.
6. The phase-gradient state uses b_r qubits.
7. The equal superposition state used for the coherent alias sampling, which has $\aleph$ qubits.
8. The QROM uses qubits (including the output) $mk_1 + \lceil \log(d/k_1) \rceil$.

That gives a total number of logical qubits

$$2\lceil \log(\mathcal{I} + 1) \rceil + N + \lceil \log d \rceil + b_r + \aleph + mk_1 + \lceil \log(d/k_1) \rceil + 1, \tag{A18}$$

with $m = \aleph + 8n_N + 4$.

3. Counting the number of permutation-unique elements

The two-electron integral tensor V_{pqrs} exhibits an eightfold permutational symmetry:

$$V_{pqrs} = V_{pqsr} = V_{qprs} = V_{qpsr} = V_{rspq} = V_{rsqp} = V_{srpq} = V_{srqp}. \tag{A19}$$

To evaluate the cost of the sparse method presented above, we need to count the number of permutation-unique elements above a given truncation threshold. We provide more details about how this counting is performed so that others can easily reproduce the numerical data presented here. We note that the number of permutation-unique elements in V_{pqrs} is given as [86]

$$\frac{1}{8}\left(\frac{N}{2}\right)\left(\frac{N}{2}+1\right)\left[\left(\frac{N}{2}\right)^2 + \frac{N}{2} + 2\right]. \tag{A20}$$

This expression can be obtained by considering the following four different classes of V_{pqrs}:

1. p, q, r, and s are all unique indices. We loop over a total of $\binom{N/2}{4}$ unique combination of quartets (p, q, r, s) and count V_{pqrs}, V_{prqs}, and V_{psrq} in this category.
2. Two indices are redundant (i.e., only three unique indices). We loop over a total of $\binom{N/2}{3}$ unique combination of triplets (p, q, r) and count V_{ppqr}, V_{pqpr}, V_{qqpr}, V_{qrpq}, V_{rrpq}, and V_{rpqr} in this category.
3. Three indices are redundant (i.e., only two unique indices). We loop over a total of $\binom{N/2}{2}$ unique combination of doublets (p, q) and count V_{ppqq}, V_{pqpq}, V_{pppq}, and V_{qqqp} here.
4. All four indices are redundant. We loop over p and count V_{pppp} in this category.

In order to obtain d, we enumerate each category and count the number of elements above a threshold. Adding $(1/2)(N/2)(N/2 + 1)$ to account for the symmetry-unique

TABLE IX. Sparse method data for the Reiher Hamiltonian. Here d is the number of permutation-unique nonzero elements above a given threshold, and λ_T is 90.4 hartree. The entry in blue corresponds to the threshold used in our resource estimates.

Threshold	d	λ	CCSD(T) Error (mE_h)
0.001	122980	1445.0	−190.61
0.0005	233787	1736.8	−16.76
0.00025	391732	1954.4	0.84
0.0002	449699	2005.4	2.04
1.00×10^{-4}	633943	2110.5	0.73
8.75×10^{-5}	668079	2123.2	0.63
7.50×10^{-5}	705831	2135.3	0.33
5.00×10^{-5}	796197	2157.6	0.22
2.50×10^{-5}	916649	2175.2	0.05
1.00×10^{-5}	1014792	2181.9	0.0027
5.00×10^{-6}	1055829	2183.2	0.0003
2.50×10^{-6}	1078952	2183.5	0.0011
1.00×10^{-6}	1094260	2183.6	0.0004

TABLE X. Sparse method data for the Li Hamiltonian. Here d is the number of permutation-unique nonzero elements above a given threshold, and λ_T is 561.5 hartree. The entry in blue corresponds to the threshold used in our resource estimates.

Threshold	d	λ	CCSD(T) Error (mE_h)
0.001	45201	1345.9	−637.18
0.0005	78643	1403.5	−239.20
0.00025	131232	1452.8	−64.44
0.0001	239085	1504.9	−8.05
5.0×10^{-5}	359942	1534.7	−1.49
4.5×10^{-5}	382082	1538.6	−1.18
4.0×10^{-5}	408391	1542.7	−0.88
3.5×10^{-5}	440501	1547.3	−0.47
3.0×10^{-5}	480468	1552.2	−0.26
2.5×10^{-5}	532212	1557.6	−0.16
1.0×10^{-5}	881193	1579.2	0.03
5.0×10^{-6}	1259007	1589.8	0.01
2.5×10^{-6}	1722770	1596.4	0.01
1.0×10^{-6}	2410637	1600.9	0.005

part of the one-body operator to this directly gives us the numerical data in Tables IX and X.

4. Numerical data for hydrogen chains and H_4

In Fig. 14, we present the number of nonzero values of V with a truncation threshold, 5×10^{-5}, for hydrogen chains and H_4.

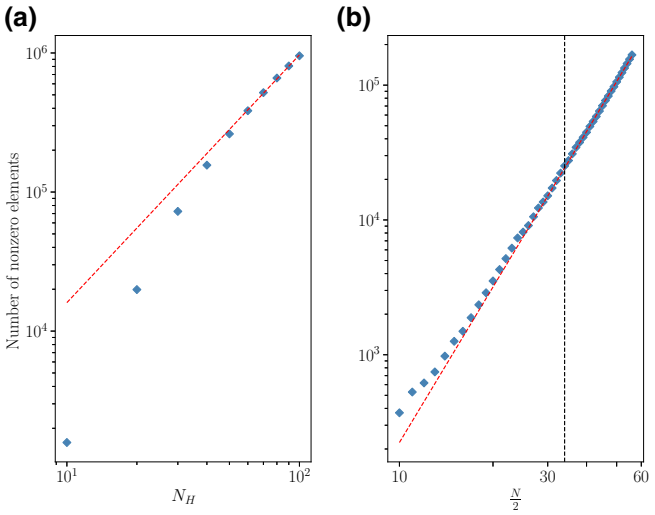

FIG. 14. (a) The number of H atom (N_H) versus the number of nonzero elements in V and (b) the number of orbitals ($N/2$) versus the number of nonzero elements in V. The last five points in (a) are used in the linear fit on a log scale where the slope is given as 1.78 with $R^2 = 0.9985$ (where R is the correlation coefficient). In (b), every point beyond $N/2 = 35$ is used for the linear fit on a log scale, which yields a slope of 3.83 with $R^2 = 0.9993$.

APPENDIX B: THE "SINGLE LOW RANK FACTORIZATION" ALGORITHM OF BERRY *ET AL.*

1. Representing the single low rank factorized Hamiltonian as a linear combination of unitaries

In Berry *et al.* the approach focuses on simulating a truncated low-rank decomposition of the Coulomb operator, which expresses the two-body terms in Eq. (1) as a sum of squared one-body terms in a form that was described for quantum computation in Ref. [48]. One work to suggest exploiting the low-rank properties of this decomposition was Ref. [30]. Specifically the factorization of the Coulomb operator used in that work is

$$V \approx W = \frac{1}{2}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} a_{p,\sigma}^{\dagger} a_{q,\sigma}\right)^2 = \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)}\left(a_{p,\sigma}^{\dagger} a_{q,\sigma} + a_{q,\sigma}^{\dagger} a_{p,\sigma}\right)\right)^2, \tag{B1}$$

where the $W_{pq}^{(\ell)}$ are scalars obtained by performing a Cholesky decomposition on a flattened version of the V_{pqrs} tensor and $L = \widetilde{\mathcal{O}}(N)$. Note that the factor of 1/2 becomes a factor of 1/8 because there is a factor of 1/2 that is then squared for adding the Hermitian conjugates. Using the Jordan-Wigner transform as in Eq. (A3) our Hamiltonian can be represented as the linear combination of unitaries:

$$W = \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} Q'_{pq\sigma}\right)^2. \tag{B2}$$

As before, when we separate out the identity for $p = q$, we can express this in terms of $Q'_{pq\sigma}$ from Eq. (A4) as

$$\begin{aligned} W &= \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} Q_{pq\sigma} + 2\sum_{r=1}^{N/2} W_{rr}^{(\ell)}\mathbb{1}\right)^2 \\ &= \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} Q_{pq\sigma}\right)^2 \\ &\quad + \frac{1}{2}\sum_{\ell=1}^{L}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q,r=1}^{N/2} W_{pq}^{(\ell)} W_{rr}^{(\ell)} Q_{pq\sigma} \\ &\quad + \frac{1}{2}\sum_{\ell=1}^{L}\left(\sum_{r=1}^{N/2} W_{rr}^{(\ell)}\mathbb{1}\right)^2 \end{aligned}$$

$$= \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} Q_{pq\sigma}\right)^2 + \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q,r=1}^{N/2} V_{pqrr} Q_{pq\sigma} + \frac{1}{2}\sum_{\ell=1}^{L}\left(\sum_{r=1}^{N/2} W_{rr}^{(\ell)}\mathbb{1}\right)^2 . \tag{B3}$$

In the last line we use that $\sum_\ell W_{pq}^{(\ell)} W_{rs}^{(\ell)} = V_{pqrs}$, which is exact when L is taken to be sufficiently large. In this expression, the second term can be combined with T to give a one-body operator T' that is identical to that in Appendix A,

$$T' = \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2}\left(T_{pq} + \sum_{r=1}^{N/2} V_{pqrr}\right) Q_{pq\sigma} . \tag{B4}$$

The third term in Eq. (B3) is proportional to the identity, and can be omitted. The fundamental two-body term in W is then just the first term of Eq. (B3),

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

The Hamiltonian to be simulated is $T' + W'$.

We can now give the expression for $\lambda = \lambda_T + \lambda_{\mathrm{SF}}$. Because T' is the same as in Appendix A, λ_T is as in Eq. (A10). For λ_{SF} we take the sum of the weightings in W' to give

$$\lambda_{\mathrm{SF}} = \frac{1}{4}\sum_{\ell=1}^{L}\left(\sum_{p,q=1}^{N/2}\left|W_{pq}^{(\ell)}\right|\right)^2 . \tag{B6}$$

There is a factor of 4 due to summing over the spins, which would give a factor out the front of 1/2. However, then this λ can be effectively divided by 2 if we choose to realize the squared operator by performing oblivious amplitude amplification as they do in Ref. [10]. After oblivious amplitude amplification of the operator A we effectively implement $2A^2 - \mathbb{1}$ where we can ignore the identity. Since this gives us a factor of 2 we can divide the λ by 2, giving λ_{SF} as in Eq. (B6).

2. The cost of qubitization of the single low rank factorized Hamiltonian

Next we describe the costing of the implementation of the single low rank factorized Hamiltonian as in Ref. [9], except taking into account the amplitude amplification approach for reducing λ. The complete Hamiltonian written as a linear combination of unitaries is (ignoring an additive term proportional to the identity)

$$H \approx \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T'_{pq} Q_{pq\sigma} + \frac{1}{8}\sum_{\ell=1}^{L}\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} Q_{pq\sigma}\right)^2 , \tag{B7}$$

where T'_{pq} is as in Eq. (A9). The procedure as given in Ref. [9] corresponds to preparing a state, performing controlled operations, then inverting the preparation. Now, we need to perform the state preparation separately on two registers. First, prepare a superposition over the first register, which selects between the terms in the factorization, as well as the one-body term. Next, prepare a superposition on the second register with weights corresponding to the square roots of $W_{pq}^{(\ell)}$ and T_{pq}. Perform the SELECT operation by using the circuit shown in Fig. 1 of Ref. [9]. Then invert the state preparation on the second register, reflect on that register, then perform the preparation, SELECT, and inverse preparation again.

The steps are shown in Fig. 15, and in detail are as follows.

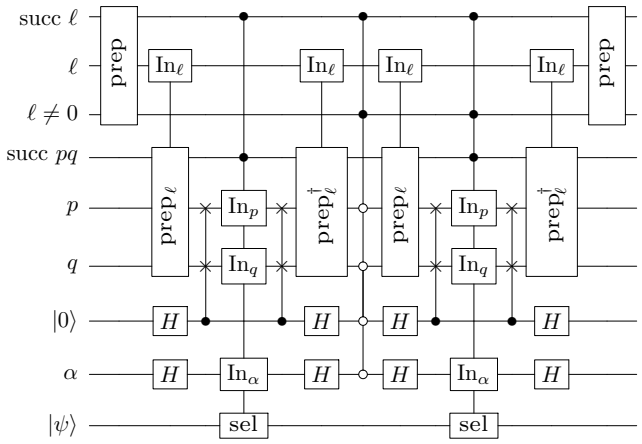

FIG. 15. The circuit for performing the state preparation and controlled operations for the single factorization approach. The register labeled ℓ is the first control register containing ℓ, and p, q labels the second and third control registers. The registers labeled succ ℓ and succ pq are the registers that flag success of the preparation of the equal superposition state for the ℓ and p, q registers. The register labeled $\ell \neq 0$ is a temporary register used to keep the result of an inequality test checking that $\ell \neq 0$, which is used to ensure that the second half of the circuit has no effect for that value of ℓ, which is used to label the one-electron term. The register labelled $|0\rangle$ is used to control swaps of p and q. The register labeled α is used to select the spin (and is initialized as $|0\rangle$). The bottom register labeled $|\psi\rangle$ is the target system. The operation labeled SEL is the SELECT as shown in Fig. 1 of Ref. [9].

1. We first prepare a state on the first register as

$$\frac{1}{\sqrt{\lambda}}\left(|0\rangle\sqrt{2\sum_{p\le q}|T'_{pq}|}+\sum_{\ell=1}^{L}|\ell\rangle\sum_{p\le q}|W_{pq}^{(\ell)}|\right). \tag{B8}$$

This has complexity as follows.

(a) Preparing an equal superposition on $L+1$ basis states has complexity $3n_L-3\eta+2b_r-9$, where b_r is the number of bits used for the rotation on the ancilla,

$$n_L=\lceil\log(L+1)\rceil, \tag{B9}$$

and η is a number such that $L+1$ is divisible by 2^η.

(b) A QROM is applied with output size $b_L=n_L+\aleph_1+2$, where the extra two qubits are for outputting the qubit showing if $\ell=0$. The complexity is

$$\left\lceil\frac{L+1}{k_L}\right\rceil+b_L(k_L-1). \tag{B10}$$

(c) An inequality test is performed with complexity $\aleph_1$.

(d) A controlled swap is performed with complexity n_L+1.

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

$$\frac{1}{\sqrt{\lambda}}\left(|0\rangle\sum_{p\le q}\sqrt{2|T'_{pq}|}\,|\theta_{pq}^{(0)}\rangle\,|p,q\rangle+\sum_{\ell=1}^{L}|\ell\rangle \times\sqrt{\sum_{r\le s}|W_{rs}^{(\ell)}|}\sum_{p\le q}\sqrt{|W_{pq}^{(\ell)}|}\,|\theta_{pq}^{(\ell)}\rangle\,|p,q\rangle\right)|+\rangle\,|+\rangle\,, \tag{B11}$$

where $\theta_{pq}^{(\ell)}$ 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. The complexity of this state preparation is as follows.

(a) First, an equal superposition over p and q is prepared with $p\le q\le N/2$. Letting $n_N=\lceil\log(N/2)\rceil$ be the numbers of bits used for the p and q registers, the cost of preparing this equal superposition is as follows.

i. For testing the two inequalities the cost is n_N-1 for $q\le N/2$ and n_N for $p\le q$.

ii. An ancilla is rotated with cost b_r-3 for b_r bits of accuracy.

iii. A reflection is performed controlled on the result of the two inequality tests and the ancilla qubit, with cost 1.

iv. The inequality tests are inverted with Cliffords, and there is a cost of b_r-3 for another qubit rotation.

v. There is a reflection on the p and q registers and the ancilla with cost $2n_N-1$.

vi. The inequality tests are performed again with cost $2n_N-1$.

vii. Another Toffoli is used to give a single qubit flagging success for both inequality tests.

That gives a cost of $6n_N+2b_r-7$.

(b) Next, a contiguous register is computed from p, and q as

$$s=p(p-1)/2+q. \tag{B12}$$

In Appendix F we show that the complexity of computing $p(p-1)/2+q$ is $n_N^2+n_N-1$. We are making an improvement over the method in Ref. [9] by not computing a contiguous register including ℓ as well. This is because it is possible to apply the QROM to two registers (provided there are not restrictions like $\ell\le s$), as we show in Appendix G.

(c) Perform the QROM on the two registers as described in Appendix G with output of size $b_p=2n_N+\aleph_2+2$, with complexity

$$\left\lceil\frac{L+1}{k_{p1}}\right\rceil\left\lceil\frac{N^2/8+N/2}{k_{p2}}\right\rceil+b_p(k_{p1}k_{p2}-1). \tag{B13}$$

(d) Perform the inequality test with cost $\aleph_2$.

(e) Perform the controlled swap with the alt values with cost $2n_N$. The sign required for the sign qubits can be implemented with Cliffords as before.

3. Perform a controlled swap between the p and q registers, with cost n_N.

4. Perform SELECT as shown in Fig. 1 of Ref. [9]. As discussed in Appendix A the cost is $2(N-2)$ Toffolis when it does not need to be controlled.

5. Reverse steps 2 and 3, where the complexities are the same except the QROM complexity, which is changed to

$$\left\lceil\frac{L+1}{k'_{p1}}\right\rceil\left\lceil\frac{N^2/8+N/2}{k'_{p2}}\right\rceil+k'_{p1}k'_{p2}. \tag{B14}$$

6. Reflect on the qubits that were prepared in step 2. The qubits we need to reflect on are as follows.

   (a) The $2n_N$ qubits for the p and q registers.
   (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 amplitude amplification.
   (d) One for the spin.
   (e) One for controlling the swap between the p and q registers.

   That gives a total of $2n_N + \aleph_2 + 3$ qubits. The reflection needs to be controlled on the success of the preparation on the ℓ register, and $\ell \neq 0$, making the total cost $2n_N + \aleph_2 + 3$ Toffolis.
7. Perform steps 2 to 5 again, but this time $L+1$ is replaced with L in Eqs. (B13) and (B14). Also, the SELECT operation needs to be controlled on $\ell \neq 0$, which flags the one-body term. That requires another Toffoli.
8. Invert the state preparation on the ℓ register, where the complexity of the QROM is reduced to

$$\left\lceil \frac{L+1}{k'_L} \right\rceil + k'_L. \tag{B15}$$

9. To complete the step of the quantum walk, perform a reflection on the ancillas used for the state preparation. There are $n_L + 2n_N + \aleph_1 + \aleph_2 + 4$, where the qubits we need to reflect on are as follows.

   (a) The n_L qubits for the ℓ register.
   (b) The $2n_N$ qubits for the p and q registers.
   (c) The $\aleph_1$ qubits for the equal superposition state used for preparing the state on the ℓ register using the coherent alias sampling.
   (d) The $\aleph_2$ qubits for the equal superposition state for preparing the state on the p, q registers.
   (e) Two qubits rotated for the amplitude amplification.
   (f) One qubit for the spin.
   (g) One qubit for controlling the swap of the p and q registers.

   This reflection has cost $n_L + 2n_N + \aleph_1 + \aleph_2 + 2$.
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.

Adding all these complexities together gives

$$\begin{aligned}
&2(3n_L - 3\eta + 2b_r - 9) + \left\lceil \frac{L+1}{k_L} \right\rceil + b_L(k_L - 1) + \left\lceil \frac{L+1}{k'_L} \right\rceil + k'_L + 2b_L - 2 + 4(6n_N + 2b_r - 7) + 4(n_N^2 + n_N - 1) \\
&\quad + \left\lceil \frac{L+1}{k_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k_{p2}} \right\rceil + b_p(k_{p1}k_{p2} - 1) + \left\lceil \frac{L+1}{k'_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k'_{p2}} \right\rceil + k'_{p1}k'_{p2} + \left\lceil \frac{L}{k_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k_{p2}} \right\rceil \\
&\quad + b_p(k_{p1}k_{p2} - 1) + \left\lceil \frac{L}{k'_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k'_{p2}} \right\rceil + k'_{p1}k'_{p2} + 4\aleph_2 + 8n_N + 4n_N + 4(N-2) + 1 + 2n_N + \aleph_2 + 3 \\
&\quad + n_L + 2n_N + \aleph_1 + \aleph_2 + 4 \\
&= 7n_L + 4n_N^2 + 40n_N - 6\eta + 12b_r + \left\lceil \frac{L+1}{k_L} \right\rceil + b_L(k_L + 1) + \left\lceil \frac{L+1}{k'_L} \right\rceil + k'_L \\
&\quad + \left\lceil \frac{L+1}{k_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k_{p2}} \right\rceil + 2b_p k_{p1} k_{p2} + \left\lceil \frac{L+1}{k'_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k'_{p2}} \right\rceil + 2k'_{p1}k'_{p2} + \left\lceil \frac{L}{k_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k_{p2}} \right\rceil \\
&\quad + \left\lceil \frac{L}{k'_{p1}} \right\rceil \left\lceil \frac{N^2/8 + N/2}{k'_{p2}} \right\rceil + \aleph_1 + 4\aleph_2 + 4N - 56,
\end{aligned} \tag{B16}$$

with $b_L = n_L + \aleph_1 + 2$, $n_L = \lceil \log(L+1) \rceil$, $b_p = 2n_N + \aleph_2 + 2$, and $n_N = \lceil \log(N/2) \rceil$.

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 N qubits for the target system.
3. There are $n_L + 2$ qubits for the ℓ 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 ℓ register uses $n_L + 2\aleph_1 + 3$ qubits. Here n_L is for the alt values, $\aleph_1$ 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 $2n_N + 2$ qubits needed for the p and q registers, a qubit that is rotated for the equal superposition, and a qubit flagging success of preparing the equal superposition.
6. The contiguous register needs $\lceil \log(N^2/8 + N/4) \rceil$ qubits.
7. The equal superposition state used for the second preparation uses $\aleph_2$ qubits.
8. The phase-gradient register uses b_r qubits.
9. The QROM needs a number of qubits $b_p k_{p1} k_{p2} + \lceil \log[(L+1)/k_{p1}] \rceil + \lceil \log[(N^2/8 + N/2)/k_{p2}] \rceil$.

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

TABLE XI. Single low rank factorization data for the Reiher Hamiltonian [23]. Here λ_T is 90.4 hartree. The entry in blue is the one used for our resource estimates.

L	λ	CCSD(T) Error (mE_h)
50	3367.6	4.79
75	3657.9	0.21
100	3854.3	1.55
125	3997.4	3.08
150	4112.7	2.07
175	4199.2	1.63
200	4258.0	0.10
225	4300.7	0.38
250	4331.9	0.16
275	4354.9	0.29
300	4372.0	0.18
325	4385.3	0.13
350	4395.6	0.06
375	4403.4	0.03
400	4409.3	0.02

TABLE XII. Single low rank factorization data for the Li Hamiltonian [36]. Here λ_T is 561.5 hartree. The entry in blue is the one used for our resource estimates.

L	λ	CCSD(T) Error (mE_h)
50	2233.7	−93.07
75	2484.5	−57.08
100	2664.5	−23.77
125	2743.5	−9.37
150	2786.9	−11.55
175	2835.5	−8.17
200	2906.9	0.69
225	2986.9	1.52
250	3035.9	0.90
275	3071.8	0.48
300	3099.2	−0.07
325	3119.3	−0.18
350	3134.2	−0.11
375	3146.0	−0.08
400	3154.8	−0.10

$$2\lceil \log \mathcal{I} \rceil + N + 2n_L + 2\aleph_1 + \aleph_2 + b_r + 2n_N + 6 + \lceil \log(N^2/8 + N/4) \rceil + b_p k_{p1} k_{p2} + \lceil \log[(L+1)/k_{p1}] \rceil + \lceil \log[(N^2/8 + N/2)/k_{p2}] \rceil \quad \text{(B17)}$$

with $b_p = 2n_N + \aleph_2 + 2$, $n_N = \lceil \log(N/2) \rceil$, $n_L = \lceil \log(L+1) \rceil$. This completes the costing of the low-rank factorization method.

APPENDIX C: THE "DOUBLE LOW RANK FACTORIZATION" ALGORITHM OF VON BURG *ET AL.*

1. Representing the double low rank factorized Hamiltonian as a linear combination of unitaries

The approach of Ref. [10] is a modification of the approach of Ref. [9]. It is also based on applying qubitization to the representations discussed in Ref. [30], but the difference is that it uses a second factorization of the Coulomb operator. The idea to combine this second factorization with qubitization was suggested but not implemented in Ref. [9]. The idea is to further factorize the low-rank operator as

$$V \approx F = \frac{1}{2} \sum_{\ell=1}^{L} \left(\sum_{\sigma \in \{\uparrow,\downarrow\}} \sum_{p,q=1}^{N/2} W_{pq}^{(\ell)} a_{p,\sigma}^\dagger a_{q,\sigma} \right)^2 = \frac{1}{2} \sum_{\ell=1}^{L} U_\ell \left(\sum_{\sigma \in \{\uparrow,\downarrow\}} \sum_{p=1}^{N/2} f_p^{(\ell)} n_{p,\sigma} \right)^2 U_\ell^\dagger, \quad \text{(C1)}$$

where $n_{p,\sigma} = a_{p,\sigma}^\dagger a_{p,\sigma}$, $W_{pq}^{(\ell)}$ are scalars obtained by performing a Cholesky decomposition on a flattened version

of the V_{pqrs} tensor, ${f_p}^{(\ell)}$ are scalars obtained from the diagonalization of the squared one-body operator, and U_ℓ are the eigenvector matrices of that diagonalization. The sum over p can be truncated at $\Xi^{(\ell)}$ to a good approximation. Generally, we have that $L = \widetilde{\mathcal{O}}(N)$ and $\Xi^{(\ell)} < N$ but in some rather special cases $\Xi^{(\ell)}$ can be asymptotically less than this (e.g., for very large systems scaling towards the thermodynamic limit). For the moment we keep the sum at $N/2$ to show how part of this two-body operator can be combined with the one-body operator. We also assume for the moment that $L = N^2/4$ so there is no approximation involved in summing to L.

Using the Jordan-Wigner transformation of Eq. (A3) one can map the operator F to qubits as

$$F = \frac{1}{2}\sum_{\ell=1}^{L} U_\ell \left[\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{N/2} {f_p}^{(\ell)} \left(\frac{\mathbb{1} - Z_{p,\sigma}}{2}\right)\right]^2 U_\ell^\dagger$$
$$= \frac{1}{8}\sum_{\ell=1}^{L} U_\ell \left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{N/2} {f_p}^{(\ell)} \left(\mathbb{1} - Z_{p,\sigma}\right)\right)^2 U_\ell^\dagger. \quad \text{(C2)}$$

The factor of 1/2 becomes a factor of 1/8 because a factor of 1/2 comes from the Jordan-Wigner transformation, which is then squared. This operator may then be expressed as

$$F = \frac{1}{8}\sum_{\ell=1}^{L} U_\ell \left(\mathcal{I}_\ell - \mathcal{Z}_\ell\right)^2 U_\ell^\dagger$$
$$= \frac{1}{8}\sum_{\ell=1}^{L} U_\ell \left[2\,\mathcal{I}_\ell\left(\mathcal{I}_\ell - \mathcal{Z}_\ell\right) - \mathcal{I}_\ell^2 + \mathcal{Z}_\ell^2\right] U_\ell^\dagger$$
$$\equiv \frac{1}{8}\sum_{\ell=1}^{L}\left(2\,\mathcal{I}_\ell U_\ell \mathcal{N}_\ell U_\ell^\dagger - \mathcal{I}_\ell^2 + U_\ell \mathcal{Z}_\ell^2 U_\ell^\dagger\right), \quad \text{(C3)}$$

where we use the definitions

$$\mathcal{I}_\ell = 2\sum_{p=1}^{N/2} {f_p}^{(\ell)}\mathbb{1}, \quad \mathcal{Z}_\ell = \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{N/2} {f_p}^{(\ell)} Z_{p,\sigma},$$
$$\mathcal{N}_\ell = \mathcal{I}_\ell - \mathcal{Z}_\ell = \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{N/2} {f_p}^{(\ell)} n_{p,\sigma}. \quad \text{(C4)}$$

The authors of Ref. [10] add the one-body part of this expression into the one-body operator to provide a new one-body operator

$$\frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T_{pq} a_{p,\sigma}^\dagger a_{q,\sigma} + \frac{1}{4}\sum_{\ell=1}^{L}\mathcal{I}_\ell U_\ell \mathcal{N}_\ell U_\ell^\dagger$$
$$= \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2}\left(T_{pq} + \frac{1}{2}\sum_{\ell=1}^{L}\mathcal{I}_\ell W_{pq}^{(\ell)}\right) a_{p,\sigma}^\dagger a_{q,\sigma} \quad \text{(C5)}$$

where we use the relation that

$$U_\ell \mathcal{N}_\ell U_\ell^\dagger = U_\ell\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{N/2} {f_p}^{(\ell)} n_{p,\sigma}\right) U_\ell^\dagger$$
$$= \sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{pq=1}^{N/2} W_{pq}^{(\ell)} a_{p,\sigma}^\dagger a_{q,\sigma}. \quad \text{(C6)}$$

The expression for the one-body term can be simplified by noting that the trace is unchanged under U_ℓ, so

$$\mathcal{I}_\ell = 2\sum_{p=1}^{N/2} {f_p}^{(\ell)}\mathbb{1} = 2\sum_{r=1}^{N/2} W_{rr}^{(\ell)}\mathbb{1}. \quad \text{(C7)}$$

Thus, the additional term becomes

$$\frac{1}{2}\sum_{\ell=1}^{L}\mathcal{I}_\ell W_{pq}^{(\ell)} = \sum_{\ell=1}^{L}\sum_{r=1}^{N/2} W_{rr}^{(\ell)} W_{pq}^{(\ell)} = \sum_{r=1}^{N/2}\sum_{\ell=1}^{L} W_{rr}^{(\ell)} W_{pq}^{(\ell)}$$
$$= \sum_{r=1}^{N/2} V_{pqrr}. \quad \text{(C8)}$$

The last equality is exact when L is large enough that there is no truncation of the rank. Then the one-body term becomes, with no approximation due to truncation of the rank,

$$\frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T'_{pq} a_{p,\sigma}^\dagger a_{q,\sigma}, \quad \text{(C9)}$$

where T'_{pq} is as in Eq. (A9). Thus the one-body operator can be taken to be T', the same as for the sparse case in Appendix A and the single factorization in Appendix B.

For the two-body operator F, we can omit $-\mathcal{I}_\ell^2$, which gives only a uniform shift in the energy, so in F' only $U_\ell \mathcal{Z}_\ell^2 U_\ell^\dagger$ is retained. Moreover, in this two-body operator one can now truncate the sum over p at $\Xi^{(\ell)}$, so

$$F' = \frac{1}{8}\sum_{\ell=1}^{L} U_\ell\left(\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p=1}^{\Xi^{(\ell)}} {f_p}^{(\ell)} Z_{p,\sigma}\right)^2 U_\ell^\dagger. \quad \text{(C10)}$$

In this expression one can also take a truncation with $L < N^2/4$, so there is a truncation of the rank for the two-body operator but not the one-body operator.

The key insight of Ref. [10] is a method for implementing the U_ℓ such that one obtains a λ scaling as $\lambda = \lambda_{T'} + \lambda_{\rm DF}$ where

$$\lambda_{\rm DF} = \frac{1}{4}\sum_{\ell=1}^{L}\left(\sum_{p=1}^{\Xi^{(\ell)}}\left|f_p^{(\ell)}\right|\right)^2. \tag{C11}$$

The factor of 1/8 becomes a factor of 1/4 because we must multiply by $2^2 = 4$ due to the sum over the spin degree of freedom but we can then divide by 2 by using the oblivious amplitude amplification, Chebyshev polynomial technique. The contribution to λ from T' can be determined as follows. One can perform a rotation of the basis to diagonalize T' as well, so the value of $\lambda_{T'}$ is the Schatten 1-norm (sum of the absolute values of the eigenvalues, or trace norm) of the matrix $T'_{pq} = T_{pq} + \sum_{rr=1}^{N/2} V_{pqrr}$. The main reason that this double-factorization method is more efficient than the one in Ref. [9] is because it has reduced λ. While the authors of Ref. [10] write that their method is more efficient by more than an order of magnitude for various systems studied, this turns out not to be the case comparing to the most efficient method in Ref. [9] and correcting the estimate of λ.

2. Cost of qubitization of the double low rank factorized Hamiltonian

The primary cost of the double-factorization method from Ref. [10] comes from the circuit shown in Eq. (78) of their Supplemental Material. The costs there are given by the following parts.

1. A state preparation as shown in their Eq. (79). This needs to be performed and inverted twice.
2. A QROM to output the angles needed in their Lemma 8. This QROM needs to be called and erased twice.
3. The Z rotations as per their Eq. (68) need to be performed $4N$ times. Those have complexity $\beth - 2$ each, with
$$\beth = \lceil 5.652 + \log(\lambda N/\epsilon)\rceil \tag{C12}$$
the bits of precision of the rotation.
4. There are $N/2$ controlled swaps performed before and after the rotations as shown in Eq. (62) of Ref. [10]. Since that is repeated, the cost is $2N$.
5. The reflection in Eq. (78) of Ref. [10] has a complexity depending on how the reflection is performed.

First note that there seems to be a problem in using the circuit exactly as shown in Eq. (79) of [10]. It is using a joint state preparation on two registers, but the qubits before the preparation cannot be cleanly divided into those for the first register and those for the second register. That would be needed in order to reflect on only one, as is needed in the procedure in Eq. (79) of Ref. [10]. We first discuss the costing for the procedure as shown in Eq. (79), then describe how to perform the separate state preparations on the two registers.

In comparing the complexity as presented here to that in Ref. [10], it must be taken into account that they are defining N in a different way. In that work N is the number of orbitals, whereas here it is the number of *spin* orbitals, which is twice as large, because it is multiplied by the spin degree of freedom. Therefore, Ref. [10] takes $N = 54$ for the FeMoCo orbitals of Ref. [23], whereas here we take $N = 108$. Also, in the expression for $\beth$, ϵ is the error in synthesizing a single step, which is taken to be 0.0001 hartree in Ref. [10].

The dominant costs are from the QROM and the rotations. In using our THC representation we are able to reduce the QROM costs below those in Ref. [10], though the rotation costs are unchanged. To understand the QROM costs, we first list the main variables used.

(a) The main rank L.
(b) $\Xi^{(\ell)}$ is the rank of each term.
(c) $(1/L)\Xi = \sum_{\ell=1}^{L}\Xi^{(\ell)}$ is the average rank of the second factorization; thus, the total number of coefficients is $L\Xi$.

Before you perform the state preparation, you need to prepare an equal superposition over $L\Xi$ basis states. That has cost $3\lceil\log L\Xi\rceil - 3\eta + 2b_r - 9$ Toffolis, as per the analysis in Appendix A, where $L\Xi$ is divisible by 2^η. The QROM used for the state preparation has an output size

$$b = 2n_\Xi + 2n_L + \aleph + 2, \tag{C13}$$

where

$$n_\Xi = \lceil\log(\max_\ell \Xi^{(\ell)})\rceil, \quad n_L = \lceil\log(L+1)\rceil,$$
$$\aleph = \lceil 2.5 + \log(\lambda/\epsilon)\rceil, \tag{C14}$$

are the bits for the two registers and the bits of precision $\aleph$ used for the keep values. We take $n_L = \lceil\log(L+1)\rceil$ rather than $n_L = \lceil\log L\rceil$ in order to use an additional value of ℓ to account for the one-electron term. There is also a size for the contiguous register

$$n_{L,\Xi} = \lceil\log(L\Xi + N/2)\rceil. \tag{C15}$$

Here the $N/2$ will be to account for the one-electron term in the simulation. In the expression for b the factor of 2 on n_Ξ and n_L accounts for the ind and alt values in sparse state preparation, and the $+2$ is for the sign bit and its alternate value. (In [10] only one bit is assumed for the sign.) The complexity for the state preparation is that of the QROM,

plus $\aleph$ for the inequality test and $n_\Xi + n_L$ for the controlled swaps.

For the state preparation, since the number of items of data is $L\Xi$, and each has size b, the complexity of the QROM computation is

$$\left\lceil \frac{L\Xi}{k_p} \right\rceil + b(k_p - 1), \tag{C16}$$

where k_p must be chosen as a power of 2. The uncomputation cost is then

$$\left\lceil \frac{L\Xi}{k'_p} \right\rceil + k'_p. \tag{C17}$$

There is cost $n_\Xi + n_L$ for the controlled swaps and $\aleph$ for the inequality test, which are inverted as well for the inverse preparation, giving cost $2n_\Xi + 2n_L + 2\aleph = b + \aleph - 2$. Taking account of the need to perform the state preparation and inverse state preparation twice, we give the cost of the state preparation for the two-body term as

$$2\left\lceil \frac{L\Xi}{k_p} \right\rceil + 2bk_p + 2\left\lceil \frac{L\Xi}{k'_p} \right\rceil + 2k'_p + 2\aleph - 4. \tag{C18}$$

Before performing the QROM for the rotations, it is necessary to use QROM to generate a register with the offsets in order to produce a contiguous register. This procedure is described in Eq. (39) and Lemma 7 of Ref. [10]. The complexity of the QROM for the offsets is

$$\left\lceil \frac{L}{k_L} \right\rceil + n_{L,\Xi}(k_L - 1). \tag{C19}$$

In uncomputing the complexity is

$$\left\lceil \frac{L}{k'_L} \right\rceil + k'_L. \tag{C20}$$

We also need to perform addition of registers twice and invert addition of registers twice, which has cost $4(n_{L,\Xi} - 1)$. That gives a total cost of

$$2\left\lceil \frac{L}{k_L} \right\rceil + 2n_{L,\Xi}(k_L + 1) + 2\left\lceil \frac{L}{k'_L} \right\rceil + 2k'_L - 4. \tag{C21}$$

Once the contiguous register is produced, one can apply the QROM to output the rotation angles. Because there are $L\Xi$ sets of $N\beth/2$ bits for the rotation angles, the cost of the QROM used for the rotation angles can be given as

$$\left\lceil \frac{L\Xi}{k_r} \right\rceil + N\beth(k_r - 1)/2, \tag{C22}$$

and the cost of the inverse QROM as

$$\left\lceil \frac{L\Xi}{k'_r} \right\rceil + k'_r. \tag{C23}$$

In Ref. [10], $\beth$ is taken to be

$$\beth = \lceil 5.652 + \log(N\lambda/2\epsilon) \rceil. \tag{C24}$$

Note that we are using our convention for N, which is double that used in Ref. [10]. Accounting for two steps with preparation and inverse preparation the cost is

$$2\left\lceil \frac{L\Xi}{k_r} \right\rceil + N\beth(k_r - 1) + 2\left\lceil \frac{L\Xi}{k'_r} \right\rceil + 2k'_r. \tag{C25}$$

For the total cost of the two-body term in the double-factorization method of Ref. [10], we add the QROM costs in Eqs. (C18), (C21), and (C25), plus the $4N(\beth - 2)$ cost for the controlled rotations, plus the $2N$ cost for the controlled swaps (for the spin), plus twice the $3n_{L,\Xi} - 3\eta + 2b_r - 9$ cost for preparing the equal superposition state, $n_{L,\Xi} - 1$ for reflection and 2 for the unary iteration for the phase estimation and making the reflection controlled. There is also a cost of $n_\Xi - 1$ for the reflection on the second register, if this were possible as claimed in Ref. [10]. The total cost is, therefore,

$$\begin{aligned} &2\left\lceil \frac{L\Xi}{k_p} \right\rceil + 2bk_p + 2\left\lceil \frac{L\Xi}{k'_p} \right\rceil + 2k'_p + 2\aleph + 2\left\lceil \frac{L}{k_L} \right\rceil \\ &+ 2n_{L,\Xi}(k_L + 1) + 2\left\lceil \frac{L}{k'_L} \right\rceil + 2k'_L + 2\left\lceil \frac{L\Xi}{k_r} \right\rceil \\ &+ N\beth(k_r - 1) + 2\left\lceil \frac{L\Xi}{k'_r} \right\rceil + 2k'_r + 4N\beth - 6N \\ &+ 7n_{L,\Xi} - 6\eta + 4b_r + n_\Xi - 26, \end{aligned} \tag{C26}$$

where $b = 2n_\Xi + 2n_L + \aleph + 2$ with the numbers of qubits as given in Eq. (C14). The values of k should be chosen as powers of 2 that minimize the cost.

As mentioned above, the procedure needs to be made more complicated, because the state preparation should be performed separately on the two registers, rather than jointly on both registers at once, because a reflection needs to be made on one register. An alternative approach is to prepare the state on the first register, then prepare the state on the second register controlled on the first register, rather than preparing the state on both registers together.

It is also necessary to account for the one-electron term. It is possible to add that in an explicit way as proposed in Ref. [10], but that significantly increases the complexity because of the need to perform high-accuracy rotations again. Instead, one can combine it with the two-electron term in a similar way as we have described above for the single low rank factorization. The principle is to use an additional basis state on the first register to flag the one-electron term. Then the second register will have a state preparation corresponding to the one-electron term for this basis state on the first register, but *only* for the first part, not the second, because we are not applying the oblivious amplitude amplification within a single step for the one-electron term.

The detailed procedure is shown in Fig. 16. The steps are as follows.

1. First we need to prepare the appropriate superposition state on the ℓ register, which is indicated as the box labeled "prep" in Fig. 16. The steps to perform that are as follows.

   (a) Prepare an equal superposition over $L+1$ basis states, with cost $3n_L - 3\eta + 2b_r - 9$ Toffolis, as per the analysis in Appendix A, where L is divisible by 2^η.

   (b) Use a QROM for state preparation on the first register outputting an alt value of size n_L and a keep value of size $\aleph_1$. The output size is

   $$b_{p1} = n_L + \aleph_1, \tag{C27}$$

   so the QROM cost is

   $$\left\lceil \frac{L+1}{k_{p1}} \right\rceil + b_{p1}(k_{p1} - 1). \tag{C28}$$

   (c) Perform an inequality test with cost $\aleph_1$.

   (d) Perform the controlled swap with cost n_L, completing the state preparation on the first register.

2. Before we prepare the state on the p register, we need to output data from the ℓ register. This is indicated by the "In$_\ell$" and "data$_\ell$" in Fig. 16. For each value on the first register we need to output $\Xi^{(\ell)}$ (with n_Ξ bits), and b_r bits for a rotation on an ancilla qubit. It is also convenient to output the offset (with $n_{L,\Xi}$ bits) at this stage, and determine if $\ell = 0$, giving output size

   $$b_o = n_\Xi + n_{L,\Xi} + b_r + 1, \tag{C29}$$

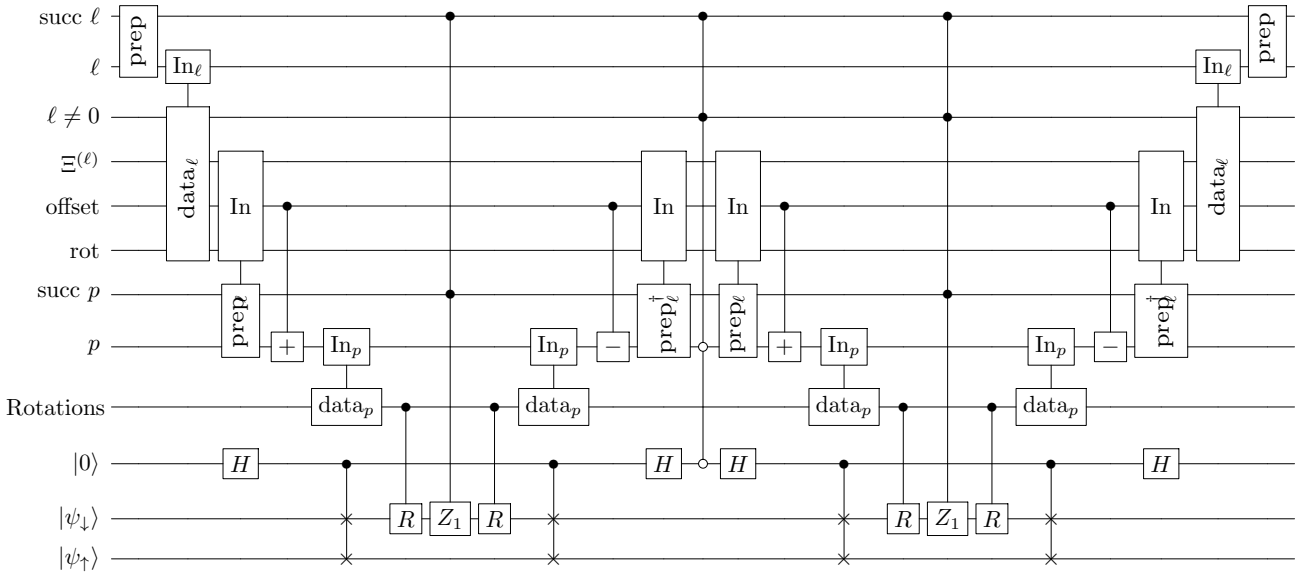

FIG. 16. The circuit for performing the state preparation and controlled operations (SELECT) for the double factorization approach. The register labeled ℓ is the first control register containing ℓ, and p labels the second control register. The registers labeled succ ℓ and succ p are the registers that flag success of the preparation of the equal superposition state for the ℓ and p registers. The register labeled $\ell \neq 0$ is a temporary register used to keep the result of an inequality test checking that $\ell \neq 0$, which is used to ensure that the second half of the circuit has no effect for that value of ℓ, which is used to label the one-electron term. The registers labeled $\Xi^{(\ell)}$, "offset," and "rot" are the outputs of the QROM on the ℓ register that are mainly used for the controlled preparation of the state on the p register. The register labeled "rotations" is the data needed for the basis rotations for implementing $Z_{p,\sigma}$. The register labeled $|0\rangle$ is used to select the spin, and $|\psi_\downarrow\rangle$ and $|\psi_\uparrow\rangle$ are the spin-down and -up components of the target system.

and QROM cost

$$\left\lceil \frac{L+1}{k_o} \right\rceil + b_o(k_o - 1). \tag{C30}$$

3. Next we need to prepare the state on the p register controlled on the ℓ register. This step is indicated by the "In" and "prep$_\ell$" in Fig. 16, and uses the data output by the QROM in the previous step. The steps needed for the preparation are as follows.

   (a) The controlled preparation of an equal superposition state may be performed in the following way with complexity $7n_\Xi + 2b_r - 6$. The steps are as follows.
   i. Use a string of $n_\Xi - 1$ Toffolis (and Clifford gates) to produce a new n_Ξ-qubit register that has zeros matching the leading zeros in the binary representation of $\Xi^{(\ell)}$, and ones after that. These qubits are used to control the Hadamards in the next step.
   ii. Perform n_Ξ controlled Hadamards with cost n_Ξ. A controlled Hadamard can be performed with a single Toffoli by catalytically using a T state, as shown in Fig. 17. Therefore, we count the cost of each controlled Hadamard as 1.
   iii. Perform an inequality test with the register containing $\Xi^{(\ell)}$ with cost n_Ξ.
   iv. Rotate the ancilla qubit with cost $b_r - 2$ based on the rotation angle given by the output of the QROM.
   v. The reflection on the result of the inequality test and ancilla qubit is a controlled phase, which is a Clifford gate.
   vi. Invert the rotation with cost $b_r - 2$ and the inequality test with Cliffords.
   vii. Perform the n_Ξ controlled Hadamards again.
   viii. Reflect about the zero state on $n_\Xi + 1$ qubits (the qubits where the state preparation is being performed and the rotated ancilla) with cost $n_\Xi - 1$.
   ix. Perform the n_Ξ controlled Hadamards again. Now the binary-to-unary conversion can be inverted with Cliffords.

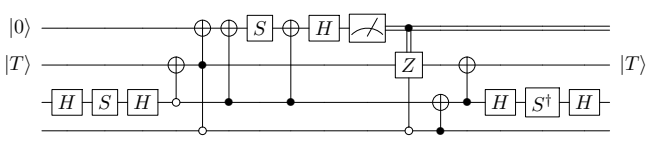

FIG. 17. A quantum circuit for catalytically implementing a controlled Hadamard using a single Toffoli gate, Clifford gates, and a catalytic state $|T\rangle = T|+\rangle$. The bottom qubit is the control, and the third qubit (second from the bottom) is the target.

   x. Perform an inequality test again with cost n_Ξ. Now, flagged on the success of the inequality test, we prepared an equal superposition state on the second register that can be used for state preparation on the second register.
   (b) We now need to add the offset to the second register to provide a contiguous register to apply QROM to for state preparation. This addition has cost $n_{L,\Xi} - 1$.
   (c) Next, apply the QROM to output the alt and keep values with size

$$b_{p2} = n_\Xi + \aleph_2 + 2, \tag{C31}$$

   where the $+2$ is for the sign bit and its alt value. In this part there are $L\Xi + N/2$ outputs to give, because the rotations for the one-electron term are needed as well, then in the next part there are only $L\Xi$. Therefore, the two QROMs have cost

$$\left\lceil \frac{L\Xi + N/2}{k_{p2}} \right\rceil + b_{p2}(k_{p2} - 1). \tag{C32}$$

   (d) The remainder of the state preparation uses an inequality test with cost $\aleph_2$, and a controlled swap with cost n_Ξ.

4. Apply the number operators via rotations and QROM with the following steps.

   (a) To apply the QROM for the rotation angles, we need to again add an offset to the second register to provide a contiguous register, with cost again $n_{L,\Xi} - 1$. This operation is indicated by the controlled box with $+$ in it in Fig. 16, and a box with $-$ in it for its inversion.
   (b) Use QROM to output the rotation angles, which is shown as "In$_p$" and "data$_p$" in Fig. 16. The QROM cost is

$$\left\lceil \frac{L\Xi + N/2}{k_r} \right\rceil + N\beth(k_r - 1)/2. \tag{C33}$$

   (c) Perform controlled swaps with the spin qubit as control, with cost $N/2$.
   (d) Apply the controlled rotations to rotate the basis with cost $N(\beth - 2)$.
   (e) Apply the Z_1 operation, controlled on success of the preparation of the ℓ and p registers. This control gives a cost of 1.
   (f) Reverse the controlled rotations and controlled swaps, with cost $N(\beth - 2)$ and $N/2$.
   (g) Reverse the QROM with cost

$$\left\lceil \frac{L\Xi + N/2}{k'_r} \right\rceil + k'_r. \tag{C34}$$

(h) Reverse the addition onto the contiguous register, with cost again $n_{L,\Xi} - 1$.

5. Invert the state preparation on the p register, which is shown as "In" and "prep$_p^\dagger$," in Fig. 16. The cost is the same as given in step 2, except the cost of the QROM is reduced to

$$\left\lceil \frac{L\Xi + N/2}{k'_{p2}} \right\rceil + k'_{p2}. \tag{C35}$$

6. The reflection on the second register *only* (which is what caused the procedure to be this complicated) is performed. The qubits that need to be reflected on are as follows.

   (a) The n_Ξ qubits that the state is prepared on.
   (b) The $\aleph_2$ that are used for the superposition state for the coherent alias sampling.
   (c) One that is rotated for the amplitude amplification.
   (d) One for the spin.

   That gives a total of $n_\Xi + \aleph_2 + 2$. This reflection also needs to be controlled on success for preparation of the ℓ register, and $\ell \neq 0$, so the complexity of the reflection is $n_\Xi + \aleph_2 + 2$.
7. Repeat steps 2 to 5. This time, because we do not need to output values for the one-electron terms, the costs of the QROM and inverse QROM for state preparation are reduced to

$$\left\lceil \frac{L\Xi}{k_{p2}} \right\rceil + b_{p2}(k_{p2} - 1), \qquad \left\lceil \frac{L\Xi}{k'_{p2}} \right\rceil + k'_{p2}. \tag{C36}$$

The costs for the QROM for the rotations are reduced to

$$\left\lceil \frac{L\Xi}{k_r} \right\rceil + N\beth(k_r - 1)/2, \qquad \left\lceil \frac{L\Xi}{k'_r} \right\rceil + k'_r. \tag{C37}$$

This time Z_1 must be controlled by the qubit flagging $\ell \neq 0$, which introduces a cost of another Toffoli.
8. Invert the QROM in step 2 and the preparation in step 1, where the costs of the QROMs are reduced to

$$\left\lceil \frac{L+1}{k'_{p1}} \right\rceil + k'_{p1}, \qquad \left\lceil \frac{L+1}{k'_o} \right\rceil + k'_o. \tag{C38}$$

9. There is also a cost for the reflection needed for constructing the quantum walk. The qubits that need to be reflected on are as follows.

   (a) The n_L qubits for the ℓ register.
   (b) The n_Ξ qubits for the p register.
   (c) The $\aleph_1$ qubits for the equal superposition state for the alias sampling for the ℓ register.
   (d) The $\aleph_2$ qubits for the equal superposition state for the p register.
   (e) The two qubits that are rotated.
   (f) One qubit for the spin.

   That gives a total of $n_L + n_\Xi + \aleph_1 + \aleph_2 + 3$ so the cost is $n_L + n_\Xi + \aleph_1 + \aleph_2 + 1$.
10. The steps of the quantum walk are made controlled using unary iteration on an ancilla. This introduces a cost of another two Toffolis for the unary iteration and making the reflection controlled.

Adding together all these costs, then gives

$$\begin{aligned}
&2(3n_L - 3\eta + 2b_r - 9) + \left\lceil \frac{L+1}{k_{p1}} \right\rceil + b_{p1}(k_{p1} - 1) + 2\aleph_1 + 2n_L + \left\lceil \frac{L+1}{k_o} \right\rceil + b_o(k_o - 1) + \left\lceil \frac{L+1}{k'_{p1}} \right\rceil + k'_{p1} \\
&+ \left\lceil \frac{L+1}{k'_o} \right\rceil + k'_o + \left\lceil \frac{L\Xi + N/2}{k_r} \right\rceil + \left\lceil \frac{L\Xi}{k_r} \right\rceil + N\beth(k_r - 1) + \left\lceil \frac{L\Xi + N/2}{k'_r} \right\rceil + \left\lceil \frac{L\Xi}{k'_r} \right\rceil + 2k'_r \\
&+ 4(7n_\Xi + 2b_r - 6) + 8(n_{L,\Xi} - 1) + \left\lceil \frac{L\Xi + N/2}{k_{p2}} \right\rceil + \left\lceil \frac{L\Xi}{k_{p2}} \right\rceil + 2b_{p2}(k_{p2} - 1) + \left\lceil \frac{L\Xi + N/2}{k'_{p2}} \right\rceil \\
&+ \left\lceil \frac{L\Xi}{k'_{p2}} \right\rceil + 2k'_{p2} + 4\aleph_2 + 4n_\Xi + 4N(\beth - 2) + 2N + 2 + n_\Xi + \aleph_2 + 2 + n_L + n_\Xi + \aleph_1 + \aleph_2 + 3
\end{aligned}$$

$$= 9n_L - 6\eta + 12b_r + \left\lceil \frac{L+1}{k_{p1}} \right\rceil + b_{p1}(k_{p1}-1) + \left\lceil \frac{L+1}{k_o} \right\rceil + b_o(k_o-1) + \left\lceil \frac{L+1}{k'_{p1}} \right\rceil + k'_{p1} + \left\lceil \frac{L+1}{k'_o} \right\rceil + k'_o$$
$$+ \left\lceil \frac{L\Xi + N/2}{k_r} \right\rceil + \left\lceil \frac{L\Xi}{k_r} \right\rceil + N\beth k_r + \left\lceil \frac{L\Xi + N/2}{k'_r} \right\rceil + \left\lceil \frac{L\Xi}{k'_r} \right\rceil + 2k'_r + 34n_\Xi + 8n_{L,\Xi}$$
$$+ \left\lceil \frac{L\Xi + N/2}{k_{p2}} \right\rceil + \left\lceil \frac{L\Xi}{k_{p2}} \right\rceil + 2b_{p2}(k_{p2}-1) + \left\lceil \frac{L\Xi + N/2}{k'_{p2}} \right\rceil + \left\lceil \frac{L\Xi}{k'_{p2}} \right\rceil + 2k'_{p2} + 3\aleph_1 + 6\aleph_2 + 3N\beth - 6N - 43, \tag{C39}$$

with $b_{p1} = n_L + \aleph_1$, $b_o = n_\Xi + n_{L,\Xi} + p$, and $b_{p2} = n_\Xi + \aleph_2 + 2$, and the numbers of qubits as defined in Eq. (C14).

Next consider the logical qubit count. The qubits used are as follows.

1. The $\lceil \log(\mathcal{I}+1) \rceil$ qubits for the control register for the state preparation and $\lceil \log(\mathcal{I}+1) \rceil - 1$ for the unary iteration on this register.
2. There are N qubits used for the system.
3. The first register that is prepared needs n_L qubits, plus one to flag success of the state preparation and one that is rotated as part of the state preparation.
4. The output of the QROM needs $n_L + \aleph_1$, another $\aleph_1$ are used for the equal superposition state to take the inequality with, and another qubit is needed as the output of the inequality test. We can ignore temporary ancillas used as more will be needed at a later step.
5. The second QROM on the first register used to output the data needed for the equal superposition state on the second register uses b_o output bits.
6. The second register needs n_Ξ bits, plus another bit for flagging success of preparing the equal superposition and another that is rotated.
7. A register with size $n_{L,\Xi}$ for outputting the contiguous register for the state preparation on the second register. This is a temporary ancilla that can be erased after the QROM with Cliffords by using the out-of-place adder. It then is computed again for the QROM for the rotation angles. There it can be computed in place, so does not add to the qubit count.
8. The QROM used for preparing the state on the second register uses b_{p2} qubits output, as well as a number of temporary qubits that are less than those needed for the rotation angles output later.
9. The state preparation needs another $\aleph_2$ qubits in a superposition, as well as another qubit flagging success of the inequality test.
10. The angles for the rotations need $k_r N\beth/2$.
11. The phase-gradient state, which needs $\beth$ qubits.
12. Another control qubit is needed for the spin.
13. A T state on a single qubit is used to perform the controlled Hadamards with Toffoli gates.

Adding together these ancilla costs gives

$$N + 2n_L + n_\Xi + 2\aleph_1 + \aleph_2 + \beth + b_o + b_{p2} + k_r N\beth/2 + 2\lceil \log(\mathcal{I}+1) \rceil + 7. \tag{C40}$$

3. Numerical determination of double low rank factorization

The numerical determination of double low rank factorization can be ambiguous without further details. In this subsection, we aim to provide full details of how the factorization is obtained in this work. The original approach proposed by Peng and Kowalski [35] worked with separate thresholds for the first and the second factorizations. Instead, von Burg *et al.* [10] proposed a truncation scheme just based on a single threshold, which we further elaborate here.

1. First, we perform either the eigendecomposition or the Cholesky decomposition of V, $V_{pqrs} = \sum_{\ell=1}^{L} W_{pq}^{(\ell)} W_{rs}^{(\ell)}$ and sort ℓ in the ascending order of the corresponding eigenvalues such that the magnitude of $W_{pq}^{(\ell)}$ for small ℓ's are larger than that of large ℓ's .
2. The second factorization was originally proposed to be done with the singular value decomposition of $W^{(\ell)}$ [35], but following von Burg *et al.* [10] we use eigendecomposition: $W_{pq}^{(\ell)} = \sum_{m=1}^{N/2} f_m^{(\ell)} U_{pm}^{(\ell)} U_{qm}^{(\ell)}$.
3. For a given threshold and looping over the first factorization index ℓ from 1 to L, we discard the mth eigenvector in the second factorization that satisfies

$$\left(\sum_{p=1}^{N/2} |f_p^{(\ell)}| \right) |f_m^{(\ell)}| < \text{threshold}\,. \tag{C41}$$

We note that if no eigenvectors are kept at ℓ_0 then we discard the rest of the vectors for $\ell > \ell_0$ without going through them.

TABLE XIII. Double low rank factorization data for the Reiher Hamiltonian [23]. Here $\lambda_{T'}$ is 38.6 hartree, which is the Schatten-norm of T'. A threshold is used to truncate eigenvectors based on the truncation strategy described in Ref. [10] Eq. (C41). The entry in blue is the one used for our resource estimates.

Threshold	L	Eigenvectors	λ	CCSD(T) Error (mE_h)
0.1	53	765	253.6	−87.91
0.05	95	1274	262.1	−12.39
0.025	135	2319	272.0	−2.78
0.0125	182	3983	280.8	−1.10
0.01	195	4700	283.4	−0.18
0.0075	216	5678	286.3	1.33
0.005	242	7181	289.5	2.27
0.0025	300	9930	292.9	1.02
0.00125	360	13031	294.8	0.44
0.001	384	14062	295.2	0.25
0.00075	414	15419	295.6	0.20
0.0005	444	17346	296.1	0.09
0.000125	567	24005	296.7	−0.01
0.0001	581	25054	296.7	−0.02
0.00005	645	28469	296.8	0.00

This scheme is used to generate the numerical data presented in Tables XIII and XIV.

4. Numerical data for hydrogen chains and H_4

The average rank of the second factorization for hydrogen chain and H_4 is shown in Fig. 18. Theoretically, Ξ cannot scale worse than $O(N)$. However, we obtain $O(N^{1.29})$ in Fig. 18(b). Therefore, in Table II we assume that Ξ scales as $O(N)$ for the H_4 case.

TABLE XIV. Double low rank factorization data for the Li Hamiltonian [36]. Here $\lambda_{T'}$ is 478.1 hartree, which is the Schatten norm of T'. A threshold is used to truncate eigenvectors based on the truncation strategy described in Ref. [10] and Eq. (C41). The entry in blue is the one used for our resource estimates.

threshold	L	Eigenvectors	λ	CCSD(T) error (mE_h)
0.1	94	1998	1076.5	−287.44
0.05	184	3765	1108.9	−91.76
0.025	205	5992	1136.0	−20.52
0.0125	247	8450	1152.1	−12.74
0.01	261	9302	1155.5	−6.73
0.0075	278	10493	1159.1	−5.77
0.005	312	12508	1163.4	−2.30
0.0025	344	16355	1168.6	1.53
0.00125	394	20115	1171.2	0.07
0.001	413	21407	1171.7	0.42
0.00075	434	23145	1172.2	0.40
0.0005	470	25751	1172.8	0.42
0.000125	589	35006	1173.7	−0.04
0.0001	614	36557	1173.7	−0.02
0.00005	679	41563	1173.9	−0.01

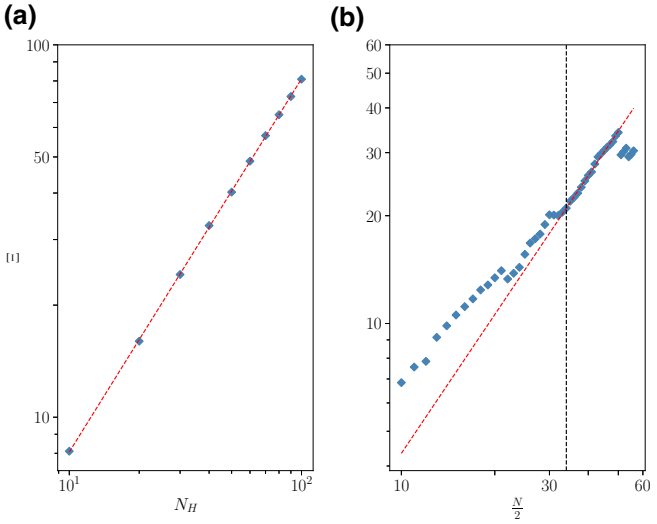

FIG. 18. (a) The number of H atoms (N_H) versus Ξ and (b) the number of orbitals ($N/2$) versus Ξ. All data points in (a) are used in the linear fit on a log scale where the slope is given as 1.00 with $R^2 = 0.9999$ (where R is the correlation coefficient). In (b), every point beyond $N/2 = 35$ except the last six points is used for the linear fit, which yields a slope of 1.29 with $R^2 = 0.9917$. The DF threshold that is used is 0.01 for (a) and 1×10^{-4} for (b).

APPENDIX D: THE RANDOMIZED TROTTER-SIMULATION ALGORITHMS OF CAMPBELL AND KIVLICHAN *ET AL.*

Here we discuss in more detail a recent randomized approach, often called qDRIFT [52], that approximates time evolution using a randomized evolution. The randomized nature of qDRIFT leads to a subtle point arising between the cases considered here and the analysis provided in Refs. [52,53] in that the results in both cases discuss the probability of failure for the algorithm rather than the mean-square error. The analysis below focuses on the mean-square error in phase estimation that arises from qDRIFT and we see that because of the possibility of large deviations, the worst case scaling of the complexity can be substantially worse than the $\mathcal{O}(\lambda^2/\epsilon^2)$ scaling anticipated from such results.

Before jumping into the analysis, we review the fundamentals of the qDRIFT algorithm. The idea behind the evolution is to implement a channel, for the Hamiltonian $H = \sum_j H_j$

$$U(\tau,0) : \rho \mapsto \sum_j p_j e^{-iH_j\tau/p_j} \rho e^{iH_j\tau/p_j}, \qquad \text{(D1)}$$

where $p_j = \|H_j\|/\sum_j \|H_j\|$ where $\|\cdot\|$ is the Schatten infinity norm (spectral norm). In our case we assume that each H_j is proportional to a unitary so then $p_j = \|H_j\|/\lambda$. This channel can be implemented by randomly selecting a term from the Hamiltonian, simulating the evolution of just

that term for a duration that scales inversely with the probability of drawing the Hamiltonian term. From Ref. [52] we have that if we define $\mathcal{E}(\tau,0):\rho\mapsto e^{-iH\tau}\rho e^{iH\tau}$

$$\|U(\tau,0)-\mathcal{E}(\tau,0)\|_\diamond \le 2\lambda^2\tau^2. \tag{D2}$$

Note that this exact same bound holds if we consider a controlled evolution using qDRIFT, which can be thought of as simulating the Hamiltonian $H_{\text{ctrl}}=|1\rangle\langle 1|\otimes H=\sum_j |1\rangle\langle 1|\otimes H_j$. As discussed in Ref. [52], the value of λ is unchanged so the bound on the error in the controlled Hamiltonian is the same as for the Hamiltonian without control. For improved performance of the phase estimation, one can also use control between positive and negative Hamiltonian evolution [16]. That corresponds to $H_{\text{ctrl}}=-Z\otimes H$, and again the analysis of the error for the controlled Hamiltonian is the same as without control, because the value of λ is unchanged. This implies that the same bound holds in the controlled and the uncontrolled case. When used in conjunction with qDRIFT, the variance in the estimate is not the only source of error.

1. qDRIFT mean-square error

First, let us examine the scaling if we were to try to apply phase estimation using qDRIFT in order to achieve a fixed root-mean-square error in the energy (for example 1.6 mHa). Existing work in Ref. [52] discusses this case; however, the failure probability scaling of $O[1/(\delta^3\epsilon^2)]$, where δ is the failure probability and ϵ is the uncertainty in the phase upon success, in that work seemingly precludes that analysis from yielding an inexpensive simulation that guarantees a fixed root-mean-square error. Here we provide a tighter analysis and demonstrate that the scaling is worse than one would naïvely expect.

We begin by using the fact that the diamond norm bounds the worst-case trace norm between outputs that can be attained over all inputs to the channels. Further, it can be seen using the arguments from Ref. [52,105] that for any positive integer R

$$\|U^R(\tau,0)-\mathcal{E}^R(\tau,0)\|_\diamond \le 2R\lambda^2\tau^2. \tag{D3}$$

Thus by the von Neumann trace inequality it follows that for any bounded observable O:

$$\begin{aligned}\max_\rho &|\text{Tr}[OU^R(\tau,0)(\rho)]-\text{Tr}[O\mathcal{E}^R(\tau,0)(\rho)]|\\ &\le 2\|O\|R\lambda^2\tau^2.\end{aligned} \tag{D4}$$

Let ϕ be the exact phase for the exact evolution, $\hat{\phi}$ be the observable for the phase, and take $O=(\hat{\phi}-\phi)^2$ to be the observable for the squared error in the phase. The quantity we wish to bound is then for initial state ρ_0

$$\begin{aligned}\langle(\hat{\phi}-\phi)^2\rangle &:= \text{Tr}[(\hat{\phi}-\phi)^2\mathcal{E}^R(\tau,0)(\rho_0)]\\ &\le \left|\text{Tr}[(\hat{\phi}-\phi)^2U^R(\tau,0)(\rho_0)]\right|\\ &\quad+\left|\text{Tr}[OU^R(\tau,0)(\rho)]-\text{Tr}[O\mathcal{E}^R(\tau,0)(\rho)]\right|,\end{aligned} \tag{D5}$$

where the expectation value on the left is defined to be that for the qDRIFT evolution.

Then $\|O\|\le\max|\pi\pm\phi|^2$, and so

$$|\text{Tr}[OU^R(\tau,0)]-\text{Tr}[O\mathcal{E}^R(\tau,0)]|\le 2\max|\pi\pm\phi|^2R\lambda^2\tau^2. \tag{D6}$$

That gives the maximum increase in the mean-square error in the phase. Now consider the case that there is a different time interval for the qDRIFT τ versus the time interval for the phase estimation t, with $\tau=t/k$ for some integer k. Then the total number of intervals for the qDRIFT R is related to the total number for phase estimation r as $R=kr$. Then the mean-square error in the phase for the exact evolution is π^2/r^2, and so

$$\langle(\hat{\phi}-\phi)^2\rangle \le \frac{\pi^2}{r^2}+\frac{2\max|\pi\pm\phi|^2r\lambda^2t^2}{k}. \tag{D7}$$

This expression corresponds to the uncertainty in the phase that we would achieve through the use of traditional phase estimation; however, we can improve the phase achieved per unit time by controlling the direction of the phase evolution as discussed in Refs. [45,105]. Using this observation we can define an energy operator $\hat{E}=\hat{\phi}/2t$ so that

$$\langle(\hat{E}-E)^2\rangle \le \frac{\pi^2}{4r^2t^2}+\frac{\max|\pi\pm\phi|^2r\lambda^2}{2k}. \tag{D8}$$

Then writing this in terms of $N_{\text{exp}}=kr$,

$$\langle(\hat{E}-E)^2\rangle \le \frac{\pi^2k^2}{4N_{\text{exp}}^2t^2}+\frac{\max|\pi\pm\phi|^2N_{\text{exp}}\lambda^2}{2k^2}. \tag{D9}$$

We aim to find the smallest N_{exp} such that the right-hand side is bounded above by ϵ^2, where ϵ is as before the bound on the allowable root-mean square error in the energy estimate. Minimizing N_{exp} for constant ϵ is the same as minimizing ϵ for constant N_{exp}. To minimize over k, take

the derivative of the right-hand side to give

$$\frac{\partial}{\partial k}\left(\frac{\pi^2 k^2}{4N_{\rm exp}^2 t^2}+\frac{\max|\pi\pm\phi|^2 N_{\rm exp}\lambda^2}{2k^2}\right) = \frac{\pi^2 k}{2N_{\rm exp}^2 t^2}-\frac{\max|\pi\pm\phi|^2 N_{\rm exp}\lambda^2}{k^3}. \tag{D10}$$

The turning point is for

$$k^2=\sqrt{2N_{\rm exp}^3}\max|1\pm\phi/\pi|\lambda t. \tag{D11}$$

Putting that into Eq. (D9) gives

$$\langle(\hat{E}-E)^2\rangle\leq\frac{\pi\max|\pi\pm\phi|\lambda}{\sqrt{2N_{\rm exp}}t}. \tag{D12}$$

Setting the right-hand side equal to ϵ^2 then gives

$$N_{\rm exp}=\frac{\pi^2\max|\pi\pm\phi|^2\lambda^2}{2\epsilon^4 t^2}. \tag{D13}$$

Because $E=\phi/2t$ and E can, in principle, be as large as λ, $\phi=\pi$ should correspond to $E\geq\lambda$ in order to resolve all possible values of E. That implies that the maximum allowable value of t is $\pi/2\lambda$, which gives

$$N_{\rm exp}=\frac{4\max|\pi\pm\phi|^2\lambda^4}{2\epsilon^4}\leq\frac{8\pi^2\lambda^4}{\epsilon^4}. \tag{D14}$$

This scaling is as $\mathcal{O}(\lambda^4/\epsilon^4)$ and will therefore yield large gate counts for the problems that we focus on. The reason for the large scaling, is that the diamond norm bounds the difference in the probabilities of measurement results, which can allow an increase in probability concentrated at the maximum error leading to large mean-square error.

The gate complexity for the unamplified qDRIFT subroutine can be found by multiplying the number of exponentials by the number of Toffoli gates needed per exponential. The Hamiltonian can be expressed such that each H_j is a weighted Pauli operator. In this case $e^{-iH_j\tau/p_j}$ can be implemented using at most $\mathcal{O}(N)$ one and two qubit Clifford operations and a single R_z gate [11]. We implement these rotations using a state that is constructed using a $q+1$ qubit phase-gradient state:

$$|\psi_{q+1}\rangle=\frac{1}{\sqrt{2^{q+1}}}\sum_{j=0}^{2^{q+1}-1}e^{\pi ij/2^q}|j\rangle. \tag{D15}$$

We can then use this as a resource state to implement a single qubit rotation using incrementer gates as described in Appendix A of Ref. [84], which requires $q-1$ Toffoli gates to implement $e^{-iZ\pi/2^q}$ using the $q+1$ qubit resource state in Eq. (D15).

Next note that for any value of $\lambda t/k$ we can round the desired rotation angle to $\lambda t/k\geq\pi/2^{\lceil\log(\pi k/\lambda t)\rceil}$. Thus we can choose the evolution time t such that we, at worst, need to halve the evolution time to ensure that the rotation angle coincides with the angle returned by the phase-gradient state. This will necessitate choosing $t\in[\pi/4\lambda,\pi/2\lambda]$, which corresponds in the best case scenario to taking $q=\log(2k)\approx(1/2)\log(32\pi^4\lambda^6/\epsilon^6)+1$. The number of Toffoli gates needed is

$$N_{\rm Toff}=N_{\rm exp}(q-1)\approx\frac{4\pi^2\lambda^4\log(32\pi^4\lambda^6/\epsilon^6)}{\epsilon^4}. \tag{D16}$$

The total number of ancillae needed by the algorithm is given by the sum of the ancillae needed to store the resource state $|\psi_{q+1}\rangle$, the ancillae needed for the phase-estimation step using the method of Ref. [16] and any ancillae needed to implement the carry logic incrementer circuit [85] and finally the N target qubits. These spatial overheads sum to

$$\begin{aligned}N_{\rm qubits}&=N+(q+1)+2\lceil\log(r+1)\rceil-1+q\\&\lesssim N+2\log(N_{\rm exp}r)+2\\&\lesssim N+2\log\left(\frac{2\lambda^2}{\epsilon^2}\right)+2.\end{aligned} \tag{D17}$$

The scaling of this algorithm is quartically worse than the cost of applying qubitized phase estimation. These comparisons suggest at first glance that randomized simulations are not a competitive technique for simulating such dynamical systems. However, the results in Ref. [18] suggest that incorporating randomness to simulate low-importance terms and using conventional methods to simulate high-importance terms may be a much more profitable approach to using randomized simulation methods for challenge problems in chemistry.

The specific numbers for the Reiher *et al.* and Li *et al.* Hamiltonians can be found using Eqs. (D16) and (D17) together with the λ values from the sparse Hamiltonian simulation method. Since qDRIFT does not have quantum costs that depend at all on the number of terms in the Hamiltonian we take the smallest thresholds considered in Tables IX and X (5×10^{-5}) for the Reiher *et al.* and Li *et al.* Hamiltonians, respectively. This corresponds to $\lambda=2183.6$ and $\lambda=1600.9$, respectively. Taking chemical accuracy of 0.0016 hartree as our target accuracy then yields $N_{\rm Toff}=1.8\times10^{28}$ for the Reiher Hamiltonian and $N_{\rm Toff}=5.2\times10^{27}$ for the Li Hamiltonian. Further, the Reiher Hamiltonian requires 168 logical qubits to simulate within this accuracy whereas the Li Hamiltonian would require approximately 211 logical qubits.

2. Confidence intervals for the eigenphases

Although the presence of (possibly) fat tails for the phase-estimation distribution using qDRIFT prevents our

analysis from yielding low-cost estimates for achieving a fixed mean squared error in the estimate of the ground-state energy for FeMoCo, such an estimate may not strictly speaking be necessary. The reason is that one may instead aim to obtain a confidence interval of small size, rather than a small mean-square error. Alternatively one may wish to consider repeated application of the quantum algorithm with sampling of the measurement results. In that case the root-mean square error is only important if one is taking the average of the measurement results, but one can instead use the Hodges-Lehmann estimator [106], which can have small root-mean square error despite a large root-mean square error in the individual samples. One could also use classically obtained prior knowledge about the ground-state energy to reject values that are outside the range of possible values, thereby reducing the impact of the fat tails.

In the following we consider the complexity when aiming to obtain a confidence interval of small size, then the Hodges-Lehmann estimator. An innovation we introduce here is a form of phase estimation that uses a Kaiser window to optimally yield a sample that is with 95% probability within chemical accuracy of the ground state (although other levels of confidence can also be found numerically using the same strategy). To explain this method, we first review the formalism of phase estimation.

In canonical phase estimation (as opposed to iterative phase estimation) we have a state of the form

$$|\psi(\phi)\rangle = \sum_{j=0}^{r-1} e^{ij\phi}\psi_j\,|j\rangle\,. \tag{D18}$$

The inner product with the phase state

$$|\hat{\phi}\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{ij\hat{\phi}}\,|j\rangle\,, \tag{D19}$$

where $\hat{\phi}$ is the estimator for ϕ (rather than an operator) gives

$$\langle\hat{\phi}|\psi(\phi)\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{ij(\phi-\hat{\phi})}\psi_j\,. \tag{D20}$$

Let us put

$$x_j = 2j/r - 1 + 1/r \tag{D21}$$

so $x_0 = -1 + 1/r$ and $x_{r-1} = 1 - 1/r$, which gives

$$\langle\hat{\phi}|\psi(\phi)\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{i(x_j+1-1/r)(\phi-\hat{\phi})r/2}\psi_j\,. \tag{D22}$$

Putting $\omega = (\phi - \hat{\phi})r/2$, we have

$$\langle\hat{\phi}|\psi(\phi)\rangle\; e^{-i(1-1/r)\omega} = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{ix_j\omega}\psi_j\,. \tag{D23}$$

Now you can define

$$\psi(x) = \sum_j \psi_j\,\delta[x + 1 - (2j+1)/r], \tag{D24}$$

which gives

$$\langle\hat{\phi}|\psi(\phi)\rangle\; e^{-i(1-1/r)\omega} = \frac{1}{\sqrt{r}}\int dx\, e^{ix\omega}\psi(x)\,. \tag{D25}$$

Because $\psi(x)$ can be obtained by multiplying by a comb function with spacing $2/r$, $|\langle\hat{\phi}|\psi(\phi)\rangle|$ as a function of ω must repeat with spacing πr (that is, it is effectively convolved with a comb).

To consider the scaling in the limit of large r, it is convenient to consider a continuous ψ that is nonzero in the interval $[-1, 1]$, which would give $|\langle\hat{\phi}|\psi(\phi)\rangle|$ that is non-repeating in ω, and consider the variance of ω when using $|\langle\hat{\phi}|\psi(\phi)\rangle|^2$ as a probability distribution. The quantity $\langle\hat{\phi}|\psi(\phi)\rangle$ is equivalent to a Fourier transform of $\psi(x)$, and the standard function on the interval $[-1, 1]$ that minimizes the variance of its Fourier transform is

$$\psi(x) = \begin{cases} \cos(\pi x/2) & |x| \le 1, \\ 0 & |x| > 1. \end{cases} \tag{D26}$$

The Fourier transform is

$$\frac{\sqrt{8\pi}\cos\omega}{\pi^2 - 4\omega^2}. \tag{D27}$$

The variance of ω can be found as

$$\int d\omega\,\omega^2 \frac{8\pi\cos^2\omega}{(\pi^2 - 4\omega^2)^2} = \frac{\pi^2}{4}. \tag{D28}$$

In the case of finite r we note that $\omega = (\phi - \hat{\phi})r/2$, so the variance in the phase estimate is divided by $r^2/4$, giving π^2/r^2.

If we want to find the 95% confidence interval, we need to solve

$$\int_{-a}^{a} d\omega\,\omega^2 \frac{8\pi\cos^2\omega}{(\pi^2 - 4\omega^2)^2} = 0.95\,. \tag{D29}$$

Numerically solving gives $a = 2.863325$, so the ratio of the size of the 95% confidence interval to the standard deviation is 1.822849, which is a bit less than the ratio for Gaussians of 1.959964.

If we want to do better, then we can use the Kaiser window. The square of the Fourier transform of the Kaiser window on $[-1, 1]$ is proportional to

$$\frac{\sinh^2\sqrt{\alpha^2-\omega^2}}{\alpha^2-\omega^2}, \tag{D30}$$

where we can adjust α to optimize the size of the confidence interval. Numerically solving gives the narrowest 95% confidence interval as $a = 2.542853$ with $\alpha = 2.179411$.

Now, what we are interested in is the minimum cost when we perform qDRIFT, and allow some probability of error from the qDRIFT and some probability from the phase estimation. With r the number of repetitions, the error from qDRIFT is

$$2r\lambda^2 t^2. \tag{D31}$$

Note that this is equivalent to taking $k = 1$ in the notation of the previous subsection, which means that the qDRIFT interval is the same as that used for phase estimation. Therefore, for smaller t, the error due to qDRIFT is smaller, but the error in estimating the *energy* is scaled by t, so smaller t leads to larger error in the energy. The total confidence interval of 95% can be obtained with

$$\int_0^{\epsilon tr} d\omega \frac{\sinh^2\sqrt{\alpha^2-\omega^2}}{\alpha^2-\omega^2} = \left(0.95 + r\lambda^2 t^2\right)\int_0^{\infty} d\omega \frac{\sinh^2\sqrt{\alpha^2-\omega^2}}{\alpha^2-\omega^2}. \tag{D32}$$

The reason for the upper limit on the integral of ϵtr is that the allowable error in the energy is ϵ. Since $E = \phi/2t$, that means the allowable error in the phase is $2\epsilon t$. Since $\omega = (\phi - \hat{\phi})r/2$, the allowable error in ω is ϵtr. The integral on the right-hand side is for normalization. The expression $0.95 + r\lambda^2 t^2$ indicates that the confidence interval due to the phase estimation needs an additional $r\lambda^2 t^2$ to account for the additional probability of error from qDRIFT. The probability of error outside the interval due to qDRIFT can increase only by half of $2r\lambda^2 t^2$, because increasing probability outside the interval by $r\lambda^2 t^2$ means that the probability elsewhere needs to be increased by $r\lambda^2 t^2$, resulting in the total error in the probability distribution of $2r\lambda^2 t^2$.

We then aim to solve for $r = N_{\text{exp}}$ for any given t and α, and minimize it. We set $a = \epsilon tr$ and $\delta = r\lambda^2 t^2$, and find

$$\frac{\epsilon^2 r}{\lambda^2} = \frac{a^2}{\delta} \tag{D33}$$

so

$$N_{\text{exp}} = \frac{\lambda^2 a^2}{\epsilon^2 \delta}. \tag{D34}$$

Note that this shows N_{exp} is proportional to λ^2/ϵ^2, which gives the scaling of the complexity up to logarithmic factors (from synthesizing rotations). Minimizing, we get the minimum value of a^2/δ as 304.744, with $\alpha = 3.05961$, $a = 3.37625$, and $\delta = 0.0374053$. We take $\epsilon = 0.0016$, so for Reiher with $\lambda = 2183.6$ we get $N_{\text{exp}} = 5.67598 \times 10^{14}$, and for Li with $\lambda = 1600.9$ we get $N_{\text{exp}} = 3.05087 \times 10^{14}$.

Next, note that the rotations for the exponentials are by an angle λt, which can be performed more efficiently if λt is equal to π divided by a power of 2. From the above $\lambda t = \epsilon\delta/\lambda a$, so for the two cases we get $\lambda\tau$ equal to $\pi/2^{28}$ times 0.693643 and 0.946117. We can round the first to $11/16$, and the second to $7/8$. That means the first would need 31 Toffolis per step, and the second would need 30 Toffolis per step for the exponential. There will also be a single Toffoli per step for the unary iteration on the control register for the phase estimation, and another Toffoli for using that control register to control between forward and reverse evolution. Then we can minimize r over α for the constant product λt to give the following.

1. For Reiher, the minimum N_{exp} is 5.67874×10^{14} for $\alpha = 3.03125$ with $\lambda\tau = (11/16)\pi/2^{28}$, giving a number of Toffolis 1.9×10^{16} and a number of logical qubits at most equal to 270.
2. For Li, the minimum N_{exp} is 3.12299×10^{14} for $\alpha = 2.8794$ with $\lambda\tau = (7/8)\pi/2^{28}$, giving a number of Toffolis 1.0×10^{16} and a number of logical qubits at most equal to 310.

These results are substantially better than those predicted using the bounds in Campbell's work. Assuming a 95% confidence interval, the number of exponentials is given for a probability of failure $P_0 = 0.05$ [see Eq. (45) in the Supplemental Material of Ref. [52]]

$$N_{\text{exp}} \leq \frac{27\pi^2\lambda^2}{2\epsilon^2 P_0^3}, \tag{D35}$$

which yields 4.0×10^{18} exponentials for the Reiher Hamiltonian and 2.1×10^{18} exponentials for the Li Hamiltonian.

An alternative approach is to consider the case that the Hodges-Lehmann estimator [106] is used instead of the mean when combining results of multiple runs of the quantum algorithm. The Hodges-Lehmann estimator is based on taking all pairwise means of the samples, and finding the median of those pairwise means. For M samples, M times the mean-square error of the estimate is asymptotically given by (see p. 245 of Ref. [107])

$$\frac{1}{12\left[\int d\omega\, p^2(\omega)\right]^2} \tag{D36}$$

for symmetric probability distribution $p(\omega)$. A reasonable goal in the case of qDRIFT is for this quantity to be

equal to ϵ^2, rather than the variance, because this quantity will correspond to the asymptotic variance under multiple samples. For the case of the probability distribution

$$p(\omega) = \frac{8\pi \cos^2 \omega}{(\pi^2 - 4\omega^2)^2}, \tag{D37}$$

the asymptotic value for the Hodges-Lehmann mean-square error is 2.38991, which is about 3% below the mean-square error for an individual sample of $\pi^2/4$.

In order to bound the effect of qDRIFT error on the Hodges-Lehmann estimator, when modifying the probability distribution the maximum reduction in $\int d\omega\, p^2(\omega)$ will be reducing the maximum probability, while increasing the probability for large values of ω by very small amounts over a wide region. It should be noted that this assumes that the error is symmetric. Calling the modified probability distribution q, we therefore need

$$\frac{1}{12r^2t^2\left[\int d\omega\, q^2(\omega)\right]^2} \leq \epsilon^2, \tag{D38}$$

while

$$\int d\omega\, |p(\omega) - q(\omega)| \leq 2r\lambda^2 t^2. \tag{D39}$$

The choice for r is then

$$r = \frac{\lambda^2}{6\epsilon^2\left[\int d\omega\, q^2(\omega)\right]^2 \int d\omega\, |p(\omega) - q(\omega)|}. \tag{D40}$$

The optimal choice of q is to truncate the probability distribution at 0.648057 times the maximum of p, which gives the multiplying factor on λ^2/ϵ^2 as 35.5192; that is,

$$N_{\text{exp}} = 35.5192\frac{\lambda^2}{\epsilon^2}. \tag{D41}$$

Therefore, with this estimate, we have the same λ^2/ϵ^2 scaling as for a confidence interval, but a considerably smaller constant factor. In a similar way as for the confidence interval, we can adjust the parameters so that λt is a convenient number for the rotations.

1. For the Reiher case we find the truncation can be adjusted to 0.683740, giving $\lambda t = \pi/2^{26}$ and $r = 6.7 \times 10^{13}$. The rotations take 25 Toffolis, so there are 1.8×10^{15} Toffolis for the simulation and 250 qubits.
2. For the Li orbitals, choosing a truncation 0.645979 gives $\lambda t = 3\pi/2^{27}$ and $r = 3.6 \times 10^{13}$. The rotation takes 26 Toffolis, so the total cost is 1.0×10^{15} Toffolis for the simulation and 296 qubits.

In this approach there is about a further order of magnitude improvement over the resource estimate based on the confidence interval. This is because it is allowing considerably larger error due to the qDRIFT, on the assumption that it can be eliminated by sampling. This estimate of the complexity does assume that the qDRIFT error is symmetric, so is not completely rigorous.

APPENDIX E: DIRECT QUBITIZATION OF THE STANDARD TENSOR HYPERCONTRACTION REPRESENTATION

In this Appendix we discuss the cost of quantum simulation when qubitization is applied directly to the tensor hypercontraction representation of Eq. (4). While the result is also an algorithm with gate complexity $\widetilde{\mathcal{O}}(N\lambda/\epsilon)$, the associated λ is much larger than the λ for the method described in the main body of the paper. We include this Appendix for completeness. Our algorithm uses qubitization as described in Sec. III A of the main paper. Here, we describe how one would implement SELECT and PREPARE oracles. As we show, each of these primitives has complexity no more than $\widetilde{\mathcal{O}}(N)$; thus, consistent with Eq. (23), the overall complexity is no more than $\widetilde{\mathcal{O}}(N\lambda/\epsilon)$.

1. Specification of oracles for qubitization and implementation of Hamiltonian selection oracle

To specify the requirements for the qubitization oracles that we query to block encode the THC Hamiltonian, we can use a similar application of the Jordan-Wigner transformation as in Appendix A to give

$$H = \frac{1}{2}\sum_{\sigma\in\{\uparrow,\downarrow\}}\sum_{p,q=1}^{N/2} T'_{pq} Q_{pq\sigma} + \frac{1}{8}\sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{p,q,r,s=1}^{N/2} V_{pqrs} Q_{pq\alpha} Q_{rs\beta}\,, \tag{E1}$$

where $Q_{pq\alpha}$ is as defined in Eq. (A4). For THC, we then replace V_{pqrs} with the approximation

$$G_{pqrs} = \sum_{\mu,\nu=1}^{M} \chi_p^{(\mu)}\chi_q^{(\mu)}\zeta_{\mu\nu}\chi_r^{(\nu)}\chi_s^{(\nu)}. \tag{E2}$$

That gives a λ value of

$$\lambda = \sum_{p,q=1}^{N/2}\left|T_{pq} + \sum_{r=1}^{N/2} V_{pqrr}\right| + \frac{1}{2}\sum_{p,q,r,s=1}^{N/2}\sum_{\mu,\nu=1}^{M}\left|\chi_p^{(\mu)}\chi_q^{(\mu)}\zeta_{\mu\nu}\chi_r^{(\nu)}\chi_s^{(\nu)}\right|. \tag{E3}$$

The state to be prepared is of the form

$$|0\rangle \sum_{p,q=1}^{N/2} \sqrt{\frac{|T'_{pq}|}{\lambda}} |\theta^T_{pq}\rangle |p\rangle |q\rangle |0\rangle |0\rangle |0\rangle |0\rangle |0\rangle |0\rangle |0\rangle |0\rangle + |1\rangle \sum_{p,q,r,s=1}^{N/2} \sum_{\mu,\nu=1}^{M} \sqrt{\frac{\left|\chi^{(\mu)}_p \chi^{(\mu)}_q \zeta_{\mu\nu} \chi^{(\nu)}_r \chi^{(\nu)}_s\right|}{2\lambda}} \times |\theta^{\zeta}_{\mu\nu}\rangle |\mu\rangle |\nu\rangle |\theta^{(\mu)}_p\rangle |p\rangle |\theta^{(\mu)}_q\rangle |q\rangle |\theta^{(\nu)}_r\rangle |r\rangle |\theta^{(\nu)}_s\rangle |s\rangle . \tag{E4}$$

Here θ^T_{pq} gives the sign of T' as before, $\theta^{(\mu)}_p$ gives the sign of $\chi^{(\mu)}_p$, and $\theta^{\zeta}_{\mu\nu}$ gives the sign of $\zeta_{\mu\nu}$. The key thing to note is that the second part of this state, corresponding to the two-body terms, can be factorized as

$$\frac{1}{\sqrt{2\lambda}} |1\rangle \sum_{\mu,\nu=1}^{M} \sqrt{|\zeta_{\mu\nu}|} |\theta^{\zeta}_{\mu\nu}\rangle |\mu\rangle |\nu\rangle \left(\sum_{p=1}^{N/2} \sqrt{|\chi^{(\mu)}_p|} |\theta^{(\mu)}_p\rangle |p\rangle\right) \times \left(\sum_{q=1}^{N/2} \sqrt{|\chi^{(\mu)}_q|} |\theta^{(\mu)}_q\rangle |q\rangle\right) \otimes \left(\sum_{r=1}^{N/2} \sqrt{|\chi^{(\nu)}_r|} |\theta^{(\nu)}_r\rangle |r\rangle\right) \left(\sum_{s=1}^{N/2} \sqrt{|\chi^{(\nu)}_s|} |\theta^{(\nu)}_s\rangle |s\rangle\right). \tag{E5}$$

The key feature of this state that makes it easier to prepare is that it factorizes. One may therefore start by preparing a state on μ and ν, then controlled on μ and ν prepare the states in brackets.

2. Using the structure of tensor hypercontraction to implement the qubitization state preparation

In preparing the overall state in Eq. (E4), one can first prepare the state on the first four registers, with amplitudes proportional to $\sqrt{|T'_{pq}|}$ for the one-body term and $\sqrt{|\zeta_{\mu\nu}|}$ for the two-body term. That is, we first prepare the state

$$|0\rangle \sum_{p,q=1}^{N/2} \sqrt{\frac{|T'_{pq}|}{\lambda}} |\theta^T_{pq}\rangle |p\rangle |q\rangle + |1\rangle \sum_{\mu,\nu=1}^{M} \sqrt{\frac{|\zeta_{\mu\nu}|}{2\lambda}} |\theta^{\zeta}_{\mu\nu}\rangle |\mu\rangle |\nu\rangle . \tag{E6}$$

After this preparation, for the two-body term flagged by 1 on the first qubit, we need to perform four applications of the mapping

$$\text{PREPARE}_{\chi} |0\rangle^{\log(N/2)} |\mu\rangle \mapsto \sum_{p=1}^{N/2} \sqrt{|\chi^{(\mu)}_p|} |\theta^{(\mu)}_p\rangle |p\rangle |\mu\rangle . \tag{E7}$$

This is a preparation on two registers, but this time there is not symmetry.

For the first preparation, the steps to be performed are as follows.

1. Rotate the first qubit to give the appropriate relative weighting between the one- and two-body terms.
2. For 1 on the first qubit, prepare an equal superposition over μ and ν for $1 \le \mu \le \nu \le M$, or for 0 on the first qubit prepare an equal superposition with the restrictions $1 \le \mu \le \nu \le N/2$. That can be performed using inequality tests and amplitude amplification in a similar way to that explained in Appendix B. The difference is that instead of just using the inequality test $\nu \le M$, you use a controlled inequality test of $\nu \le N/2$ for 0 on the first qubit, or $\nu \le M$ for 1 on the first qubit. The extra inequality tests increase the Toffoli cost to $8n_M + 2b_r + \mathcal{O}(1)$, where b_r are the bits of precision for rotation on an ancilla and $n_M = \lceil \log M \rceil$.
3. Compute $\mu(\mu-1)/2 + \nu$, which can be performed with complexity $n_M^2 + n_M - 1$. In the case of 0 on the first qubit (for the one-body term), add $M(M+1)/2$ to yield a contiguous register. That addition has complexity $2n_M + \mathcal{O}(1)$.
4. Perform a QROM using this contiguous register to alt values of μ and ν, keep probabilities, and two values (one is the alt value) of the sign $\theta^{\zeta}_{\mu\nu}$. The output size for the QROM is

$$b_{\zeta} = 2n_M + \aleph + 2, \tag{E8}$$

where $\aleph$ are the bits of precision for $\zeta_{\mu\nu}$. The complexity of the QROM is

$$\left\lceil \frac{L_{\zeta}}{k_{\zeta}} \right\rceil + b_{\zeta}(k_{\zeta} - 1), \tag{E9}$$

where

$$L_{\zeta} = M(M+1)/2 + N^2/8 + N/4 \tag{E10}$$

and k_{ζ} must be a power of 2.
5. Perform an inequality test between the keep value and another register in an equal superposition, with cost $\aleph$.
6. Perform a swap between μ, ν and the alt values controlled on the result of the inequality test, with cost $2n_M$. The signs can be applied with Clifford gates.
7. Apply a swap between the μ and ν registers controlled by a qubit in a $|+\rangle$ state, with cost n_M.

The total complexity is then

$$C_{P_\zeta} = \left\lceil \frac{L_\zeta}{k_\zeta} \right\rceil + b_\zeta k_\zeta + n_M^2 + 12n_M + b_r + \mathcal{O}[\log(1/\epsilon_{\rm rot})], \tag{E11}$$

where $\epsilon_{\rm rot}$ is the accuracy required for the rotation on the first qubit. In inverting the state preparation, all costs are the same except the QROM cost, which is reduced to give a total cost

$$C_{P_\zeta^\dagger} = \left\lceil \frac{L_\zeta}{k'_\zeta} \right\rceil + k'_\zeta + n_M^2 + 14n_M + b_r + \aleph + \mathcal{O}[\log(1/\epsilon_{\rm rot})]. \tag{E12}$$

For the preparation given in Eq. (E7), the considerations are similar, except we do not take advantage of symmetry, so $L_\chi = MN/2$. There the preparation needs a QROM to run through all values of p and μ, and it is convenient to use the QROM for two registers as in Appendix G. The total Toffoli costs are as follows.

1. Prepare an equal superposition over $N/2$ basis states in the p register, with cost $3n_N - 3\eta + 2b_r - 9$, where η is the largest number such that 2^η is a factor of $N/2$, and b_r is bit of precision for rotation on an ancilla.
2. Because we need only to output alternate values of p (and not μ), the output size for the QROM is

$$b_\chi = n_N + \aleph. \tag{E13}$$

Therefore, we can apply the QROM with cost (using the method for separate registers in Appendix G)

$$\left\lceil \frac{M}{k_{\chi 1}} \right\rceil \left\lceil \frac{N}{2k_{\chi 2}} \right\rceil + b_\chi(k_{\chi 1}k_{\chi 2} - 1). \tag{E14}$$

3. Perform the inequality test with cost $\aleph$.
4. The controlled swap does not need to touch the register with μ, because we are just preparing a superposition over p for a given μ. Therefore, the controlled swap has cost n_N.

The total costs are then

$$C_{P_\chi} = \left\lceil \frac{M}{k_{\chi 1}} \right\rceil \left\lceil \frac{N}{2k_{\chi 2}} \right\rceil + b_\chi k_{\chi 1}k_{\chi 2} + 3n_N - 3\eta + 2b_r + \mathcal{O}(1) \tag{E15}$$

for preparation and

$$C_{P_\chi^\dagger} = \left\lceil \frac{M}{k'_{\chi 1}} \right\rceil \left\lceil \frac{N}{2k'_{\chi 2}} \right\rceil + k'_{\chi 1}k'_{\chi 2} + b_\chi + 3n_N - 3\eta + 2b_r + \mathcal{O}(1) \tag{E16}$$

for inverse preparation. It is also possible to combine the preparation of some of the equal superposition states together, which would give slightly different costs. We do not analyse that here, because the method that is best tends to depend on the particular example.

We can now combine all of the Toffoli costs. Clearly, we need to prepare and unprepare the χ state four times and prepare and unprepare the ζ state once. Thus the overall cost of preparation and unpreparation is

$$C_P + C_{P^\dagger} = C_{P_\zeta} + C_{P_\zeta^\dagger} + 4C_{P_\chi} + 4C_{P_\chi^\dagger}. \tag{E17}$$

For the total cost, there will also be $4N$ for the cost of implementing the SELECT operation, and $2n_M + 4n_N + 5\aleph + \mathcal{O}(1)$ for reflections on the control registers to construct the overall step of the quantum walk. There are $2n_M + 4n_N + \mathcal{O}(1)$ qubits that the state is prepared on, and we need to reflect on the $5\aleph$ qubits used for equal superposition states as well.

The logical qubit costs are as follows, where we omit a number of single ancillas (about 20) for simplicity.

1. The $2\lceil\log(\mathcal{I}+1)\rceil - 1$ qubits for the control register for state preparation and unary iteration on that register.
2. The N system registers.
3. For storing μ, ν, p, q, r, s there are $2n_M + 4n_N$ qubits needed.
4. The $5\aleph$ from registers that are used as comparison registers for the inequality tests in the coherent alias sampling.
5. The phase-gradient state has b_r bits.
6. The contiguous register for the ζ state preparation, which has $\lceil \log L_\zeta \rceil$ qubits.
7. The output of the QROM in the ζ state preparation uses b_ζ bits.
8. The QROM in the ζ state preparation uses another $b_\zeta(k_\zeta - 1) + \lceil\log(L_\zeta/k_\zeta)\rceil$ temporary qubits. At this point the number of qubits used would normally be at a maximum, and one can find the number of qubits needed by adding this number of qubits to the numbers in the other items listed above. Otherwise, one would ignore this cost, and add the costs given below.
9. There are $4b_\chi$ qubits needed for the outputs of the four QROMs for the four χ-state preparations.
10. There are another $b_\chi(k_{\chi 1}k_{\chi 2} - 1) + \lceil\log(M/k_{\chi 1})\rceil \lceil\log(N/2k_{\chi 2})\rceil$ temporary qubits used in the final QROM for χ-state preparation.

Adding these numbers of qubits gives us

$$2\lceil\log(\mathcal{I}+1)\rceil + N + 2n_M + 4n_N + 5\aleph + b_r + \lceil\log L_\zeta\rceil + b_\zeta + \mathcal{O}(1) + \max\left[b_\zeta(k_\zeta - 1) + \lceil\log(L_\zeta/k_\zeta)\rceil\,,\ 3b_\chi + b_\chi k_{\chi 1} k_{\chi 2} + \lceil\log(M/k_{\chi 1})\rceil\lceil\log(N/2k_{\chi 2})\rceil\right]. \quad \text{(E18)}$$

For realistic parameters, the first expression in the max should be taken, corresponding to the largest number of qubits being used in the ζ-state preparation.

3. Comparison of λ associated with direct qubitization and nonorthogonal qubitization

Here we present numerical data for the two FeMoCo Hamiltonians, which compare our proposed use of THC [Eq. (13)] against the naïve use of THC as defined in Eq. (9) and described in this Appendix. As mentioned in the main text, the use of Eq. (9) results in the two-body λ, i.e., λ_2, that is a few orders of magnitude greater than that of Eq. (13). In particular, we compare the values of Eqs. (13) and (9) as a function of M. As shown in Fig. 19, significant reduction in λ_2 is observed going from Eq. (9) to Eq. (13). Furthermore, in Fig. 20, we illustrate the same point for hydrogen chain and H_4. Compared to the naïve approach, we not only achieve more than an order of magnitude reduction in λ_2 but also achieve a reduced scaling of λ_2 with respect to system size and the number of basis functions. Specifically, we note that the empirical scaling based on the linear fits on a log scale suggests that we achieve $N_H^{3.16}$ to $N_H^{1.16}$ size scaling reduction in hydrogen chain and $N^{4.06}$ to $N^{2.09}$ basis scaling reduction in H_4. We argue that the use of THC in qubitization is only successful when utilizing its nonorthogonal form as proposed in this work. This is one of the key observations in our work.

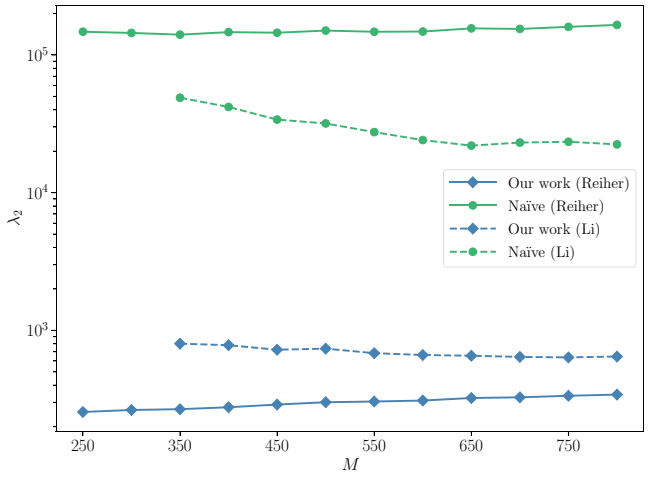

FIG. 19. Plots of λ_2 versus M for two FeMoCo Hamiltonians (Reiher and Li) where λ_2 is computed by Eq. (9) (labeled "Naïve") and Eq. (13) (labeled "Our work"). These numerical values are based on the THC factors that produced the numerical data in Tables IV and V.

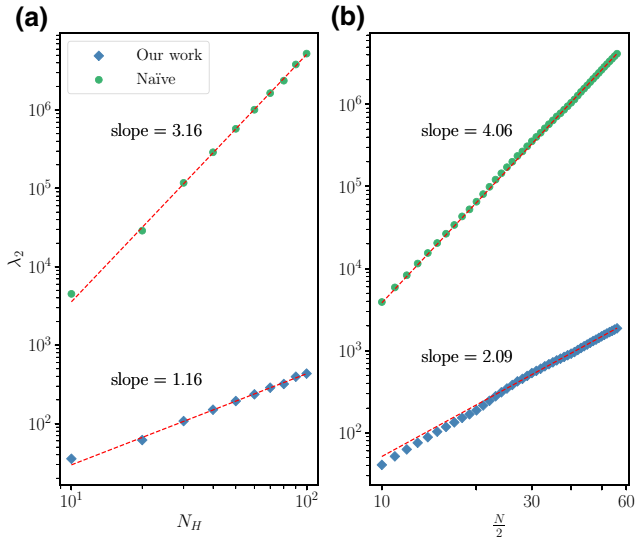

FIG. 20. Plot of λ_2 versus N_H for hydrogen chain and versus $N/2$ for H_4, where λ_2 is computed by Eq. (9) (labeled "Naïve") and Eq. (13) (labeled "Our work"). These numerical values are based on the THC factors that produce the numerical data in Fig. 9. R^2 is greater than 0.99 in all fits. Note that the slope of "Our work" in (a) is slightly different from what is reported in Table VII because (a) uses only λ_2 whereas Table VII is based on total λ.

APPENDIX F: COMPUTING CONTIGUOUS REGISTERS

In this Appendix we show the formula for computing contiguous registers. This step is used in state preparations for both the main algorithm of this paper as well as a modification we introduce to the double factorization of von Burg *et al.* [10], which is not discussed in their paper but which we believe is required for their algorithm to execute correctly. The need for this is first discussed in Eq. (29) of this paper. For contiguous registers we need to compute

$$p(p+1)/2 + q, \quad \text{(F1)}$$

where p and q have the same number of bits, which we denote n. We show that this formula can be computed using $n^2 + n - 1$ Toffolis, considerably simplifying the analysis of the complexity. In this Appendix we take p and q to start from 0 rather than 1 as is used in the discussion of the Hamiltonians. That is why the formula is $p(p+1)/2 + q$ rather than $p(p-1)/2 + q$.

Writing p and q in terms of their bits

$$p = \sum_{j=0}^{n-1} p_j 2^j, \qquad q = \sum_{j=0}^{n-1} q_j 2^j, \quad \text{(F2)}$$

we have

$$\begin{aligned} p^2 &= \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} p_j p_k 2^{j+k} \\ &= \sum_{j=0}^{n-1} p_j 2^{2j} + 2\sum_{j=0}^{n-1}\sum_{k=0}^{j-1} p_j p_k 2^{j+k} \\ &= \sum_{j=0}^{n-1} p_j 2^{2j} + \sum_{\ell=0}^{2n-3} 2^{\ell+1} \sum_{j=\max(0,\ell-n+1)}^{\lfloor(\ell-1)/2\rfloor} p_{\ell-j} p_j , \end{aligned} \tag{F3}$$

where $\lfloor(\ell-1)/2\rfloor$ is to ensure that $\ell - j > j$. Breaking this up into odd and even gives

$$p^2 = p_0 + \sum_{\ell=1}^{n-2} 2^{2\ell+1} \sum_{j=\max(0,2\ell-n+1)}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n-1} 2^{2\ell} \left(p_\ell + \sum_{j=\max(0,2\ell-n)}^{\ell-1} p_{2\ell-1-j} p_j \right). \tag{F4}$$

That gives

$$p^2 + p = 2p_0 + \sum_{\ell=1}^{n-1} 2^\ell p_\ell + \sum_{\ell=1}^{n-2} 2^{2\ell+1} \sum_{j=\max(0,2\ell-n+1)}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n-1} 2^{2\ell} \left(p_\ell + \sum_{j=\max(0,2\ell-n)}^{\ell-1} p_{2\ell-1-j} p_j \right). \tag{F5}$$

Dividing by 2 then gives

$$p(p+1)/2 = p_0 + \sum_{\ell=1}^{n-1} 2^{\ell-1} p_\ell + \sum_{\ell=1}^{n-2} 2^{2\ell} \sum_{j=\max(0,2\ell-n+1)}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=\max(0,2\ell-n)}^{\ell-1} p_{2\ell-1-j} p_j \right). \tag{F6}$$

Adding q gives

$$\begin{aligned} p(p+1)/2 + q &= \sum_{\ell=0}^{n-1} 2^\ell q_\ell + p_0 + \sum_{\ell=1}^{n-1} 2^{\ell-1} p_\ell + \sum_{\ell=1}^{n-2} 2^{2\ell} \sum_{j=\max(0,2\ell-n+1)}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=\max(0,2\ell-n)}^{\ell-1} p_{2\ell-1-j} p_j \right) \\ &= q_0 + p_0 + p_1 + \sum_{\ell=1}^{n-2} 2^\ell (q_\ell + p_{\ell+1}) + 2^{n-1} q_{n-1} \\ &\quad + \sum_{\ell=1}^{n-2} 2^{2\ell} \sum_{j=\max(0,2\ell-n+1)}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=\max(0,2\ell-n)}^{\ell-1} p_{2\ell-1-j} p_j \right). \end{aligned} \tag{F7}$$

The next steps depend on whether n is odd or even. First, for n even, $n-1 = 2\ell - 1$ for $\ell = n/2$. Then we can write

$$\begin{aligned} p(p+1)/2 + q = q_0 + p_0 + p_1 + \sum_{\ell=1}^{n-2} 2^\ell (q_\ell + p_{\ell+1}) + 2^{n-1} q_{n-1} + \sum_{\ell=1}^{n/2-1} 2^{2\ell} \sum_{j=0}^{\ell-1} p_{2\ell-j} p_j \\ + \sum_{\ell=n/2}^{n-2} 2^{2\ell} \sum_{j=2\ell-n+1}^{\ell-1} p_{2\ell-j} p_j + \sum_{\ell=1}^{n/2-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=0}^{\ell-1} p_{2\ell-1-j} p_j \right) \\ + 2^{n-1} \left(p_{n/2} + \sum_{j=0}^{n/2-1} p_{n-1-j} p_j \right) + \sum_{\ell=n/2+1}^{n-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=2\ell-n}^{\ell-1} p_{2\ell-1-j} p_j \right). \end{aligned} \tag{F8}$$

This can be written as

$$p(p+1)/2+q = q_0+p_0+p_1 \tag{F9a}$$

$$+\sum_{\ell=1}^{n/2-1} 2^{2\ell-1}\left(q_{2\ell-1}+p_{2\ell}+p_\ell+\sum_{j=0}^{\ell-1} p_{2\ell-1-j}p_j\right) \tag{F9b}$$

$$+\sum_{\ell=1}^{n/2-1} 2^{2\ell}\left(q_{2\ell}+p_{2\ell+1}+\sum_{j=0}^{\ell-1} p_{2\ell-j}p_j\right) \tag{F9c}$$

$$+2^{n-1}\left(q_{n-1}+p_{n/2}+\sum_{j=0}^{n/2-1} p_{n-1-j}p_j\right) \tag{F9d}$$

$$+\sum_{\ell=n/2}^{n-2} 2^{2\ell}\sum_{j=2\ell-n+1}^{\ell-1} p_{2\ell-j}p_j \tag{F9e}$$

$$+\sum_{\ell=n/2}^{n-2} 2^{2\ell+1}\left(p_{\ell+1}+\sum_{j=2\ell-n+1}^{\ell-1} p_{2\ell-j}p_{j+1}\right). \tag{F9f}$$

The last line has shifted ℓ to simplify the following discussion. This expression has split into lines as follows.

1. In Eq. (F9a) three bits, which are added together.
2. In Eq. (F9b) bits, which are added together and multiplied by odd powers $2^{2\ell-1}$ below 2^{n-1}. There are $\ell+3$ of these bits to be added together.
3. In Eq. (F9c) bits, which are added together and multiplied by *even* powers of $2^{2\ell}$ below 2^{n-1}. There are $\ell+2$ of these bits to be added together.
4. In Eq. (F9d) there are $n/2+2$ bits to be added together and multiplied by 2^{n-1}.
5. In Eq. (F9e) there are $n-\ell-1$ bits to be added together and multiplied by even powers $2^{2\ell}$ above 2^{n-1} up to 2^{2n-4}.
6. In Eq. (F9f) there are $n-\ell$ bits to be added together and multiplied by odd powers $2^{2\ell+1}$ above 2^{n-1} up to 2^{2n-3}.

Now we can apply the approach used for bit sums in Appendix A of Ref. [18]. We separate the bits into those that are multiplied by 1, 2, 4, 8, and so on. We add triples of bits at each level using the adder in Fig. 4(b) from Ref. [85]. Adding each triple of bits takes one Toffoli, reduces the number of bits at that level by 2, and increases the number of bits at the next level by 1. So, if there are m bits at one level, then the number of Toffolis needed for that level is $\lfloor m/2\rfloor$, and there are an additional $\lfloor m/2\rfloor$ carry bits at the next higher level.

Here the first level, those bits multiplied by 1, has only three bits, meaning one Toffoli is needed, and there is one more carry bit at level 2. For level 2, $\ell=1$, and there are $\ell+3=4$ bits. Adding the carry bit, there are now 5 bits, which take 2 Toffolis to sum together, and give another 2 carry bits for the next level, 3, where the bits are multiplied by 2^2. For level 3 with $\ell=1$, there are $\ell+2=3$ bits, and including the 2 carry bits gives 5. These 5 bits can again be added together with 2 Toffolis, giving 2 Toffolis for the next level.

In general, for the lines in Eq. (F9a) to Eq. (F9f) we have the following costings.

1. In Eq. (F9a) the three bits are added with one Toffoli, giving one carry bit.
2. In Eq. (F9b), for term ℓ, we have ℓ carry bits from the previous level. We can see this for the case $\ell=1$ because there was one carry bit from adding the triple of bits from level 1. For $\ell>1$, we have the number of carry bits from Eq. (F9c) with term $\ell-1$, which gave ℓ carry bits. The number of bits to be added together is therefore $\ell+3+\ell$. These bits can be added together with cost $\lfloor(2\ell+3)/2\rfloor=\ell+1$ Toffolis and the same number of carry bits.
3. In Eq. (F9c) for term ℓ, we have $\ell+1$ carry bits coming from Eq. (F9b) with the same value of ℓ. The number of bits is $\ell+2$ plus the $\ell+1$ carry bits, giving a total of $2\ell+3$. These can again be added with cost $\ell+1$, giving $\ell+1$ carry bits.
4. In Eq. (F9d) there are $n/2+2$ bits, plus $n/2$ carry bits from Eq. (F9c), since the last term there has $\ell=n/2-1$. That gives a total of $n+2$ bits to add, which can be added with $n/2+1$ Toffolis and giving the same number of carry bits.

TABLE XV. Here we show how to compute a contiguous register for an even number of bits $n = 6$. The entries p_0, p_0p_1, and so forth show entries with different bits. The entries with $\times$ show bits that are carried from lower levels. The first column shows the level number, the second column shows the multiplying factor (power of 2), the third column shows which line from Eq. (F9a) to Eq. (F9f) this corresponds to, the fourth column shows the ℓ value from that equation, and the last column shows the number of Toffolis. The $\cancel{p_0}$ shows identical bits that are added with no cost, and the resulting bit is shown in blue in the next level as p_0.

Level	Factor	Line of Equation	ℓ	q	p	p_0	p_1	p_2	p_3	p_4	p_5			Toffolis
12	2^{11}										$\times$			
11	2^{10}										$\times$	$\times$		1
10	2^9	(F9f)	4							p_4p_5	p_5	$\times$	$\times$	2
9	2^8	(F9e)	4						p_3p_5	$\times$	$\times$	$\times$		2
8	2^7	(F9f)	3					p_2p_5	p_3p_4	p_4	$\times$	$\times$	$\times$	3
7	2^6	(F9e)	3				p_1p_5	p_2p_4	$\times$	$\times$	$\times$	$\times$		3
6	2^5	(F9d)		q_5		p_0p_5	p_1p_4	p_2p_3	p_3	$\times$	$\times$	$\times$		4
5	2^4	(F9c)	2	q_4	p_5	p_0p_4	p_1p_3	$\times$	$\times$	$\times$				3
4	2^3	(F9b)	2	q_3	p_4	p_0p_3	p_1p_2	p_2	$\times$	$\times$				3
3	2^2	(F9c)	1	q_2	p_3	p_0p_2	$\times$	$\times$						2
2	2	(F9b)	1	q_1	p_2	p_0p_1	p_1	$\times$						2
1	1	(F9a)		q_0	p_1	p_0								1
0	NA				$\cancel{p_0}$	$\cancel{p_0}$								

5. In Eq. (F9e) there are $n - \ell + 1$ carry bits to be included. This can be seen for $\ell = n/2$, because there were $n/2 + 1$ carry bits from Eq. (F9d). For $\ell > n/2$, it can be seen because there are $n - \ell + 1$ carry bits from the sixth line by using $n - \ell'$ with $\ell' = \ell - 1$. That gives a total number of bits $2(n - \ell)$, which can be added together with $n - \ell$ Toffolis and giving $n - \ell$ carry bits.
6. In Eq. (F9f) there are $n - \ell$ bits coming from the fifth line. That gives a total number of bits $2(n - \ell)$, which can be added together with $n - \ell$ Toffolis and giving $n - \ell$ carry bits.
7. Note that for the final term from Eq. (F9f) with $\ell = n - 2$, the number of carry bits is $n - (n - 2) = 2$. These two bits can be added together with one more Toffoli gate.

Now that we have quantified the number of Toffolis at each level, we can add them together to give

$$1 + 2\sum_{\ell=1}^{n/2-1}(\ell+1) + \sum_{\ell=1}^{n/2-1}(\ell+1) + n/2 + 1 + 2\sum_{\ell=n/2}^{n-2}(n-\ell) + 1 = n(n+3)/2 - 1 \tag{F10}$$

Toffolis. There are also $n(n-1)/2$ Toffolis needed to compute all of the products p_jp_k for $j \neq k$. Adding those Toffolis gives the total $n^2 + n - 1$. An illustration of the case of even $n = 6$ is given in Table XV.

Now we consider the case for n odd. We can write

$$\begin{aligned} p(p+1)/2 + q = {} & q_0 + p_0 + p_1 + \sum_{\ell=1}^{n-2} 2^\ell (q_\ell + p_{\ell+1}) + 2^{n-1} q_{n-1} + \\ & + \sum_{\ell=1}^{(n-3)/2} 2^{2\ell} \sum_{j=0}^{\ell-1} p_{2\ell-j}p_j + 2^{n-1} \sum_{j=0}^{(n-3)/2} p_{n-1-j}p_j + \sum_{\ell=(n+1)/2}^{n-2} 2^{2\ell} \sum_{j=2\ell-n+1}^{\ell-1} p_{2\ell-j}p_j \\ & + \sum_{\ell=1}^{(n-1)/2} 2^{2\ell-1} \left(p_\ell + \sum_{j=0}^{\ell-1} p_{2\ell-1-j}p_j \right) + \sum_{\ell=(n+1)/2}^{n-1} 2^{2\ell-1} \left(p_\ell + \sum_{j=2\ell-n}^{\ell-1} p_{2\ell-1-j}p_j \right). \end{aligned} \tag{F11}$$

As before, we rewrite giving

$$p(p+1)/2+q = q_0 + p_0 + p_1 \tag{F12a}$$
$$+ \sum_{\ell=1}^{(n-1)/2} 2^{2\ell-1}\left(p_{2\ell-1}+p_{2\ell}+p_\ell+\sum_{j=0}^{\ell-1} p_{2\ell-1-j}p_j\right) \tag{F12b}$$
$$+ \sum_{\ell=1}^{(n-3)/2} 2^{2\ell}\left(q_{2\ell}+p_{2\ell+1}+\sum_{j=0}^{\ell-1} p_{2\ell-j}p_j\right) \tag{F12c}$$
$$+ 2^{n-1}\left(q_{n-1}+\sum_{j=0}^{(n-3)/2} p_{n-1-j}p_j\right) \tag{F12d}$$
$$+ \sum_{\ell=(n+1)/2}^{n-1} 2^{2\ell-1}\left(p_\ell+\sum_{j=2\ell-n}^{\ell-1} p_{2\ell-1-j}p_j\right) \tag{F12e}$$
$$+ \sum_{\ell=(n+1)/2}^{n-2} 2^{2\ell}\sum_{j=2\ell-n+1}^{\ell-1} p_{2\ell-j}p_j\,. \tag{F12f}$$

Now the lines are as follows.

1. In Eq. (F12a) three bits are added together.
2. In Eq. (F12b) bits, which are added together and multiplied by odd powers $2^{2\ell-1}$ below 2^{n-1}. There are $\ell+3$ of these bits to be added together.
3. In Eq. (F12c) bits, which are added together and multiplied by *even* powers of $2^{2\ell}$ below 2^{n-1}. There are $\ell+2$ of these bits to be added together.
4. In Eq. (F12d) there are $(n+1)/2$ bits to be added together and multiplied by 2^{n-1}.
5. In Eq. (F12e) there are $n-\ell+1$ bits to be added together and multiplied by odd powers $2^{2\ell-1}$ above 2^{n-1} up to 2^{2n-3}.
6. In Eq. (F12f) there are $n-\ell-1$ bits to be added together and multiplied by even powers $2^{2\ell}$ above 2^{n-1} up to 2^{2n-4}.

The addition works as follows.

1. In Eq. (F12a) the three bits are added with one Toffoli, giving one carry bit.
2. As before, in Eq. (F12b) there are ℓ carry bits from the previous level, giving $2\ell+3$ terms, which can be added together with $\ell+1$ Toffolis and giving $\ell+1$ carry bits.
3. As before, in Eq. (F12c) for term ℓ there are $\ell+1$ carry bits from the previous level and $\ell+2$ bits giving a total of $2\ell+3$, so the number of Toffolis and carry bits are $\ell+1$.
4. In Eq. (F12d), this time there is carry from Eq. (F12b) with $\ell=(n-1)/2$, so there are $(n+1)/2$ carry bits combined with the $(n+1)/2$ bits already in this term, for a total of $n+1$ bits to sum. the number of Toffolis and carry bits to the next level is therefore $(n+1)/2$.
5. In Eq. (F12e), this time there are $n-\ell+1$ bits carried from the previous level. This can be seen for the first term with $\ell=(n+1)/2$ because there are $(n+1)/2$ bits carried from Eq. (F12d). For $\ell=(n+1)/2$ there are $n-\ell'$ bits carried from Eq. (F12f) with $\ell'=\ell-1$. Thus the total number of bits to be summed is $2(n-\ell+1)$, which can be done with $n-\ell+1$ Toffolis and carry bits.
6. In Eq. (F12f), there are $n-\ell+1$ carry bits from Eq. (F12e), plus $n-\ell-1$, for a total of $2(n-\ell)$. These may be summed with $n-\ell$ Toffolis and carry bits.
7. For the last term on Eq. (F12e) with $\ell=n-1$, there are $n-\ell+1=n-(n-1)+1=2$ bits to sum, which takes one more Toffoli.

The total number of Toffolis is therefore

$$1+\sum_{\ell=1}^{(n-1)/2}(\ell+1)+\sum_{\ell=1}^{(n-3)/2}(\ell+1)+(n+1)/2 + \sum_{\ell=(n+1)/2}^{n-1}(n-\ell+1)+\sum_{\ell=(n+1)/2}^{n-2}(n-\ell)+1 = n(n+3)/2-1. \tag{F13}$$

TABLE XVI. Here we show how to compute a contiguous register for an odd number of bits $n = 7$. The entries p_0, p_0p_1, and so forth show entries with different bits. The entries with $\times$ show bits that are carried from lower levels. The first column shows the level number, the second column shows the multiplying factor (power of 2), the third column shows which line from Eq. (F12a) to Eq. (F12f) this corresponds to, the fourth column shows the ℓ value from that equation, and the last column shows the number of Toffolis. The $\cancel{p_0}$ shows identical bits that are added with no cost, and the resulting bit is shown in blue in the next level as p_0.

Level	Factor	Line of equation	ℓ	q	p	p_0	p_1	p_2	p_3	p_4	p_5	p_6			Toffolis
14	2^{13}											$\times$			
13	2^{12}											$\times$	$\times$		1
12	2^{11}	(F12e)	6								p_5p_6	p_6	$\times$	$\times$	2
11	2^{10}	(F12f)	5							p_4p_6	$\times$	$\times$	$\times$		2
10	2^9	(F12e)	5						p_3p_6	p_4p_5	p_5	$\times$	$\times$	$\times$	3
9	2^8	(F12f)	4					p_2p_6	p_3p_5	$\times$	$\times$	$\times$	$\times$		3
8	2^7	(F12e)	4				p_1p_6	p_2p_5	p_3p_4	p_4	$\times$	$\times$	$\times$	$\times$	4
7	2^6	(F12d)		q_6		p_0p_6	p_1p_5	p_2p_4	$\times$	$\times$	$\times$	$\times$			4
6	2^5	(F12b)	3	q_5	p_6	p_0p_5	p_1p_4	p_2p_3	p_3	$\times$	$\times$	$\times$			4
5	2^4	(F12c)	2	q_4	p_5	p_0p_4	p_1p_3	$\times$	$\times$	$\times$					3
4	2^3	(F12b)	2	q_3	p_4	p_0p_3	p_1p_2	p_2	$\times$	$\times$					3
3	2^2	(F12c)	1	q_2	p_3	p_0p_2	$\times$	$\times$							2
2	2	(F12b)	1	q_1	p_2	p_0p_1	p_1	$\times$							2
1	1	(F12a)		q_0	p_1	p_0									1
0	NA				$\cancel{p_0}$	$\cancel{p_0}$									

This is the same formula as before, so adding the $n(n-1)/2$ Toffolis needed to compute the products p_jp_k for $j \neq k$ again gives a total cost of $n^2 + n - 1$. Thus we see that, regardless of whether the number of bits is odd or even, the number of Toffolis is $n^2 + n - 1$. An example for odd $n = 7$ is given in Table XVI.

APPENDIX G: QROM APPLIED TO TWO REGISTERS

When performing QROM on two registers where one is iterating through all values of both registers, it is possible to perform the QROM efficiently without computing a contiguous register as was done in prior work. Say we have registers with variables x and y, which take N_1 and N_2 different values, and so can be represented on $n_1 = \lceil \log N_1 \rceil$ and $n_2 = \lceil \log N_2 \rceil$ bits. We choose k_1 and k_2 to be powers of 2.

Then for the QROM, we iterate through $\lceil N_1/k_1 \rceil$ values on the most significant $n_1 - \log k_1$ bits of x, and $\lceil N_2/k_2 \rceil$ values on the most significant $n_2 - \log k_2$ bits of y. For each of those values, we give the QROM output for all combinations of values on the $\log k_1$ less-significant bits of x and the $\log k_2$ less-significant bits of y. Then the last step is to perform swaps controlled by the less-significant bits of x and y to move the correct data to the output.

For the complexity of this procedure, the complexity of the iteration through the $\lceil N_1/k_1 \rceil$ values on the most significant bits of x is $\lceil N_1/k_1 \rceil - 2$, or $\lceil N_1/k_1 \rceil - 1$ when it needs to be controlled by another qubit. For each of those values on x, it is used as a control for iterating through $\lceil N_2/k_2 \rceil$ values on the most significant bits of y. Since that iteration is done in a controlled way, its cost is $\lceil N_2/k_2 \rceil - 1$. Since it is done $\lceil N_1/k_1 \rceil$ times, the overall cost is

$$\left\lceil \frac{N_1}{k_1} \right\rceil \left(\left\lceil \frac{N_2}{k_2} \right\rceil - 1 \right). \tag{G1}$$

Adding to that the cost of the iteration on x gives a cost of

$$\left\lceil \frac{N_1}{k_1} \right\rceil \left(\left\lceil \frac{N_2}{k_2} \right\rceil - 1 \right) + \left\lceil \frac{N_1}{k_1} \right\rceil - 2 = \left\lceil \frac{N_1}{k_1} \right\rceil \left\lceil \frac{N_2}{k_2} \right\rceil - 2, \tag{G2}$$

where the -2 is if the overall QROM does not need to be made controlled. We omit the -2 for simplicity, and consistency with the way these costs for QROM are usually quoted.

Then the cost of the controlled swaps at the end is identical to what it usually is for the QROM, so for output size b it will be $b(k_1k_2 - 1)$. That gives a total cost (omitting the -2) of

$$\left\lceil \frac{N_1}{k_1} \right\rceil \left\lceil \frac{N_2}{k_2} \right\rceil + b(k_1k_2 - 1). \tag{G3}$$

The cost here should be compared to a cost for the case of a contiguous register

$$\left\lceil \frac{N_1N_2}{k} \right\rceil + b(k - 1), \tag{G4}$$

with $k = k_1k_2$. The cost using a contiguous register will be slightly less, but the difference will often be less than the cost of computing a contiguous register. For example, say $N_1 = 350$, $N_2 = 72$, and $b = 20$. Then with $k_1 = 8$

and $k_2 = 4$, the cost of the QROM is decreased by only 4 using a contiguous register, but the cost of computing the contiguous register is 17.

The number of ancillas is increased to

$$\lceil \log(N_1/k_1) \rceil + \lceil \log(N_2/k_2) \rceil + bk_1k_2. \tag{G5}$$

That is because there are $\lceil \log(N_1/k_1) \rceil$ required for iteration on the first register and $\lceil \log(N_2/k_2) \rceil$ for iteration on the second register, as well as bk_1k_2 for the outputs. In practice this usually only needs one more ancilla than using a contiguous register, which is less than would be needed to store the contiguous register itself.

In the same way, in uncomputing the QROM where it is necessary to perform a phase fixup, there is the same change to the cost for outputting the data, so the cost becomes

$$\left\lceil \frac{N_1}{k_1} \right\rceil \left\lceil \frac{N_2}{k_2} \right\rceil + k_1k_2. \tag{G6}$$

In the example with $N_1 = 350$, $N_2 = 72$, but taking $k_1 = 16$, $k_2 = 8$, the cost is increased only by one Toffoli over the cost for a contiguous register. The number of ancillas needed is

$$\lceil \log(N_1/k_1) \rceil + \lceil \log(N_2/k_2) \rceil + k_1k_2. \tag{G7}$$

[1] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan, Quantum algorithms for quantum chemistry and quantum materials science, arXiv:2001.03685 (2020).

[2] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik, Quantum chemistry in the age of Quantum computing, Chem. Rev. **119**, 10856 (2019).

[3] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan, Quantum computational chemistry, Rev. Mod. Phys. **92**, 015003 (2020).

[4] Alan Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon, Simulated quantum computation of molecular energies, Science **309**, 1704 (2005).

[5] Narbe Mardirossian and Martin Head-Gordon, Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals, Mol. Phys. **115**, 2315 (2017).

[6] K. Ruedenberg, R. C. Raffenetti, and R. D. Bardo, in *Proceedings of the 1972 Boulder Seminar Research Conference on Theoretical Chemistry* (Wiley, New York, 1973), p. 164.

[7] Alexei Y. Kitaev, Quantum measurements and the Abelian Stabilizer Problem, arXiv:quant-ph/9511026 (1995).

[8] Daniel S. Abrams and Seth Lloyd, Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors, Phys. Rev. Lett. **83**, 5162 (1999).

[9] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod McClean, and Ryan Babbush, Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization, Quantum **3**, 208 (2019).

[10] Vera von Burg, Guang Hao Low, Thomas Haner, Damian Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer, Quantum computing enhanced computational catalysis, arXiv:2007.14460 (2020).

[11] James D. Whitfield, Jacob Biamonte, and Alan Aspuru-Guzik, Simulation of electronic structure hamiltonians using quantum computers, Mol. Phys. **109**, 735 (2011).

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

[13] David Poulin, Alexei Y. Kitaev, Damian Steiger, Matthew Hastings, and Matthias Troyer, Fast Quantum Algorithm for Spectral Properties, Phys. Rev. Lett. **121**, 010501 (2017).

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

[15] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan, Low-Depth Quantum Simulation of Materials, Phys. Rev. X **8**, 011044 (2018).

[16] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Phys. Rev. X **8**, 041015 (2018).

[17] Guang Hao Low and Nathan Wiebe, Hamiltonian Simulation in the Interaction Picture, arXiv:1805.00675 (2018).

[18] Ian D. Kivlichan, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Wei Sun, Zhang Jiang, Nicholas Rubin, Austin Fowler, Alán Aspuru-Guzik, Hartmut Neven, and Ryan Babbush, Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization, Quantum **4**, 296 (2020).

[19] Jarrod R. McClean, Fabian M. Faulstich, Qinyi Zhu, Bryan O'Gorman, Yiheng Qiu, Steven R. White, Ryan Babbush, and Lin Lin, Discontinuous galerkin discretization for quantum simulation of chemistry, New J. Phys. **22**, 093015 (2020).

[20] Steven R. White, Hybrid grid/basis set discretizations of the Schrödinger equation, J. Chem. Phys. **147**, 244102 (2017).

[21] Ivan Kassal, Stephen P. Jordan, Peter J. Love, Masoud Mohseni, and Alan Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, Proc. Natl. Acad. Sci. **105**, 18681 (2008).

[22] Ryan Babbush, Dominic W. Berry, Jarrod R. McClean, and Hartmut Neven, Quantum simulation of chemistry with sublinear scaling in basis size, Npj Quantum Inf. **5**, 92 (2019).

[23] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer, Elucidating reaction mechanisms on quantum computers, Proc. Natl. Acad. Sci. **114**, 7555 (2017).

[24] Helmut Beinert, Richard Holm, and Eckard Munck, Iron-sulfur clusters: Nature's modular, multipurpose structures, Science **277**, 653 (1997).

[25] Jarrod R. McClean, Ryan Babbush, Peter J. Love, and Alan Aspuru-Guzik, Exploiting locality in quantum computation for quantum chemistry, J. Phys. Chem. Lett. **5**, 4368 (2014).

[26] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 32324 (2012).

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

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

[29] Here and throughout the paper we use $\widetilde{\mathcal{O}}(\cdot)$ to indicate an asymptotic upper bound suppressing polylogarithmic factors in the scaling.

[30] Mario Motta, Erika Ye, Jarrod R. McClean, Zhendong Li, Austin J. Minnich, Ryan Babbush, and Garnet Kin-Lic Chan, 'Low rank representations for quantum simulation of electronic structure, Npj Quantum Inf. **7**, 1 (2021).

[31] Francesco Aquilante, Linus Boman, Jonas Boström, Henrik Koch, Roland Lindh, Alfredo Sánchez de Merás, and Thomas Bondo Pedersen, in *Linear-Scaling Techniques in Computational Chemistry and Physics* (Springer, 2011), p. 301.

[32] J. L. Whitten, Coulombic potential energy integrals and approximations, J. Chem. Phys. **58**, 4496 (1973).

[33] E. J. Baerends, D. E. Ellis, and P. Ros, Self-consistent molecular Hartree-Fock-Slater calculations I. The computational procedure, Chem. Phys. **2**, 41 (1973).

[34] J. A. Jafri and J. L. Whitten, Electron repulsion integral approximations and error bounds: Molecular applications, J. Chem. Phys. **61**, 2116 (1974).

[35] Bo Peng and Karol Kowalski, Highly efficient and scalable compound decomposition of two-electron integral tensor and its application in coupled cluster calculations, J. Chem. Theory Comput. **13**, 4179 (2017).

[36] Zhendong Li, Junhao Li, Nikesh S. Dattani, C. J. Umrigar, and Garnet Kin-Lic Chan, The electronic complexity of the ground-state of the FeMo cofactor of nitrogenase as relevant to quantum simulations, J. Chem. Phys. **150**, 024302 (2019).

[37] Motta, Shee, Zhang, and Chan, J. Chem. Theory Comput. **15**, 3510 (2019).

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

[39] Edward G. Hohenstein, Robert M. Parrish, and Todd J. Martínez, Tensor hypercontraction density fitting. I. Quartic scaling second- and third-order Møller-Plesset perturbation theory, J. Chem. Phys. **137**, 1085 (2012).

[40] Robert M. Parrish, Edward G. Hohenstein, Todd J. Martínez, and C. David Sherrill, Tensor hypercontraction. II. Least-squares renormalization, J. Chem. Phys. **137**, 224106 (2012).

[41] Edward G. Hohenstein, Robert M. Parrish, C. David Sherrill, and Todd J. Martínez, Communication: Tensor hypercontraction. III. Least-squares tensor hypercontraction for the determination of correlated wavefunctions, J. Chem. Phys. **137**, 221101 (2012).

[42] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush, Quantum Simulation of Electronic Structure with Linear Depth and Connectivity, Phys. Rev. Lett. **120**, 110501 (2018).

[43] Ian D. Kivlichan, Nathan Wiebe, Ryan Babbush, and Alan Aspuru-Guzik, Bounding the costs of quantum simulation of many-body physics in real space, J. Phys. A: Math. Theor. **50**, 305301 (2017).

[44] Jacob T. Seeley, Martin J. Richard, and Peter J. Love, The bravyi-kitaev transformation for quantum computation of electronic structure, J. Chem. Phys. **137**, 224109 (2012).

[45] David Wecker, Bela Bauer, Bryan K. Clark, Matthew B. Hastings, and Matthias Troyer, Gate-count estimates for performing quantum chemistry on small quantum computers, Phys. Rev. A **90**, 022305 (2014).

[46] Borzu Toloui and Peter J. Love, Quantum algorithms for quantum chemistry based on the sparsity of the CI-matrix, arXiv:1312.2579 (2013).

[47] Matthew B. Hastings, Dave Wecker, Bela Bauer, and Matthias Troyer, Improving quantum algorithms for quantum chemistry, Quantum Inf. Comput. **15**, 1 (2015).

[48] David Poulin, M. B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doherty, and Matthias Troyer, The trotter step size required for accurate quantum simulation of quantum chemistry, Quantum Inf. Comput. **15**, 361 (2015).

[49] Ryan Babbush, Jarrod McClean, Dave Wecker, Alan Aspuru-Guzik, and Nathan Wiebe, Chemical basis of trotter-suzuki errors in chemistry simulation, Phys. Rev. A **91**, 22311 (2015).

[50] Ryan Babbush, Dominic W. Berry, Ian D. Kivlichan, Annie Y. Wei, Peter J. Love, and Alan Aspuru-Guzik, Exponentially more precise quantum simulation of fermions in second quantization, New J. Phys. **18**, 33032 (2016).

[51] Ryan Babbush, Dominic W. Berry, Yuval R. Sanders, Ian D. Kivlichan, Artur Scherer, Annie Y. Wei, Peter J. Love, and Alan Aspuru-Guzik, Exponentially more precise quantum simulation of fermions in the configuration interaction representation, Quantum Sci. Technol. **3**, 015006 (2018).

[52] Earl Campbell, Random Compiler for Fast Hamiltonian Simulation, Phys. Rev. Lett. **123**, 070503 (2019).

[53] Ian D. Kivlichan, Christopher E. Granade, and Nathan Wiebe, Phase estimation with randomized Hamiltonians, arXiv:1907.10070 (2019).

[54] Craig Gidney and Austin G. Fowler, Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation, Quantum **3**, 135 (2019).

[55] Robert M. Parrish, Edward G. Hohenstein, Todd J. Martínez, and C. David Sherrill, Discrete variable representation in electronic structure theory: Quadrature grids for least-squares tensor hypercontraction, J. Chem. Phys. **138**, 194107 (2013).

[56] Edward G. Hohenstein, Sara I. L. Kokkila, Robert M. Parrish, and Todd J. Martínez, Quartic scaling second-order approximate coupled cluster singles and doubles via tensor hypercontraction: THC-CC2, J. Chem. Phys. **138**, 124111 (2013).

[57] Edward G. Hohenstein, Sara I. L. Kokkila, Robert M. Parrish, and Todd J. Martínez, Tensor hypercontraction equation-of-motion second-order approximate coupled cluster: Electronic excitation energies in $\mathcal{O}(N^4)$ time, J. Phys. Chem. B **117**, 12972 (2013).

[58] Udo Benedikt, Karl-Heinz Böhm, and Alexander A. Auer, Tensor decomposition in post-Hartree–Fock methods. II. CCD implementation, J. Chem. Phys. **139**, 224101 (2013).

[59] Robert M. Parrish, C. David Sherrill, Edward G. Hohenstein, Sara I. L. Kokkila, and Todd J. Martínez, Communication: Acceleration of coupled cluster singles and doubles via orbital-weighted least-squares tensor hypercontraction, J. Chem. Phys. **140**, 181102 (2014).

[60] Jianfeng Lu and Lexing Ying, Compression of the electron repulsion integral tensor in tensor hypercontraction format with cubic scaling cost, J. Comput. Phys. **302**, 329 (2015).

[61] Sara I. L. Kokkila Schumacher, Edward G. Hohenstein, Robert M. Parrish, Lee Ping Wang, and Todd J. Martínez, Tensor hypercontraction second-order Møller-Plesset perturbation theory: Grid optimization and reaction energies, J. Chem. Theory Comput. **11**, 3042 (2015).

[62] Chenchen Song and Todd J. Martínez, Atomic orbital-based SOS-MP2 with tensor hypercontraction. I. GPU-based tensor construction and exploiting sparsity, J. Chem. Phys. **144**, 174111 (2016).

[63] Jianfeng Lu and Lexing Ying, Fast algorithm for periodic density fitting for bloch waves, Ann. Math. Sci. Appl. **1**, 321 (2016).

[64] Wei Hu, Lin Lin, and Chao 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).

[65] Chenchen Song and Todd J. Martínez, Atomic orbital-based SOS-MP2 with tensor hypercontraction. II. Local tensor hypercontraction, J. Chem. Phys. **146**, 034104 (2017).

[66] Jianfeng Lu and Kyle Thicke, Cubic scaling algorithms for RPA correlation using interpolative separable density fitting, J. Comput. Phys. **351**, 187 (2017).

[67] Chenchen Song and Todd J. Martínez, Analytical gradients for tensor hyper-contracted MP2 and SOS-MP2 on graphical processing units, J. Chem. Phys. **147**, 161723 (2017).

[68] Felix Hummel, Theodoros Tsatsoulis, and Andreas Grüneis, Low rank factorization of the coulomb integrals for periodic coupled cluster theory, J. Chem. Phys. **146**, 124105 (2017).

[69] Roman Schutski, Jinmo Zhao, Thomas M. Henderson, and Gustavo E. Scuseria, Tensor-structured coupled cluster theory, J. Chem. Phys. **147**, 184113 (2017).

[70] Kun Dong, Wei Hu, and Lin Lin, Interpolative separable density fitting through centroidal voronoi tessellation with applications to hybrid functional electronic structure calculations, J. Chem. Theory Comput. **14**, 1311 (2018).

[71] Wei Hu, Meiyue Shao, Andrea Cepellotti, Felipe H. da Jornada, Lin Lin, Kyle Thicke, Chao Yang, and Steven G. Louie, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Springer, Cham, 2018), Vol. 10861 LNCS, p. 604.

[72] Ivan Duchemin and Xavier Blase, Separable resolution-of-the-identity with all-electron Gaussian bases: Application to cubic-scaling RPA, J. Chem. Phys. **150**, 174120 (2019).

[73] Joonho Lee, Lin Lin, and Martin 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).

[74] Devin A. Matthews, Improved grid optimization and fitting in least squares tensor hypercontraction, J. Chem. Theory Comput. **16**, 1382 (2020).

[75] William J. Huggins, Jarrod R. McClean, Nicholas C. Rubin, Zhang Jiang, Nathan Wiebe, K. Birgitta Whaley, and Ryan Babbush, Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers, Npj Quantum Inf. **7**, 23 (2021).

[76] Data and code repository for "even more efficient quantum computations of chemistry through tensor hypercontraction", (2020).

[77] Pauli Virtanen *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in python, Nat. Methods **17**, 261 (2020).

[78] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne, JAX: composable transformations of Python+NumPy programs, (2018).

[79] John Duchi, Elad Hazan, and Yoram Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Machine Learning Res. **12**, 2121 (2011).

[80] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu, Theory of Trotter Error with Commutator Scaling, Phys. Rev. X **11**, 011020 (2021).

[81] Andrew M. Childs and Nathan Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[82] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019), p. 193.

[83] Mario Szegedy, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2004), p. 32.

[84] Yuval R. Sanders, Dominic W. Berry, Pedro C. S. Costa, Louis W. Tessler, Nathan Wiebe, Craig Gidney, Hartmut Neven, and Ryan Babbush, Compilation of fault-tolerant quantum heuristics for combinatorial optimization, PRX Quantum **1**, 020312 (2020).

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

[86] Attila Szabo and Neil S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Dover Publications, Mineola, New York, 1996).

[87] Isaiah Shavitt and Rodney J. Bartlett, *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory* (Cambridge University Press, Cambridge University Press is in Cambridge (UK), 2009).
[88] Joonho Lee and Martin Head-Gordon, Regularized orbital-optimized second-order møller–plesset perturbation theory: A reliable fifth-order-scaling electron correlation model with orbital energy dependent regularizers, J. Chem. Theory Comput. **14**, 5203 (2018).
[89] Krishnan Raghavachari, Gary W. Trucks, John A. Pople, and Martin Head-Gordon, A fifth-order perturbation comparison of electron correlation theories, Chem. Phys. Lett. **157**, 479 (1989).
[90] Florian Weigend, Accurate coulomb-fitting basis sets for H to Rn, Phys. Chem. Chem. Phys. **8**, 1057 (2006).
[91] Francesco Aquilante, Luca De Vico, Nicolas Ferré, Giovanni Ghigo, Per-åke Malmqvist, Pavel Neogrády, Thomas Bondo Pedersen, Michal Pitoňák, Markus Reiher, Björn O. Roos, Luis Serrano-Andrés, Miroslav Urban, Valera Veryazov, and Roland Lindh, Molcas 7: The next generation, J. Comput. Chem. **31**, 224 (2010).
[92] Thom H. Dunning, Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen, J. Chem. Phys. **90**, 1007 (1989).
[93] Clyde Edmiston and Klaus Ruedenberg, Localized atomic and molecular orbitals, Rev. Mod. Phys. **35**, 457 (1963).
[94] Austin G. Fowler, Time-optimal quantum computation, arXiv:1210.4626 (2012).
[95] Craig Gidney and Austin G. Fowler, Flexible layout of surface code computations using AutoCCZ states, arXiv:1905.08916 (2019).
[96] Daniel Litinski, Magic state distillation: Not as costly as you think, Quantum **3**, 205 (2019).
[97] Craig Gidney and Martin Ekerå, How to factor 2048 bit RSA integers in 8 h using 20 million noisy qubits, arXiv:1905.09749 (2019).
[98] Andrew M. Childs and Yuan Su, Nearly Optimal Lattice Simulation by Product Formulas, Phys. Rev. Lett. **123**, 050503 (2019).
[99] Xin Xing, Hua Huang, and Edmond Chow, A linear scaling hierarchical block low-rank representation of the electron repulsion integral tensor, J. Chem. Phys. **153**, 084119 (2020).
[100] Lin Lin and Yu Tong, Near-optimal ground state preparation, Quantum **4**, 372 (2020).
[101] Jarrod McClean, Ryan Babbush, Peter Love, and Alán Aspuru-Guzik, Exploiting locality in quantum computation for quantum chemistry, J. Phys. Chem. Lett. **5**, 4368 (2014).
[102] Frank Verstraete, J. Ignacio Cirac, and José I Latorre, Quantum circuits for strongly correlated quantum systems, Phys. Rev. A **79**, 32316 (2009).
[103] Guang Hao Low and Isaac L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, Phys. Rev. Lett. **118**, 010501 (2017).
[104] Ryan Babbush, Dominic W. Berry, and Hartmut Neven, Quantum simulation of the sachdev-ye-kitaev model by asymmetric qubitization, Phys. Rev. A **99**, 040301 (2019).
[105] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe, Time-dependent hamiltonian simulation with l1-norm scaling, Quantum **4**, 254 (2020).
[106] J. L. Hodges and E. L. Lehmann, Estimates of location based on rank tests, Ann. Math. Statistics **34**, 598 (1963).
[107] E. L. Lehmann, *Elements of Large-Sample Theory* (Springer-Verlag, New York, 1999).